Analysis and control of online interactions through neural natural language processing

Léo Laugier

To cite this version:

Léo Laugier. Analysis and control of online interactions through neural natural language processing. Artificial Intelligence [cs.AI]. Institut Polytechnique de Paris, 2022. English. NNT : 2022IPPAT035 . tel-03884481

HAL Id: tel-03884481
https://theses.hal.science/tel-03884481v1

Submitted on 5 Dec 2022

NNT : 2022IPPAT035

Thèse de doctorat

Analysis and Control of Online Interactions through Neural Natural Language Processing

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à Télécom Paris

École doctorale n°626 École doctorale de l'Institut Polytechnique de Paris (ED IP Paris)
Spécialité de doctorat: Mathématiques et Informatique

Thèse présentée et soutenue à Palaiseau, le 08 novembre 2022, par

LÉO LAUGIER

Composition du Jury :

Benoît Sagot Directeur de recherche, INRIA (ALMAnaCH)	Président
Serena Villata Chargée de recherche, CNRS/Université Côte d'Azur (I3S/3IA)	Rapporteur
François Yvon Directeur de recherche, CNRS/Université Paris-Saclay (LISN)	Rapporteur
Ion Androutsopoulos Professor, Athens University of Economics and Business	Examinateur
Marine Carpuat Assistant Professor, University of Maryland (Department of CS)	Examinateur
Slav Petrov Distinguished Scientist, Google AI Research	Examinateur
Thomas Bonald Professeur, Télécom Paris (LTCI)	Directeur de thèse
Lucas Dixon Research scientist, Google AI Research (PAIR)	Co-directeur de thèse

Cette thèse de doctorat est dédiée à mes parents.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisors, **Thomas Bonald** and **Lucas Dixon**, who chose three years ago to trust me and keenly supported the university-industry collaboration leading to this research project. Their complementary scientific advice actively matured my scientific endeavor. I learned a lot from their sides and I wish to thank them for being great sources of inspiration and supportive in all matters. Moreover, I was extremely lucky to have been supervised by such exceptionally kind, available and patient researchers, giving me enough freedom to pursue ambitious ideas. I could not have wished for a better work atmosphere, both in the laboratory and remotely, and I sincerely hope our collaboration will not stop after this Ph.D. ends.

I am honored to have six outstanding external researchers on my defense committee. In particular, I thank **Serena Villata** and **François Yvon** from CNRS, who agreed to spend time and effort reviewing this thesis. I also thank the other examiners, **Ion Androutsopoulos**, **Marine Carpuat**, **Slav Petrov**, and **Benoît Sagot**, for having accepted to share their world-renowned expertise by participating in my committee. I am very grateful to them and hope that reading this thesis will be pleasant for them.

This Ph.D. project would never have been possible without **Google**'s doctoral fellowship funding the entirety of my research. Thanks to their generous support, I could work in excellent material and intellectual conditions and travel to all conferences I wanted to attend. Besides, I would like to thank **Google Cloud Plaform** research credits program for the grant they offered me to store data and train models on their TPUs. I have been fortunate to collaborate with many talented researchers and engineers whose suggestions were always highly valuable to me. I strongly benefited from **Jeffrey Sorensen**'s expertise. I want to express my deep gratitude for our many fruitful discussions, the time he spent reviewing all our papers, and his effort in our projects. I am greatly thankful for the guidance of **Nithum Thain** in my first Ph.D. year. Many thanks to the members of the **Jigsaw** and **PAIR** teams at Google Research, especially **Fernando Diaz**, **Douglas Eck**, **Bastien Girschig**, **Misha Khalman**, **Preethi Lahoti**, **Federico Lopez**, **Marie Pellat**, **Ludovic Peran**, **Roberto Rama**, **Gerard Serra**, **Michael Terry**, **Raghuram Vadapalli**, **Ben Wedin** and **Jessica Yung**.

In addition, I would like to express my sincerest gratitude to **Ioannis Pavlopoulos** from Stockholm University and the Athens University of Economics and Business. Besides his never-failing encouragement, I would like to thank him for his friendly support throughout these years, his hard work, his dedication, and the time he spent reviewing several of my papers. I am also glad that we could co-author several of them together.

I sincerely enjoyed every moment I spent in the **DIG** research group at the Institut Polytechnique de Paris, talking about the very nature of intelligence to the best way of coding rare words with *Concept*'s icons. Notably, I thank my journey companions who used to be or still are my fellow Ph.D. students: **Céline Comte**, **Nathan de Lara**, **Edouard Pineau**, **Quentin Lutz**, **Armand Boschin**, **Simon Delarue**, and **Rajaa El Hamdani**. Ph.D. life would not have been much fun without my brilliant lab mates. I do not forget **Marc Jeanmougin** for his precious help regarding IT issues or theoretical computer science questions. I thank him for his benevolence and his constant support. I would also like to express my gratitude to all the other rock stars I have crossed paths with at the **Institut Polytechnique de Paris**: **Antoine Amarilli**, **Mariam Barry**, **Nicolas Bouche**, **Lihu Chen**, **Cyril Chhun**, **Jean-Louis Dessalles**, **Georges Hebrail**, **Chadi Helwé**, **Louis Jachiet**, **Lanfang Kong**, **Minh Huong Le Nguyen**, **Pierre-Henri Paris**, **Nedeljko Radulovic**, **Zacchary Sadeddine**, **Mauro Sozio**, **Fabian Suchanek**, and **Tiphaine Viard**.

I have a special thought for Nokia Bell Labs and INRIA researchers from the LINCS laboratory, and with whom I spent my first Ph.D. year. I greatly improved my Python skills thanks to **Marc-Olivier Buob**, **François Durand**, **Fabien Mathieu**, **Ludovic Noirie** and **Élie de Panafieu**.

As this Ph.D. thesis concludes more than twenty years of my formal education, I would like to thank all the professors who encouraged me during my time in Paris, Los Alamos, Saclay, and Berkeley. In particular, **Loys Bonod** and **Jean-Marc Sarfati** prepared me well for the academic and scientific rigor. **Jean-Claude Pena** made computer science my vocation as early as high school; **Paul A. Johnson** convinced me to pursue research, and **Laurent El Ghaoui** made me want to specialize in machine learning.

Furthermore, I am indebted to all engineering and scientific community members who contributed to the Python open-source libraries and tools cited and used throughout this thesis. In particular, many thanks to **Adam Roberts** from Google Research, as my experiments often relied on his Tensorflow implementation of T5. Finally, like many computer scientists, I also owe a lot to **Mathieu Nebra** and his major contributions to making computer science more accessible.

On a more personal level, I would like to give a special thanks to my **close friends**. While I cannot cite everyone, be sure that I have a very special thought for you. In particular, life would not be the same without the never-failing support of my dear friends **Arnaud**, **Bruno**, **Étienne**, **Juba**, **Nicolas**, and **Rafaël**, with whom I had many debates about the essence of a Ph.D.

Last but not least, I am forever indebted to my **family**. Again, as it is hard for me to mention everyone, I address a collective "thank you" to all. I dedicate a few additional words to some of them. I thank **Violaine** for her motivation and for keeping up with my ups and downs during the Ph.D. I am also grateful to **Lucas** for his unconditional support over the years. I have a special thought for **my grandparents**, especially for **Danielle**. Finally, I am deeply thankful to **my parents**. They sparked my interest in science, but more importantly, they ensured that I had a fantastic education. They have contributed more than they will ever imagine to this journey.

Abstract

Natural Language Processing is motivated by applications where computers should gain a semantic and syntactic understanding of human language. Recently, the field has been impacted by a paradigm shift. Deep learning architectures coupled with self-supervised training have become the core of state-of-the-art models used in Natural Language Understanding and Natural Language Generation. Sometimes considered as foundation models, these systems pave the way for novel use cases. Driven by an academic-industrial partnership between the Institut Polytechnique de Paris and Google AI Research, the present research has focused on investigating how pretrained neural Natural Language Processing models could be leveraged to improve online interactions.

This thesis first explored how self-supervised style transfer could be applied to the toxic-to-civil rephrasing of offensive comments found in online conversations. In the context of toxic content moderation online, we proposed to fine-tune a pretrained text-to-text model (T5) with a denoising and cyclic auto-encoder loss. The system, called CAE-T5, was trained on the largest toxicity detection dataset to date (Civil Comments) and generates sentences that are more fluent and better at preserving the initial content compared to earlier text style transfer systems, according to several scoring systems and human evaluation. Plus, the approach showed it could be generalized to additional style transfer tasks, such as sentiment transfer.

Then, a subsequent work investigated the human labeling and automatic detection of toxic spans in online conversations. Contrary to toxicity detection datasets and models which classify whole posts as toxic or not, toxic spans detection aims at highlighting toxic spans, that is to say, the spans that make a text toxic when detecting such spans is possible. We released a new labeled dataset to train and evaluate systems, which led to a shared task at the 15th International Workshop on Semantic Evaluation. Systems proposed to address the task include strongly supervised models trained using annotations at the span level as well as weakly supervised approaches, known as rationale extraction, using classifiers trained on potentially larger external datasets of posts manually annotated as toxic or not, without toxic span annotations. Furthermore, the ToxicSpans dataset and systems proved useful to analyzing the performances of humans and automatic systems on toxic-to-civil rephrasing.

Finally, we developed a recommender system based on online reviews of items, taking part in the topic of explaining users' tastes considered by the predicted recommendations. The method uses textual semantic similarity models to represent a user's preferences as a graph of textual snippets, where the edges are defined by semantic similarity. This textual, memory-based approach to rating prediction holds out the possibility of improved explanations for recommendations. The method is evaluated quantitatively, highlighting that leveraging text in this way can outperform both memory-based and model-based collaborative filtering baselines.

Résumé

Le traitement automatique du langage naturel est motivé par des applications où les ordinateurs doivent acquérir une compréhension sémantique et syntaxique du langage humain. Récemment, le domaine a été impacté par un changement de paradigme. Les architectures d'apprentissage profond couplées à des techniques d'apprentissage auto-supervisé sont devenues le cœur des modèles correspondant à l'état de l'art en compréhension et génération du langage naturel. Parfois considérés comme des "foundation models", ces systèmes ouvrent la voie à de nouveaux cas d'utilisation. Née d'un partenariat académique et industriel entre l'Institut Polytechnique de Paris et Google AI Research, la présente recherche s'est concentrée sur l'étude de la façon dont les modèles neuronaux de traitement du langage naturel pré-entraînés pouvaient être utilisés pour améliorer les interactions en ligne.

Cette thèse a d'abord exploré comment le transfert de style auto-supervisé pouvait être appliqué à la reformulation non-toxique de commentaires offensants dans les conversations en ligne. Dans le contexte de la modération de contenu toxique en ligne, nous avons proposé une méthode de réglage fin d'un modèle texte-à-texte pré-entraîné (T5) avec une fonction-objectif consistant en un auto-encodeur débruiteur cyclique. Le système, baptisé CAE-T5, a été entraîné sur le plus grand jeu de données de détection de toxicité publé à ce jour (Civil Comments) et génère des phrases plus fluides et préservant mieux le contenu initial, comparé aux systèmes antérieurs de transfert de style de texte, selon plusieurs systèmes d'évaluation automatique et une étude faisant appelle à l'évaluation humaine. De plus, l'approche a montré qu'elle pouvait être généralisée à d'autres tâches de transfert de style, comme le transfert de sentiments.

Ensuite, les travaux de recherche ont porté sur l'étude de l'annotation humaine et la détection automatique des sous-ensembles de mots toxiques dans les conversations en ligne. Contrairement aux jeux de données et aux modèles de détection de toxicité qui classifient des messages entiers comme toxiques ou non, la détection des mots toxiques vise à mettre en évidence les mots responsables de la toxicité du message, lorsqu'une telle détection est possible. Nous avons publié un nouveau jeu de données annoté pour entraîner et évaluer les systèmes automatiques, ce qui a conduit à une tâche partagée lors du 15e International Workshop on Semantic Evaluation. Les systèmes proposés pour cette tâche comprennent des modèles fortement supervisés, entraînés à l'aide d'annotations au niveau des mots, ainsi que des approches faiblement supervisées, connues sous le nom d'extraction de raisons, utilisant des classifieurs entraînés sur des ensembles de données externes, potentiellement plus importants, de

messages annotés manuellement comme toxiques ou non, sans annotations à l'échelle des mots. En outre, le jeu de données et les systèmes se sont avérés utiles pour analyser les performances des systèmes automatiques et des humains en matière de reformulation des messages toxiques en messages civils.

Enfin, nous avons développé un système de recommandation basé sur des avis en ligne, s'inscrivant dans l'explicabilité des préférences prises en compte par les recommandations prédites. La méthode utilise des modèles basés sur la similarité sémantique textuelle pour représenter les préférences d'un utilisateur sous la forme d'un graphe de fragements de texte, où les arrêtes sont définies par la similarité sémantique. Ce modèle de prédiction de notes à mémoire, basé sur le texte, offre la possibilité d'améliorer les explications des recommandations. La méthode est évaluée quantitativement, et nous permet de conclure que l'exploitation du texte de cette manière peut surpasser les performances de modèles de référence utilisé en filtrage collaboratif.

Contents

Notation

Datasets

V Vocabulary of $|V|$ tokens (e.g. subwords). We use the same notation for a token and its one-hot encoding.

s Sequence (e.g. sentence) $s^1 s^2 \dots s^m$ of m tokens.

x_i The i$^{\text{th}}$ input data point. In the context of text transfer, the input is called the source text.

y_i The i$^{\text{th}}$ target, associated with x_i. If produced by a human, the target is called an annotation. If it is a noisy version of the input, the target is called a pseudo-label, otherwise it is referred to as a label. To differentiate the predicted output with the target, the latter is often referred to as ground-truth output. When there is no confusion, the target y is associated with input x.

$\hat{y}_i$ The i$^{\text{th}}$ predicted output, associated with x_i.

Numbers and Arrays

a A scalar

$\mathbf{a}$ A vector

$\mathbf{A}$ A matrix

Optimization

$\mathcal{L}$ A loss function.

Probability

X The random variable observed by the x_i.

$P\left(X\right)$ The data-generating distribution of X.

$\hat{P}\left(X\right)$ The empirical probability of X observed by the training set.

$P_\theta\left(X\right)$ The model distribution of X, parametrized by weight vector θ.

$P(x;\theta)$ The probability $P_\theta(X = x)$ that $X = x$.

$\bar{A}$ The opposite of event A.

Sequences

$\frown$ Concatenation operator: $(a_1, a_2, \ldots, a_n) \frown (b_1, b_2, \ldots, b_m) = (a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_n)$

: Slicing operator: $a_{1:n} = (a_1, a_2, \ldots, a_n)$

Acronyms

k-NN k-Nearast Neighbors. 10, 106, 108, 109, 111–113, 117, 127, 130

AE Auto-Encoder. 32, 33, 40, 42, 57, 71

AI Artificial Intelligence. 13, 14, 18–21, 26, 27, 44–46, 52, 53, 56, 129, 130

ANN Artificial Neural Network. 9, 23, 29, 46, 47, 49–54, 56, 59

BERT Bidirectional Encoder Representations from Transformers. 23, 31, 40, 42, 44–46, 50, 66, 68, 71–73, 92, 135, 136, 139, 140

BLEU Bilingual Evaluation Understudy. 20, 67, 71, 73, 82

CAE-T5 Conditional Auto-Encoder Text-To-Text Transfer Transformer. 57, 64, 65, 68, 72, 73, 75–80, 82, 129, 130

CC Cycle Consistency. 59, 60, 63–65, 71

CF Collaborative Filtering. 106, 108, 109, 115

CNLG Conditional Natural Language Generation. 20, 21, 42, 43, 80, 81

CNN Convolutional Neural Network. 23, 51, 52, 59, 91

DAE Denoising Auto-Encoder. 23, 33, 40, 59, 61–65, 71

DL Deep Learning. 14, 15, 45, 46, 49, 51, 56, 59, 117, 127, 129

FCNN Fully-Connected Neural Network. 23, 48, 49, 51

FCP Fraction of Concordant Pairs. 108, 109, 116, 127

FFNN Feedforward Neural Network. 9, 23, 47, 48, 50–52

GAN Generative Adversarial Network. 59

GLUE General Language Understanding Evaluation. 19, 22, 40, 46

GPT Generative Pre-trained Transformer. 23, 31, 38, 43–46, 66, 71, 73

GPU Graphics Processing Unit. 45, 127

HPC High-Performance Computing. 51, 54, 56

LM Language Model. 11, 15, 23, 34–38, 40, 42–46, 54, 62–64, 71, 73, 80, 125, 127, 129–131

LSTM Long Short-Term Memory. 23, 52, 135, 136, 139, 140

ML Machine Learning. 14, 15, 17, 19, 24, 26–28, 34, 35, 37, 43, 45–47, 50, 54, 56–58, 82, 106, 129–131

MLM Masked Language Model. 32, 40

NLG Natural Language Generation. 19–22, 28, 31, 35, 37, 38, 42, 45, 52, 58, 61, 62, 82

NLP Natural Language Processing. 9, 11, 14, 15, 17–24, 26, 29–36, 38, 42–46, 48, 50–52, 54, 56, 58, 111, 127, 129, 130

NLU Natural Language Understanding. 19–22, 28, 31, 32, 35, 38–40, 42, 43, 50, 52, 58, 82, 105, 106, 127

PLM Pretrained Language Model. 23, 45, 106

PPL Perplexity. 20, 66, 68, 71–73, 79, 80, 97–99

RMSE Root-Mean-Square Error. 108, 109, 115–119

RNN Recurrent Neural Network. 23, 52–54, 62

ROC-AUC Area Under the Receiver Operating Characteristic Curve. 66

SuperGLUE Super General Language Understanding Evaluation. 19, 22, 46

SVD Singular Value Decomposition. 108, 117

t-SNE t-distributed Stochastic Neighbor Embedding. 10, 63, 107, 111

T5 Text-To-Text Transfer Transformer. 23, 31, 43–46, 64, 65, 125

TFCP Time-based Fraction of Concordant Pairs. 10, 116–120, 126

TPU Tensor Processing Unit. 45, 71

UNLG Unconditional Natural Language Generation. 20, 21, 27, 31, 38, 42, 43

USE Universal Sentence Encoder. 10, 51, 67, 71, 80, 111, 112, 125–127

List of Figures

List of Tables

Chapter 1

Introduction

1.1 General context: bridging the gap between human language and semantically-driven computer science

Natural language is a straightforward communication system for humans. It has evolved naturally in human brains and intertwines technological change. Indeed, the development of spoken languages enabled civilizations to thrive by easing knowledge transfer. Homo sapiens individuals, while living in communities, unconsciously learn to speak or sign, and children quickly gain the cognitive ability to express rich and subjective semantics. With the first writing systems [248], ancient Mesopotamians invented a revolutionary concept to spread complex quantitative and qualitative information. Johannes Gutenberg scaled this idea with printing technology, paving the way for the scientific revolution and making general knowledge available to the masses. This trend has been accelerated in the last two centuries with the advent of information and communications technology. On the one hand, logicians and computer scientists constructed formal languages for machines to execute programs. On the other hand, the general public started adapting to computers and designing countless applications that fit their habits. Further, the Internet made people around the world more connected. Nowadays, social networks and online platforms are integral parts of our daily lives. With the democratization of social interactions online, humans are accustomed to sharing their opinions on various topics; people gather and talk at a scale never seen in human history. While online participation offered exciting opportunities to humanity, it also engendered threats and challenges for the twenty-first century. In this thesis, we addressed some of them in light of recent progress in Artificial Intelligence (AI). In particular, our approach focused on bridging the gap between highly informal human experiences of the world and the formal technologies that computers rely on.

1.2 Scientific context: recent progress in artificial intelligence prepares the ground for major technological shifts

Research in AI studies agents that have some perception of their environment and take actions in order to achieve one or several goals [394] . Intelligence refers to natural intelligence found among animals, and AI focuses on the ability of machines to make decisions automatically. Following Alan Turing's theory of computation, AI foundations have been built up by computer scientists and mathematicians as a formal science. Yet, its very nature makes it deeply connected to many disciplines in social, natural, or applied sciences and this thesis is an example of the interdisciplinarity of AI. Specifically, we narrowed down our exploration to Machine Learning (ML), a statistical and computational paradigm consisting of algorithms improved by the automatic learning of relevant signals from data. ML has become ubiquitous in AI for its ability to train models that yield top performances in perception (e.g., computer vision, natural language processing), information retrieval, robotics or predictive analytics for empirical (i.e., natural and social) sciences, without manual—and often labor-intensive—feature engineering. In its standard "supervised" learning form, the goal is to infer a probability distribution $P(y|x;\theta)$ matching unnoisy observations of input x_i and their label y_i.

In the 2010s, Big Data processing and specialized hardware scaled the applications of a family of ML methods called Deep Learning (DL) [276], directly inspired by biological neural networks to tackle cognitive functions. The DL revolution started when deep artificial neural networks won large-scale computer vision tasks [249]. Subsequently, neural architectures have been applied to other fields and showed remarkable results. Thus, Natural Language Processing (NLP) research shifted towards neural (connectionist) approaches, and this thesis sought to take part in this scientific trend by exploring new and challenging applications of DL for NLP. In fact, a question central to this work is to what extent could these AI models impact people's behavior in their social interactions online.

1.3 Research context: an academic-industrial partnership

The genesis of this Ph.D. results from a collaboration between the team Data, Intelligence and Graphs (DIG) of the Information Processing and Communications Laboratory at Télécom Paris and the Conversation AI team from Jigsaw, a unit within Google that explores global challenges and threats online (disinformation, censorship, cyberbullying and violent extremism), and develops scalable technological solutions for the safety of open societies. The DIG team has a long experience in industrial collaborations, and this research benefited from the synergy produced by the mixed environment. In particular, it combined the scientific rigor of fundamental research with a clear vision of concrete downstream applications.In addition, the research was enriched by collaborations with People + AI Research, a multidisciplinary team at Google Research that explores the human side of AI, and the Department of Informatics at the Athens University of Economics and Business, which has strong expertise in NLP.

1.4 Thesis outline

Here is an outline of the remainder of this thesis, along with our scientific contributions. The thesis is organized as follows. First of all, Chapter 2 introduces the state-of-the-art in neural Natural Language Processing. We see in particular how Machine Learning training paradigms, Deep Learning architectures, and Language Models have shaped the current developments of the field. Then Chapter 3 focuses on a specific application of self-supervised text-to-text transfer, namely style transfer. We present an approach to addressing the task when detoxification of abusive comments is considered without a parallel dataset. In Chapter 4, we describe the new task of TOXICSPANS detection, which we tackled thanks to a new annotated dataset and both strongly supervised and weakly supervised (cf. Section 2.4.2) systems. Chapter 5 discusses a novel algorithm to compute rating predictions from the sentences written by users who reviewed a set of items. Finally, Chapter 6 concludes the thesis.

1.5 Publications

The research conducted during the Ph.D. resulted in publications listed here in chronological order.

- L. Laugier, J. Pavlopoulos, J. Sorensen and L. Dixon: **Civil Rephrases Of Toxic Texts With Self-Supervised Transformers**. *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2021)* [267]

- J. Pavlopoulos, L. Laugier, J. Sorensen and I. Androutsopoulos: **Semeval-2021 task 5: Toxic spans detection**. *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval 2021)* [381]

- J. Pavlopoulos, L. Laugier, A. Xenos, J. Sorensen and I. Androutsopoulos: **From the Detection of Toxic Spans in Online Discussions to the Analysis of Toxic-to-Civil Transfer**. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL 2022)* [382]

- A. Xenos, J. Pavlopoulos, I. Androutsopoulos, L. Dixon, J. Sorensen, L. Laugier: **Toxicity Detection can be Sensitive to the Conversational Context**. *First Monday, 27(5).* [544]

- L. Laugier, T. Bonald and L. Dixon: **Semantic Encoding of Review Sentences for Memory-Based Recommenders** . *Preprint*

Chapter 2

Neural Natural Language Processing

Chapter 2 abstract

This chapter describes the current state-of-the-art material and methods to address Natural Language Processing with Machine Learning. The subsequent chapters investigate the concepts described in this chapter in novel methods and/or applications. In what follows, we introduce in Section 2.1 motivations behind Natural Language Processing, applications, and why ML is well suited to tackle them. We then present in Section 2.4 the different natures of data processed by ML and the resulting learning paradigms. Section 2.5 provides an overview of modern training and inference techniques implementing the concepts described beforehand. Finally, Section 2.6 summarizes existing neural architectures used as computation models for the previously discussed methods.

2.1 Motivations

2.1.1 The syntactic nature of computers makes them formal machines

Computers (from Latin "*computo*", i.e., calculate, count, sum up) are machines designed to execute programs—software—made of sequences of instructions coded by discrete numbers. These instructions and the data they operate on are expressed in the base-2 numeral system, which optimizes computations on physical hardware. Science and engineering have improved hardware and software capabilities but have remained on the same basis: the steps of algorithms are implemented by formal instructions processed by arithmetic logic units. Computers are perfectly fit to make fast calculations and exceed the arithmetic performance of the best mental calculators. This led early computer scientists to dream of machines able to perform other if not all, cognitive functions observed in animals.

2.1.2 Semantics of natural language is structured but ambiguous

From an animal's perspective, there is no need to consciously formalize instructions before executing some complex but intuitive tasks. In particular, humans rarely need to communicate with formal languages. Natural language derives its richness from its complexity and ambiguity. Famous examples of syntactic ambiguity (e.g. "Time flies like an arrow; fruit flies like a banana" [366]) or lexical ambiguity (e.g. "Will, will Will will Will Will's will?" [166]) are widely used in linguistics to illustrate semantic ambiguity. It is fundamental to human expressions, and that is why it is found throughout the history of literature. Comedy and humor abound in equivocations[1] and word plays such as puns based on homonymy and polysemy. Figures of speech (e.g., metaphors) and rhetorical devices have nourished poetry and argumentations. Tone and register calibrate a speech to specific target audiences or environments. For that matter, context is often crucial to addressing semantic disambiguation. Naturally, specific contexts impose humans to control natural language by simplification rules. For example, the language used by aviators for radio communication is codified to optimize the quality and quantity of information transmitted in a short time and in noisy environments while being fast to learn. Though constructed international auxiliary languages, such as Esperanto, were designed with regular and easy grammar but never replaced complex ordinary local languages. It is worth noting that spoken language can be irregular and fluctuate in time and space while still achieving its goal of connecting agents. Writing, and *a fortiori* printing motivated needs for language codification through relatively well-defined spelling and grammar[2]. All in all, there is a gap between the way computers process information and the mechanism behind natural language.

2.2 Methods

NLP has been studied since the origin of computer science. Computers cannot straightforwardly process natural language due to incompatibility between formal instructions and ambiguity in natural language. Even nowadays, computers remain far from a general understanding of language. Yet, AI strategies are well adapted to address NLP challenges. Further, Alan Turing designed a test to compare artificial and human intelligence, called the imitation game [507]. The test consists of a human evaluator asked to engage in written conversations with another human on the one hand and a machine on the other hand without knowing the nature of its interlocutor. If the human evaluator cannot tell the machine from the human, then the machine is considered to have passed the test by showing human-equivalent intelligent behavior. The discussion about the link between NLP capabilities and strong AI remains open. In its Chinese room argument, Searle [452] argues that an AI could pass the Turing test *via* a pure syntactic method without any actual understanding of natural language.

Early NLP approaches used handwritten-rule-based systems called symbolic NLP that used to be prerequisite

[1] A.k.a. *quae pro quibus* in Latin languages.

[2] This phenomenon is not exclusive to linguistics, as similar needs have existed in music transcription for instance.

to solving higher-level semantic tasks. Inspired by Chomskyan theories of linguistics [77], and grammar teaching in education, symbolic AI combined logic, lexicon, and semantic for NLP. Although this approach based on complex handcrafted feature engineering [258, 365, 576] benefited from total transparency and explainability, it lacked robustness to multifaceted language as well as generalization capabilities. In the 1990s, ML revolutionized NLP thanks to increased computational resources and data as well as the emergence of algorithms learning to solve tasks without explicit specifications of how to do it. As a result, the paradigm shifted toward statistical learning, imitating babies' ability to start language learning well before entering school. Indeed, their brain unconsciously and regularly processes examples of language heard, combined with other sources of signals perceived in their environment.

2.3 Applications

In this research we focused on the written form of language. Although speech processing perfectly tackles research efforts to process natural language, we did not investigate it due to the nature of the data that differs from the discrete nature of text. Speech processing is usually not considered as part of NLP since it often involves speech-to-text conversion, a.k.a. Automatic Speech Recognition (ASR) [124, 293], before processing text (NLP); this pipeline may be followed by a Text-to-Speech (TTS) system. The set of NLP tasks is often decomposed into two complementary subsets: Natural Language Understanding and Natural Language Generation.

Natural Language Understanding The sub-topic of NLP focusing on reading comprehension is called Natural Language Understanding (NLU). Morpho-syntactic and information extraction applications of NLP include token analysis such as lemmatisation [351], sequence labelling—Part-Of-Speech (POS) tagging [390, 360] and Named-Entity Recognition (NER) [505, 204, 324, 260]—or syntactic parsing [4, 68], while NLU is often evaluated with higher-level semantic tasks. Standard NLU benchmarks have been proposed to test systems' abilities to understand complex language features. NLU systems can be evaluated on the hidden benchmarks' test sets, and the leaderboards are publicly available. Meta-benchmarks standardize the evaluation of NLU by collecting human-annotated example pairs on a set of tasks reflecting a deep understanding of language. The General Language Understanding Evaluation (GLUE)[3] [521] and its more challenging successor SuperGLUE[4] [522] respectively aggregate 9 and 8 tasks—one task being common to both benchmarks—to test general language understanding and robustness of systems. Additionally, the SQuAD[5] dataset is also used to test NLU. The tasks aim at probing NLU systems on syntactic correctness, semantic similarity, inference, reasoning, as well as question answering. Table 2.1 summarizes and organizes common NLP tasks according to their types and frameworks.

[3] https://gluebenchmark.com/

[4] https://super.gluebenchmark.com/

[5] https://rajpurkar.github.io/SQuAD-explorer/

Natural Language Generation There is a set of tasks regarded as Natural Language Generation (NLG). These tasks may or may not imply strong NLU abilities; the boundary between NLU and NLG is not well defined. Unconditional Natural Language Generation (UNLG) [114, 170] refers to tasks where NLG has no other constraint than the generated text to be syntactically and semantically correct. If the goal is to generate open-ended text, such as a coherent story, UNLG does not need strong NLU abilities. On the contrary, in prompt completion tasks, UNLG needs some good NLU capabilities for consistency purposes.

When the generated text has to meet additional requirements, NLG is qualified as Conditional Natural Language Generation (CNLG). Controllable generation [232] is an example of CNLG. In particular, if the condition relies on some input text, the task is called a text-to-text task and involves significant NLU of the input text before generating the output text.

Text-to-text systems Text-to-text systems have been driven by research in machine translation [241, 88, 93]. The task represents economic interest for industrial stakeholders, whose research and development laboratories are at the forefront of these technologies. Automatic summarization [117, 428, 570] and conversational models (also known as dialogue systems or chatbots) [590, 203, 583, 579] are the other two main text-to-text tasks. More modestly, attribute transfer in text like style transfer (cf. Chapter 3) is another text-to-text task recently investigated. It should be noted that, more generally, NLP can be involved in multimodal sequence-to-sequence tasks like image captioning [545, 70, 556, 491, 490, 591, 321] (image-to-text) and text-to-image generation [403, 412, 435] with significant results impacting the entire AI field.

NLU and NLG differ in a key aspect: NLU is easier to evaluate than NLG. Automatic metrics are indeed straightforward in NLU: accuracy, F1, and correlations are widely accepted as appropriate evaluation measures of classification tasks. However, standard CNLG metrics [436] (mainly BLEU [374], METEOR [268] and NIST [296] in machine translation and ROUGE [295] in summarization) have been abundantly discussed: despite a relative simplicity, these metrics based on n-gram overlaps cannot fully capture the diversity of language for expressing similar semantics with different words, especially when there are few (if any) ground-truth examples per test example available. For this reason, model-based metrics like RUSE [467], BERTScore [575], BLEURT [454] and NUBIA [227] have been recently proposed to overcome the limitations of word-based metrics: for instance, NLU models are used to compare the semantics of pairs of (predicted, target) texts. Chapter 3 and Section 3.9.2 in particular touch upon issues related to CNLG metrics. For its part, UNLG is commonly evaluated with Perplexity (PPL), an information-theory-based metric measuring how the probability distribution of the generated text is similar to the distribution of text written by humans [55].

All in all, among NLU, UNLG and CNLG, tasks may be tackled by notably different approaches (cf. Section 2.5),

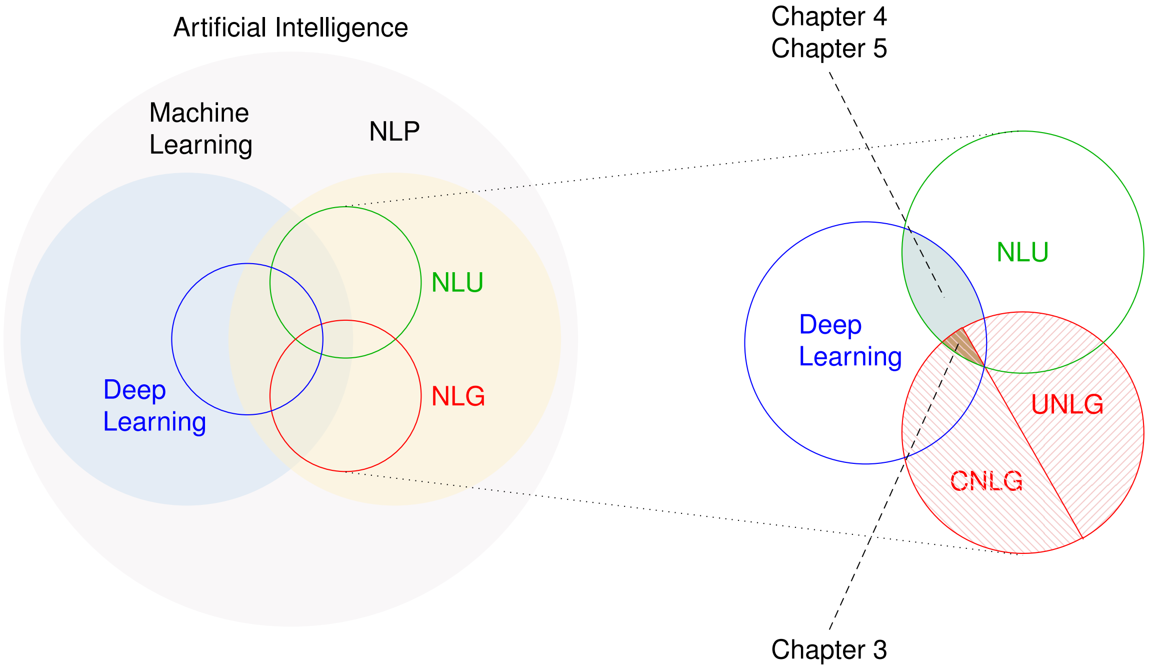


Figure 2.1: Venn diagram of the scientific topics explored in the present research. Chapter 4 and Chapter 5 lie in the blue-green (large) colorized area while Chapter 3 can be categorized in the brown (small) colorized area. Figure best viewed in color.

but there is no type of task easier than the others. Modern NLP research constantly transposes techniques developed for some tasks to others [554]. Moreover, there is a significant research effort going on aiming at unifying the resolution of multi-type NLP tasks (cf. Section 2.5.8). Figure 2.1 summarizes the scientific topics introduced so far, their intersections, and the areas where our research falls within.

As NLP has gained in maturity, we sought to apply it in novel ways to additional real word challenges focused on the online expressions of Internet users. Before the popularisation of the Web, remote interlocutors expressed either through spontaneous oral language (phone calls) or codified written language (epistolary correspondence). Nowadays, every person with Internet access gets the opportunity to share their views through posts, comments, or reviews on Web services that we all have integrated into our daily lives. Through online chat and microblogging, not only have people had direct access to global-scale instant messaging, but new forms of language appeared, halfway between oral and written languages; this raises important research issues from the perspective of NLP. Actually, we will describe specific applications in which modern NLP can impact online human interactions after this chapter discusses methods used in state-of-the-art NLP, summarized in a taxonomy shown in Figure 2.2. Further information can be found in recent high-quality literature reviews [225, 300, 400, 359, 290, 307, 170].

Type	Task	Dataset	Benchmark	Problem framework
NLU	Linguistic acceptability	CoLA [526]	GLUE	Single-sentence classification
	Sentiment analysis	SST-2 [477]	GLUE	Single-sentence classification
	Semantic similarity	MRPC [111]	GLUE	Pairwise sentence classification
		STS-B [62]	GLUE	Pairwise sentence regression
		QQP [207]	GLUE	Pairwise sentence classification
	Natural language inference	MNLI [537, 53]	GLUE	Pairwise sentence classification
		QNLI [409]	GLUE	Pairwise sentence classification
		RTE [94, 21, 147, 32]	GLUE	Pairwise sentence classification
		CB [102]	SuperGLUE	Pairwise sentence classification
	Coreference resolution	WNLI [282]	GLUE	Pairwise sentence classification
		WSC [282]	SuperGLUE	Pairwise sentence classification
	Causal reasoning	COPA [424]	SuperGLUE	Pairwise sentence classification
	Word sense disambiguation	WiC [391]	SuperGLUE	Triple-wise sentence classification
	Question answering	MultiRC [233]	SuperGLUE	Triple-wise (passage - question - answer) text classification
		BoolQ [81]	SuperGLUE	Pairwise (passage - yes/no question) text classification
		ReCoRD [572]	SuperGLUE	Pairwise (passage - list-based query) text classification
		SQuAD [408, 410]	N/A	Extractive QA
U**NLG**	Cloze completion	CBT [183]	N/A	Single-word prompt completion
		LAMBADA [373]	N/A	Single-word prompt completion
	Story generation	HellaSwag [564]	N/A	Multi-word prompt completion
		StoryCloze [350, 464]	N/A	Multi-word prompt completion
C**NLG**	Machine Translation	WMT [39, 115, 40, 41, 42, 43, 44]	N/A	Text-to-text transfer
	Abstractive summarization	CNN/Daily Mail [181, 354, 453]	N/A	Text-to-text transfer
		XSum [357]	N/A	Text-to-text transfer
	Dialog	QuAC [76]	N/A	Text-to-text QA
		CoQA [416]	N/A	Text-to-text QA

Table 2.1: Common NLP downstream tasks and evaluation datasets

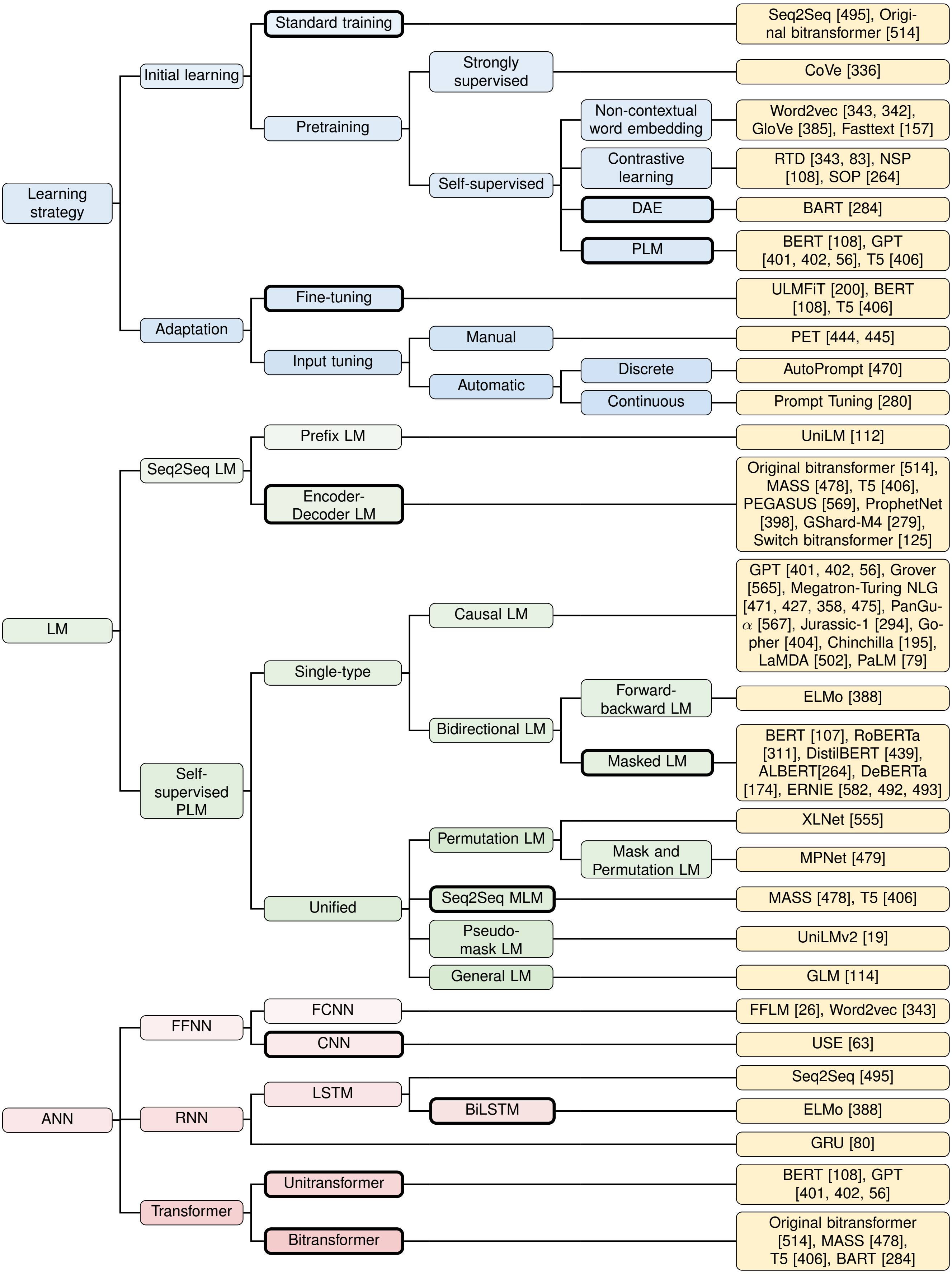

Learning strategy
Initial learning
Standard training
Seq2Seq [495], Original bitransformer [514]
Pretraining
Strongly supervised
CoVe [336]
Self-supervised
Non-contextual word embedding
Word2vec [343, 342], GloVe [385], Fasttext [157]
Contrastive learning
RTD [343, 83], NSP [108], SOP [264]
DAE
BART [284]
PLM
BERT [108], GPT [401, 402, 56], T5 [406]
Adaptation
Fine-tuning
ULMFiT [200], BERT [108], T5 [406]
Input tuning
Manual
PET [444, 445]
Automatic
Discrete
AutoPrompt [470]
Continuous
Prompt Tuning [280]
LM
Seq2Seq LM
Prefix LM
UniLM [112]
Encoder-Decoder LM
Original bitransformer [514], MASS [478], T5 [406], PEGASUS [569], ProphetNet [398], GShard-M4 [279], Switch bitransformer [125]
Self-supervised PLM
Single-type
Causal LM
GPT [401, 402, 56], Grover [565], Megatron-Turing NLG [471, 427, 358, 475], PanGu-α [567], Jurassic-1 [294], Gopher [404], Chinchilla [195], LaMDA [502], PaLM [79]
Bidirectional LM
Forward-backward LM
ELMo [388]
Masked LM
BERT [107], RoBERTa [311], DistilBERT [439], ALBERT[264], DeBERTa [174], ERNIE [582, 492, 493]
Unified
Permutation LM
XLNet [555]
Mask and Permutation LM
MPNet [479]
Seq2Seq MLM
MASS [478], T5 [406]
Pseudo-mask LM
UniLMv2 [19]
General LM
GLM [114]
ANN
FFNN
FCNN
FFLM [26], Word2vec [343]
CNN
USE [63]
RNN
LSTM
Seq2Seq [495]
BiLSTM
ELMo [388]
GRU [80]
Transformer
Unitransformer
BERT [108], GPT [401, 402, 56]
Bitransformer
Original bitransformer [514], MASS [478], T5 [406], BART [284]

Figure 2.2: Taxonomy of methods and architectures used in Neural NLP. Leaves show examples of publications using these techniques. Thick boxes indicate methods leveraged and/or investigated in our research works. Figure best viewed in color.

2.4 Learning with signals from the Experience

2.4.1 The three founding paradigms of Machine Learning

Animals learn with their nervous system processing signals coming from their sensory systems. The quality and quantity of knowledge acquired by learning depend on the nature, length, repetition, and context of the experiences. For example, one can learn a language through immersion, lessons, the use of dictionaries, grammar books, or other media. In the same manner, artificial agents have several ways of experiencing the real world. While animals get signals through their sensory nervous system, artificial agents process information in various forms. Learning algorithms have been traditionally categorized into paradigms depending on the nature of the available signals.

On the one hand, the experience of dataset-based systems is made of a set of observed examples. When examples are only composed of a collection of features represented by x_i, the method is known as "unsupervised" and consists in modeling relevant properties of the data-generating distribution $P(X)$. If, in addition to features in x_i, an example contains an associated target y_i, then learning becomes "supervised" as when a teacher reveals the exact answer to a question. In supervised learning, the goal is to approximate the distribution $P(Y|X)$ of targets y_i given the inputs x_i. When targets have been produced by humans, they are referred to as "annotations".

On the other hand, "reinforcement learning" is based on direct interactions with an environment. Agents receive a direct reward or penalty for an action taken in the environment. Even though reinforcement learning principles have been explored in NLP [547, 151, 319, 539, 438, 510, 317], we did not investigate it during the present research.

Interestingly, current ML may combine ideas from different paradigms. Weakly supervised learning comprises dataset-based experiences in between unsupervised and supervised learning, the latter being sometimes called "strongly supervised learning" to avoid confusion.

2.4.2 Limitations of strong supervision spur weak supervision

As strong supervision [245] requires a large amount of well-annotated examples, it requires solid data-labeling processes. Therefore, softer labeling schemes have been studied in order to relax these constraints. Enormous quantities of high-quality labeled examples are often expensive in terms of money and time to collect as it requires humans to produce individual annotations. Yet, it is sometimes possible to trade off quality annotation for quantity with what is commonly referred to as weak supervision[6]. Following Zhou [592], we attempt dataset-based learning taxonomy displayed on two axis in Figure 2.3. In the real world, supervision can be inaccurate, inexact, or incomplete. The following paragraphs introduce weakly supervised datasets as we go down the rabbit hole of relaxed supervision constraints.

[6] Semi-supervised learning is also employed in the literature but often denotes a subcategory of weak supervision introduced later: incomplete supervision.

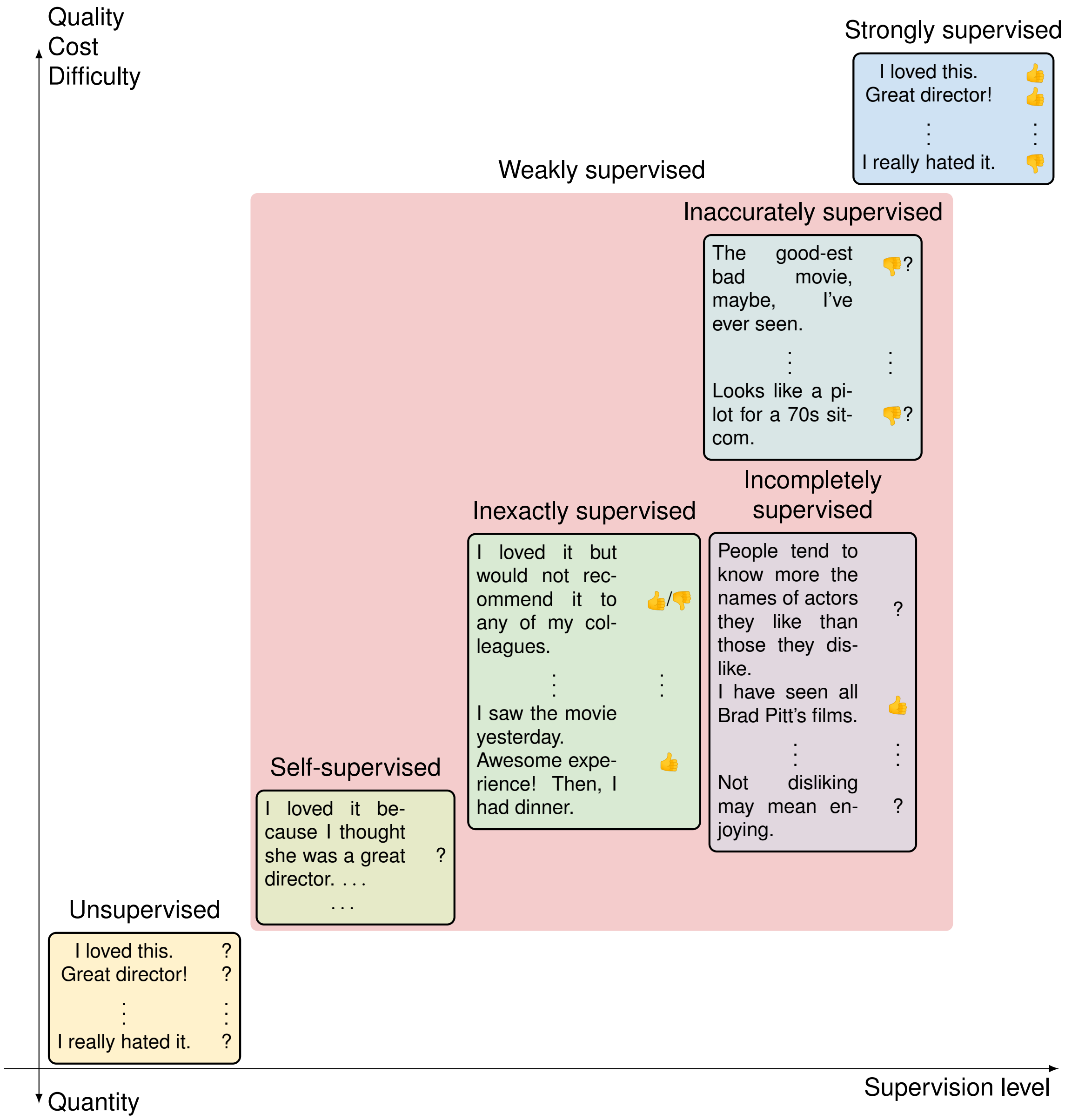

Quality
Cost
Difficulty
Strongly supervised
I loved this. 👍
Great director! 👍
I really hated it. 👎
Weakly supervised
Inaccurately supervised
The good-est bad movie, maybe, I've ever seen. 👎?
Looks like a pilot for a 70s sitcom. 👎?
Incompletely supervised
People tend to know more the names of actors they like than those they dislike. ?
I have seen all Brad Pitt's films. 👍
Not disliking may mean enjoying. ?
Inexactly supervised
I loved it but would not recommend it to any of my colleagues. 👍/👎
I saw the movie yesterday. Awesome experience! Then, I had dinner. 👍
Self-supervised
I loved it because I thought she was a great director. . . . ?
. . .
Unsupervised
I loved this. ?
Great director! ?
I really hated it. ?
Quantity
Supervision level

Figure 2.3: 2D-plot of a taxonomy of dataset-based learning depending on the supervision level. Chapter 3 and Chapter 4 use concepts that can be categorized as weak supervision. Figure best viewed in color.

Inaccurate supervision Targets are collected either automatically or manually. On the one hand, examples (x_i, y_i) can be pre-processed automatically from existing databases: input text may be classified according to their metadata like their source, tags, or ratings. Similarly, parallel corpora of translated legislation are used in machine translation. While offering good scalability characteristics, automatic pre-processing is prone to errors. On the other hand, manual annotation requires humans to examine inputs before carefully assigning some label(s). Though human experts often yield high-quality annotations, scaling expert annotation is expensive. That is why datasets are often labeled by crowdworkers through crowdsourcing platforms such as Appen[7] or Amazon Mechanical Turk[8]. As crowdworkers are paid pro rata with the number of examples annotated, their incentives lean towards quantity rather than quality. Furthermore, ambiguity, context [544] or audience dependency may make some examples subject to subtle interpretation and misunderstood by crowdworkers if their reference frame differs too much from that of the target audience. For example, pejorative American idioms “Monday-morning quarterback”[9] and “carpetbagger”[10] may be confusing for a native English-speaking crowdworker with Indian cultural background. All in all, noise in automatic or manual targets may make supervision inaccurate.

Inexact supervision Properties of language can appear at different granularities: subwords, words, phrases, clauses, sentences, paragraphs, etc. Splitting text into independent parts is a complex task due to semantic consistency that can be long-term. Therefore, an input may contain more text than the subset of text (called “span”) actually responsible for explaining its labeling. For instance, datasets made of review-rating pairs [175, 222] may contain long reviews stating positive and negative aspects but associated with one single positive or overall negative rating of the reviewed item. Finding local characteristics in long texts is at the heart of NLP research on interpretability and explainability [419, 288, 355, 340, 59, 487] and is at the core of Chapter 4.

Incomplete supervision Some ML approaches can leverage a set of labeled and unlabeled examples. In particular, data augmentation [123] is a trick used to increase the size of the training dataset automatically. In NLP, data augmentation can be materialized by synonym replacement [527] or cyclic translation [486] (cf. Chapter 3). Active [459] and semi-supervised [593] learning are well-known approaches described by [592] though barely applied to NLP. Actually, as it exploits both unlabeled and labeled data, incomplete supervision echoes a major breakthrough in recent NLP training schemes, called self-supervised learning [309].

Self-supervision Open challenges in AI that are keys to animal intelligence are the development of commonsense knowledge [106] and transfer of learning [386]. ML algorithms have proved quite efficient at overfitting individual low-level tasks, provided they have enough various labeled sample data. Yet, strong supervision hinders generalization

[7] `https://appen.com/`

[8] `https://www.mturk.com/`

[9] Term referring to American football and meaning "someone who criticizes from hindsight.”[535]

[10] Term referring the American Civil War and meaning “one who comes to a place or organization with which they have no previous connection with the sole or primary aim of personal gain, especially political or financial gain.” [536]

because it cannot guarantee pitfalls that would not be covered by the training data. Indeed, collecting datasets of labeled data large and diverse enough to cover the space of all possibilities is illusory and does not correspond to processes observed in biological learning.

2.4.3 Supervisory signals from unlabeled data: a goldmine for Natural Language Processing?

Even if the terminology of dataset-based ML is open to debate and may seem idle at first, we think it raises fundamental questions on how ML can improve from a biomimetics perspective. Historically, the presence of targets differentiates supervised learning from unsupervised learning. The supervised signal was considered to come exclusively from the target, and unsupervised algorithms were designed to discover structures or patterns in the input features, e.g., anomaly detection, correlations (PCA [135, 199, 218]), cluster and community detection (k-means [327, 313, 133]). Nevertheless, some untagged data sources like language hold far more “supervisatory signal” [273] than others (like physical measurement in time or space). Thus, the diptych labeled / unlabeled learning is a proposition[11] that could clarify the origin of the supervisory signal. Take the following sentences: $s_{\text{train}} =$ ``People often stop watching a movie that they consider bad, before it ends.'' and $s_{\text{test}} =$ ``I stopped watching this movie after only 1 hour''. Now, consider the task of sentiment analysis [371, 477, 571], where the system has access to unlabeled training sentences including s_{train} and is queried with inferring the sentiment in s_{test}. Even without direct access to external labels, self-supervised learning introduces objective functions to capture the supervisory structure found in the training data (which may implicitly or explicitly contain labels, cf. Section 2.5.3)[12]. Besides, it has recently played a crucial role in the development of AI systems as described in Section 2.5. Self-supervision is even qualified as “the next frontier of AI” by LeCun and Ishan [273], revealing the “dark matter of intelligence”.

In conclusion, we saw that ML could process signals in various ways. The original supervised/unsupervised/reinforcement learning paradigms have been extended to a richer set of in-between setups, resembling real-world conditions. The following chapters derive examples of weakly supervised systems attempting to solve practical problems raised by online interactions.

[11] https://twitter.com/yoavgo/status/1489364641000181774

[12] Note that strong supervision and self-supervision are equivalent from the point of view of UNLG.

2.5 Training and inference

2.5.1 Supervised training

The traditional approach of supervised learning is well described with probabilistic learning. The true conditional label distribution $P(Y|X)$ (derived from the data generating distributions $P(Y,X)$ and $P(X)$) is approximated by a parametric family of model distributions $P_\theta(Y|X)$. Learning consists in tuning the weight θ by optimizing the value of a loss (or objective) function $\mathcal{L}$ so that the optimal model distribution $P_{\hat{\theta}}(Y|X)$ gets "closest" to the empirical conditional label distribution $\hat{P}(Y|X)$, defined by the training set. The maximum likelihood estimator [131] with regard to the observed examples (x_i, y_i) is often considered a good choice for $\hat{\theta}$ because of its statistical properties.

Beyond ambiguity and context dependency (cf. Section 2.1.2), natural language carries another important specificity: it is made of variable length sequences of discrete tokens. In NLU, sequentiality appears in the input text x modeled as sequences of tokens $\left(x^1, x^2, \ldots, x^m\right)$. In NLG, the output is a variable-length sequence of tokens $\left(y^1, y^2, \ldots, y^p\right)$. NLG is often viewed as a multiclass classification task with an infinite number of classes: the possible classes are all the tuples made of tokens from the vocabulary V.

The discrete nature of language makes it particularly suited to supervised probabilistic classification, whose goal is to learn a parametric function f_θ—called the model—that predicts class $\hat{y}$ given an input x. For instance, f_θ may correspond to the optimal decision rule, i.e. $\hat{y} = f_\theta(x) = \arg\max_y P(y|x;\theta)$ [35]. In classification, maximum likelihood (L) estimation is equivalent to minimizing the cross-entropy loss (H)—a function grounded in information theory [90]—or, alternatively, the statistical distance known as the Kullback–Leibler divergence D_{KL} [252]. The optimization problem can be formalized in the equivalent formula given in Equation (2.1).

$$\begin{aligned}
\hat{\theta} &= \arg\min_\theta \sum_i \mathcal{L}(x_i, y_i; \theta) = \arg\min_\theta H\left(\hat{P}(Y|X), P_\theta(Y|X)\right) \triangleq \arg\min_\theta \mathbb{E}_{\hat{P}(Y|X)}\left[-\log\left(P_\theta(Y|X)\right)\right] \\
&= \arg\min_\theta D_{KL}\left(\hat{P}(Y|X)\,||P_\theta(Y|X)\right) \triangleq \arg\min_\theta \mathbb{E}_{\hat{P}(Y|X)}\left[\log\left(\frac{\hat{P}(Y|X)}{P_\theta(Y|X)}\right)\right] \\
&= \arg\max_\theta L\left(\theta|\left(x_i, y_i\right)\right) \triangleq \arg\max_\theta P\left(\left(y_i\right)_i | \left(x_i\right)_i; \theta\right) = \arg\max_\theta \prod_i P\left(y_i|x_i;\theta\right)
\end{aligned} \tag{2.1}$$

2.5.2 Language Representation Learning

Representation Learning [28] is an important ML concept at the core of the conception of neural architectures (cf. Section 2.6). Neural classifiers models successively apply linear and non-linear transformations to input features in order to make data points easily separable. Transformations generate internal representations of inputs and outputs that may acquire relevant algebraic properties. For instance, linear separation in Euclidean space is particularly intuitive (cf. Figure 2.4).

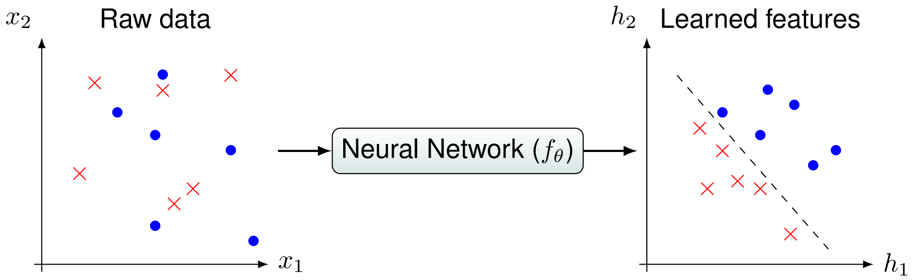


Figure 2.4: 2D-illustration of the goal behind learning “good” representations with an Artificial Neural Network. Red crosses and blue dots are data points belonging to one class or another. Representation (or feature) learning aims at making these classes easily distinguishable in a latent space.

In NLP, we are interested in relevant numerical representations of text [186, 26]. A common approach to language Representation Learning is text embedding, in which we aim at embedding chunks of text in some vector space where semantic and syntactic properties are appropriately distributed. The granularity of the distributed representation may vary [271, 239, 338] but the leading approach consists in embedding each word (or subwords [456, 251, 108]) with a parametric embedding matrix θ_{Emb} of size $d_{\text{Emb}} \times |V|$. The embedding matrix can be used as standalone for text analysis. Word embedding actually predates the neural NLP era. Still, an embedding matrix often constitutes the first—and sometimes only—layer of a neural model.

2.5.3 Self-supervised Representation Learning

The goal of self-supervised Representation Learning is to automatically extract general syntactic and semantic knowledge (like common sense) from tremendous quantities of unlabeled data. NLP has the characteristics of having easy access to a huge amount of raw, rich, and various datasets online. Unlabeled text of diverse nature and quality may be present in online encyclopediæ [482] (in 2022, Wikipedia[13] had ~ 29 billion words, ~ 3.9 billion of which are in the English version), news datasets [220] (Google News dataset has ~ 100 billion words), books corpora [596] (~ 800 million words in the Toronto Books Corpus), web crawl (scrape) corpora [406, 402, 56, 79] (Common Crawl[14] has hundreds of billions of words) or social media conversations[15][16] [2] (LaMDA [502] was trained on ~ 1.56 trillion words from public dialogs).

Knowledge may be stored in explicit or implicit forms. For instance, the sentence `“1961 was the last year in which the majority of Hollywood films were released in black and white.”` [533] explicitly states a historical fact related to cinematography. It also participates in the commonsense knowledge that black and white movies preceded color movies, and the shift happened in the mid-twentieth century. More implicitly, the sentence `“In December 2015, Moore announced his support for Vermont Senator Bernie Sanders in the 2016`

[13] https://en.wikipedia.org/
[14] https://commoncrawl.org/
[15] https://twitter.com/
[16] https://www.reddit.com/

`United States presidential election.`" [534] refers to American politics to indicate the political views of director Michael Moore. Further text detailing Senator Bernie Sanders' political positions may add valuable background to understanding Michael Moore's work. In the following sections, a corpus used for self-supervision will be represented by a sequence of unlabeled tokens $\left(w^1, w^2, \ldots, w^q\right)$ with $q > 10^9$ in practice.

Non-contextual self-supervised word embeddings Implementations of self-supervised Representation Learning in NLP are based on the distributional hypothesis proposed by linguists: words are context-dependent. This principle has been summarized by "You shall know a word by the company it keeps" in Firth's work on collocational meaning [164]. First uses of unlabeled text to learn word representations were introduced by Collobert and Weston [85], Turian et al. [506] and Collobert et al. [86]. It became dominant in NLP when latent semantic analysis revealed powerful properties learned by the self-supervised Word2vec [343, 342, 429, 163], GloVe [385] and Fasttext [157]. The key idea was to design a loss function forcing the encoding of semantic similarity between two words. Cosine similarity between the word embeddings was chosen as a proxy for semantic similarity.

Word2vec In NLP, self-supervision is put in practice with appropriate tasks designed to learn semantic similarity from the context in which words occur. Mikolov et al. [343] proposed two efficient objectives to make a model learn word embeddings from the contexts they appear in: continuous bag-of-words and skip-gram. In both objectives, training consists in the classification task of maximizing the likelihood $P\left(y|x;\theta\right)$ where x is the input text and y is the pseudo-label text. The principle consists in making a fixed-size window slide along the entire corpus $\left(w^1, w^2, \ldots, w^q\right)$ and using the tokens in the window to build x and y. Let $2 \times c + 1$ be the window size. The window is made of a target token w^t at its center and a set of context tokens $\left\{w^{t-c}, w^{t-c+1}, \ldots, w^{t-2}, w^{t-1}, w^{t+1}, w^{t+2}, \ldots, w^{t+c-1}, w^{t+c}\right\}$.

Continuous bag of words In the continuous bag-of-word (CBOW) version, the input is made of the context tokens, and the pseudo-label is the target token.

Skip-gram Conversely, skip-gram considers the target token to be the input and the context tokens to be the pseudo-label.

A word of caution: even though here the self-supervised training considers words in their context, the learned embeddings are called "non-contextual" or "static" because they are stored in a fixed $d_{\text{Emb}} \times |V|$-matrix, later used to embed words independently of the context they appear in. Incidentally, cosine similarity has also been applied later for sentence similarity [450, 64] and we leverage "sentence embeddings" in the subsequent chapters (cf. Chapter 3 and Chapter 5).

2.5.4 Stack structures

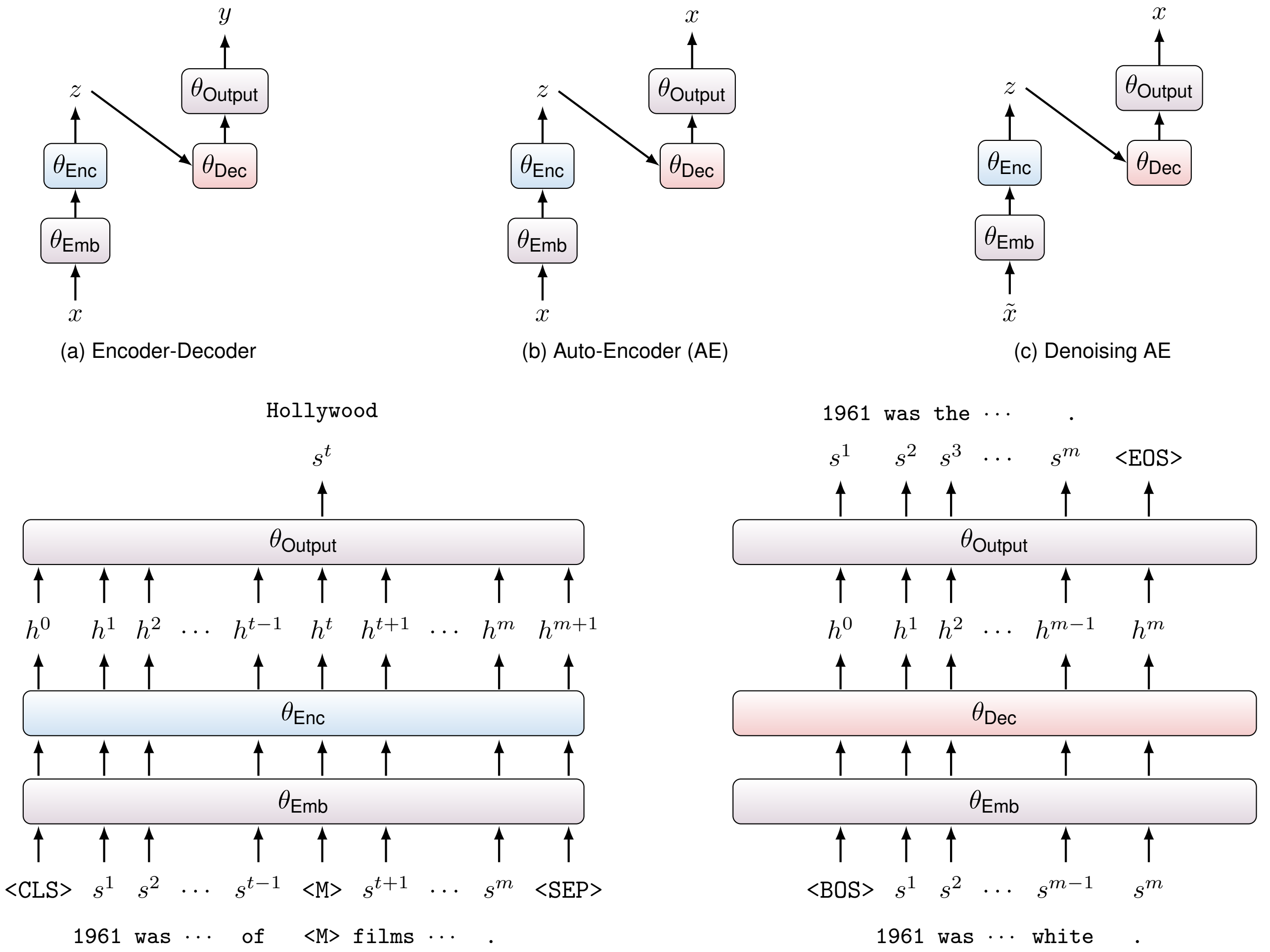
In neural NLP, the backbone structures of models can be classified according to the type of task they are mainly designed for (cf. Section 2.3). Layers fulfilling a specific function are grouped into what is called a stack. Neural NLP developments have juggled the single-stack and multi-stack structures we present hereunder.

Single-stack structures A system mainly designed for NLU is called an encoder as its function is to encode discrete representation of text x into a latent representation $z = f_{\theta_{\text{Enc}}}(x)$. The goal is to encode the semantics of the input in a continuous distributed space. z can be a single vector representing the entire input $x = (x^1, x^2, \ldots, x^m)$ or a sequence of m token representations. BERT [108], its derivatives [311, 439, 264, 174], XLNet [555], ELECTRA [83] and ERNIE [582, 492, 493] are examples of modern NLU systems modeled with encoders.

NLG-oriented (and in particular UNLG-designed) systems are often modeled with a stack called decoder, whose role is to decode internal dense representations z in discrete sequences of tokens y. Decoders have been used to model NLG systems such as GPT [401, 402, 56], Grover [565], Megatron-Turing NLG [471, 427, 358, 475], PanGu-α [567], Jurassic-1 [294], Gopher [404], Chinchilla [195], LaMDA [502] and PaLM [79]. If combined with an encoder, the decoder is trained to maximize the likelihood $P(y|z;\theta)$. In language modeling (cf. Section 2.5.7), generating tokens of $\hat{y}$ is then realized via sampling the model distribution according to some decision rule (cf. Section 2.5.7). What is more, since natural language has properties of symbolic time series [298], NLG is aptly addressed by auto-regressive decoders. Auto-regressive generation consists in generating one token at a time with a feedback loop at each time step. The iterative process stops when a special End-Of-Sequence token `<EOS>` is reached.

There are two key properties that differentiate the design of encoders and decoders. First, encoders have access to the entire input text x. Therefore, they a priori know the sequence length, while decoders cannot know in advance the length of the sequence y they will generate, especially when generation is auto-regressive. The sequence length variability is a concept characterizing NLP as opposed to computer vision, where images have a fixed set of pixels. Second, by having initial access to the entire input x, encoders can process the entire sequence in any order. On the contrary, auto-regressive decoders may only process information coming from previous tokens.

Multi-stack structures In addition to single-stack structures, multi-stack structures have been proposed. The most common multi-stack structure is the double stack. It is made of an encoder and a decoder (cf. Figure 2.5). Encoder-decoders are general structures proposed in sequence-to-sequence learning [495]. In particular, they are adapted to text-to-text tasks, where learning a suitable representation z in a latent space is sought-after. For example, disentanglement can be a good latent space property when we aim at transferring some text attribute from one subspace to another. Actually, latent representation learned with strongly supervised learning was the main idea of Sutskever et al. [495]. More recently, encoder-decoder structures have been employed by the bitransformer [514], MASS [478], T5 [406], PEGASUS [569], BART [284], ProphetNet [398], GShard-M4 [279] and the switch



y
x
z
θ_{Output}
θ_{Enc}
θ_{Dec}
θ_{Emb}
$\tilde{x}$
Hollywood
s^t
h^0 h^1 h^2 $\cdots$ h^{t-1} h^t h^{t+1} $\cdots$ h^m h^{m+1}
<CLS> s^1 s^2 $\cdots$ s^{t-1} <M> s^{t+1} $\cdots$ s^m <SEP>
1961 was ⋯ of <M> films ⋯ .
1961 was the ⋯ .
s^1 s^2 s^3 $\cdots$ s^m <EOS>
h^0 h^1 h^2 $\cdots$ h^{m-1} h^m
<BOS> s^1 s^2 $\cdots$ s^{m-1} s^m
1961 was ⋯ white .

(a) Encoder-Decoder

(b) Auto-Encoder (AE)

(c) Denoising AE

(d) Masked Language Model [108]

(e) Causal Language Model (trained with teacher forcing)

Figure 2.5: Structures of neural models found in the literature. h_i is the contextualized representation corresponding to the i$^{\text{th}}$ input token.

bitransformer [125].

The number, nature, and assembly of stacks in neural NLP is only limited by computational resources, optimization issues, and the imagination of their developers. For example, Baziotis et al. [22] proposed to combine two encoder-decoders, i.e. a quadruple-stack structure, for self-supervised abstractive summarization.

Note that the structure-function mapping proposed above is flexible. Indeed, several works have used single-stack structures for text-to-text [56, 112, 19, 114] while others introduced encoder-decoders for a set of applications including NLU-only tasks [406].

Strongly supervised NLP models learn input representations for the task they are trained on. Yet, since neural models tend to overfit when not appropriately regularized, representations may be too task-specific. In contrast, we may be looking for more general representations to strengthen transfer learning [483, 503, 54, 504, 370, 9, 10, 97, 407, 269, 121, 47, 141] properties of models, inspired by human abilities to solve new tasks by leveraging knowledge acquired on related problems seen in the past. In fact, pretraining a model on a self-supervised task is an efficient (and perhaps sufficient) approach to teach general knowledge that will be used to solve downstream tasks [154].

2.5.5 Self-supervised Encoder-Decoders

Auto-encoders Auto-Encoder (AE) [557, 51, 247, 302, 185, 188, 303, 154, 144] are models traditionally used for learning representations (a.k.a. feature learning) from unlabeled data. The hack is to train an encoder-decoder, like in the strong supervised setup, except that the label y *is* the input x (cf. Figure 2.5) or a subset of x's features. Auto-Encoders illustrate the terminology point discussed in Section 2.4.2. Indeed, Auto-Encoders deal with raw unlabeled input text but are trained with a pseudo-label built automatically from the input. Neither unsupervised learning nor strong supervision faithfully correspond to this situation where the supervisory signal comes from the input *itself*.

If unconstrained, an Auto-Encoder could simply learn the identity function, which is useless. Therefore, training and modeling techniques have been developed to force the decoder to reconstruct the input x from a compressed salient encoding. Several such means have been introduced such as Variational Auto-Encoders (VAE) [237, 238] and concrete Auto-Encoders [1] but the most efficient ones for NLP are specific regularized auto-encoders, called the Denoising Auto-Encoders (DAE).

Denoising Auto-Encoders Bengio et al. [29] and Alain and Bengio [6] showed that training an Auto-Encoder to reconstruct a corrupted version $\tilde{x}$ of the input x makes the model learn rich representations for x [517, 518]. Corruption alters the input data by sampling $\tilde{x}$ from a noise distribution given x, $C\left(\tilde{X}|x\right)$. Then, learning $P\left(x|\tilde{x};\theta\right)$ teaches the model to capture important information from x while ignoring the noise (cf. Figure 2.5). Glorot et al. [148] proved the relevancy of DAEs for NLP by using them in sentiment analysis. Common text corrupting strategies include single-token or multi-token span replacement with a special mask `<M>` token [108, 311, 264, 174, 582, 492, 493, 284, 83, 478, 569, 88, 112] (cf. Section 2.5.8), a sequence of sentinel tokens `<X>`, `<Y>`, ... [406] or tokens randomly sampled from the vocabulary [108]. Other noising functions have been introduced such as random token deletions (dropping) [284], permutations (shuffling) [555, 284, 19], capitalization [312], truncations or else rotations. In NLP, training with a Denoising Auto-Encoder is sometimes referred to as Full Text Reconstruction (FTR), as opposed to Corrupted Text Reconstruction (CTR, cf. Section 2.5.8) [307].

Denoising Auto-Encoders can be trained to learn transferable representations of the input as well as the output in generation tasks. Nevertheless, without further and stronger supervisory signals, they cannot a priori solve downstream tasks. As a matter of fact, the learned representations may be too general with too much information lost by compression. Downstream tasks may only need a subset of the semantics included in the input or output text, hence the need of “domain adaptation”. Thus, the complete training of a model can be decomposed into two consecutive optimization phases: pretraining a subset of (or all) parameters with self-supervision before fine-tuning the entire model on strongly supervised downstream task(s) [184]. Note that there is a current research trend trying to get rid of the fine-tuning hypothesis, which will be detailed in Section 2.5.9.

2.5.6 The pretraining/fine-tuning pipeline

Pretraining Even if it is possible to pretrain NLP models with strong supervision [336], self-supervised pretraining has prevailed in neural NLP for the reasons given in Section 2.4.3. Non-contextual self-supervised word embeddings (cf. Section 2.5.3) have been used as representation features in NLP models. Whether pretrained or not, word embedding often makes up the first layer of a neural model; Turian et al. [506] showed the benefits of pretrained word embeddings over random initialization.

Pretrained model Additional self-supervised pretraining tasks have been proposed to pretrain the entire model rather than the initial layer [343, 385, 388]. Dai and Le [95], Liu et al. [306] and Howard and Ruder [200] were among the first to impulse the now widely-used trend of full-model pretraining in NLP [108, 311, 439, 264, 174, 555, 83, 582, 492, 493, 401, 402, 56, 565, 471, 427, 358, 475, 567, 294, 404, 195, 502, 79, 478, 406, 569, 284, 398, 279, 125, 56, 112, 19, 114], in order to increase transfer learning abilities. Initially, the original denoising auto-encoding objective (cf.Section 2.5.5) was used for self-supervised pretraining. Alternatively, contrastive learning [443] is another approach to self-supervised pretraining. This classification task consists of learning which pairs of text are comparable. Examples of contrastive learning include Negative Sampling [343], Replaced Token Detection (RTD) [83], Next Sentence Prediction (NSP) [108] and Sentence Order Prediction (SOP) [264]. Yet, the dominant approach to self-supervised pretraining heavily relies on Language Models, introduced in Section 2.5.7.

Fine-tuning After pretraining, a second phase is often needed to adapt and transfer the knowledge acquired at pretraining toward more restrained problem(s). Strong supervision on few[17] labeled examples are used to *fine-tune* (update) the model's parameters with an objective focused on one or several specific downstream task(s). Despite significantly improving the experimental results on downstream tasks, there is no clear theoretical grounding regarding pretraining benefits. Goodfellow et al. [154] puts forward two arguments. First, in the optimization perspective, models benefit from parameter initialization in the appropriate region [120]. The second reason is probability-grounded: knowledge of the input and/or output distributions would facilitate the process of learning the right mapping from inputs to outputs; a data-driven *temet nosce*[18] maxim so to speak.

As we have seen, self-supervision is implemented through pretraining in modern neural NLP. There exist several loss functions to pretrain a model, but a specific family of objectives has prevailed in modern ML-based NLP: Language Models.

[17] Compared to the pretraining dataset size

[18] Latin for “know thyself”

2.5.7 Language Models

Standard properties of modern Language Models A Language Model (LM) is a probability distribution over variable-length sequences of tokens belonging to a pre-defined vocabulary V (i.e., symbols pertaining to a language). A LM parametrized by θ assigns probability $P(s|\theta)$ to all sequences $s = \left(s^1, s^2, \ldots, s^m\right) \in V^m, \forall m \in \mathbb{N}$. LMs had initially been introduced in the context of automatic speech recognition [212] and have been applied to NLP [26]. In a probabilistic approach, language modeling is the natural way to formulate NLG. LMs not only relate to the well-known classification setup but also enable direct and simple learning of both the syntactic complexity of language and the semantic knowledge phrased by language [220]. For instance, assigning high probability to the English sentence ``People often stop watching a movie that they consider bad, before it ends.'' contributes to encoding that a well-formed English sentence starts with a capital letter and ends with a period and that the word ``bad'' has more probability of directly following the word ``consider'' than the word ``they''. Additionally, a system with some understanding of natural language (vocabulary, grammar, and semantics) may learn a commonsense fact about logical human behavior with this sentence.

Recent ML for NLP has substantially benefited from specific models trained to compute LMs, to the point where models and the distribution they predict are often identified. Indeed, neural architectures like transformers (cf. Section 2.6) and the massive increase in models' capacities improved the expressiveness of LMs.

Neural Language Models Early implementations of LM computing relied upon word-count-based methods. A n-gram LM is a statistical LM [462, 346] computing the frequency of markovian sequences of n tokens—called n-grams[19]—appearing in text [329, 461, 155]. When seen as weighted finite automata, an n-gram LM is an instance of statistical NLP systems at the intersection of formal language theory and linguistics. Because statistical LMs use discrete representations of tokens, they face two issues: sparsity and complexity. Sparsity implies that n-grams not appearing in the training set have null probability, hence hindering generalization. Besides, the complexity of statistical LM is in $O\left(e^n\right)$ with respect to the the context length n, limiting the Markov assumption to low n^{th} order in practical cases. Both limitations are addressed by continuous space Language Models [449]. In this approach, widely used nowadays, neural networks (cf. Section 2.6) are trained to compute LMs. Neural LMs (NLMs) can be characterized by the specific backbone architecture used, e.g., Feedforward NLM, Recurrent NLM, Transformer-based NLM, etc. It is common for NLMs to learn an internal representation of individual tokens, or entire sequences, with dense and low-dimensional vectors. In general, each token s^t—s^t being an input token in NLU and an output token in NLG—is represented at each layer i by a vector h_i^t called the hidden representation of s^t or the hidden state of the model at time step t. The output layer dealing with projecting the last hidden representations h^t back to logits in the discrete vocabulary space is often called the "LM Head".

[19] n-grams are particular skip-grams, seen in Section 2.5.3

Auto-regressive Language Models LMs can be auto-regressive (cf. Section2.5.4). At each generation step t, the model has access to representations of the previously generated token(s) $\hat{s}^{1:t-1} = \left(\hat{s}^1, \hat{s}^2, \dots, \hat{s}^{t-1}\right)$. Auto-regressive LMs are computed with the chain rule (cf. Equation (2.2)).

$$P(s|\theta) = P\left(s^1, s^2, \dots, s^m|\theta\right) = \prod_{t=1}^{m} P\left(s^t|\hat{s}^1, \hat{s}^2, \dots, \hat{s}^{t-1}; \theta\right) \tag{2.2}$$

Search techniques for Generation There are various strategies to generate tokens from an auto-regressive LM at inference time. They can be grasped by their quality-complexity trade-off. Let T be the maximum size of a generated sequence. In theory, T can be infinite, but in practice, $T \in \mathbb{N}$.

Exhaustive search Under the assumption of a good auto-regressive LM, searching the sequence with maximum likelihood requires a brute-force approach. Exhaustive search consists in exploring the entire distribution of all possible sequences made of tokens from the vocabulary V. Equation (2.3) formalizes the exhaustive search.

$$\hat{s} = \underset{s \in \bigcup_{k=1}^{T} V^k}{\arg\max}\, P\left(s|\theta\right) \tag{2.3}$$

Exhaustive search has exponential complexity as its running time is in $\Theta\left(|V|^T\right)$. In NLP, $|V| \sim 10^4$ so generating long and/or many sequences quickly becomes intractable.

Greedy search The auto-regressive property lets us sacrifice some quality for major savings in execution time. With greedy search, the generated sequence $\hat{s} = \left(\hat{s}^1, \hat{s}^2, \dots, \hat{s}^T\right)$ is generated one step at a time. In practice, tokens generated after an End-Of-Sequence token (<EOS>) are ignored (or the process stops before they are generated). At each time step, greedy search yields $\hat{s}^t = \arg\max_{s^t \in V} P\left(S^t = s^t|\hat{s}^1, \hat{s}^2, \dots, \hat{s}^{t-1}; \theta\right)$. Because it simply considers the distribution on the vocabulary V at each time step, greedy search takes linear execution time, in $\Theta\left(|V| \cdot T\right)$.

Beam search A trade-off has been proposed to control quality and complexity with a parameter B called the beam width. In beam search [171], a set $\mathcal{H}_t$ of the B most probable sequences from $\mathcal{H}_{t-1}$ concatenated with any token $w \in V$ is iteratively maintained at each time step t, as described by equations (2.4).

$$
\begin{aligned}
\mathcal{H}_0 &= \emptyset \\
\mathcal{H}_t &= \underset{\substack{\mathcal{H}\subset\tilde{\mathcal{H}}_t: \\ |\mathcal{H}|=B}}{\arg\max} \sum_{\hat{s}^{1:t}\in\mathcal{H}} P_{S^{1:t}\sim\mathcal{H}}\left(S^{1:t}=\hat{s}^{1:t}|\theta\right) \\
\text{where } \tilde{\mathcal{H}}_t &= \bigcup_{\substack{\left(\hat{s}^{1:t-1},w\right) \\ \in\mathcal{H}_{t-1}\times V}} \hat{s}^{1:t-1} \frown (w) \\
\text{Finally } \hat{s} &= \underset{\hat{s}^{1:T}\in\mathcal{H}_T}{\arg\max} P_{S^{1:T}\sim\mathcal{H}}\left(S^{1:T}=\hat{s}^{1:T}|\theta\right)
\end{aligned}
\tag{2.4}
$$

Beam search has a linear run time in $\Theta\left(B\cdot|V|\cdot T\right)$ with a constant factor equal to B. Beam search is widely used in NLG, especially in machine translation, and plays a role in increasing the diversity of generated text since it yields B alternative generated text. The beam width B gives direct control over the quality-complexity trade-off: $B=1$ is greedy search, while incrementing B exponentially from V to V^T at each time step corresponds to exhaustive search.

Teacher forcing when training auto-regressive Language Models Auto-regressive LMs use a feedback loop at inference time. The same process can be used at train time, but an alternative scheme is often preferred for its time efficiency and robustness. When making a prediction, teacher forcing [538] gives the model access to information not accessible at inference time: the ground-truth target. Instead of predicting the next token based on the previously generated tokens ($P\left(s^t|\hat{s}^1,\hat{s}^2,\ldots,\hat{s}^{t-1};\theta\right)$), teacher forcing computes the prediction of the next token based on the past ground-truth tokens: $P\left(s^t|s^1,s^2,\ldots,s^{t-1};\theta\right)$. The benefits are twofold and illustrated by the following situation where the LM is trained on the sentence ``People often stop watching a movie that they consider bad, before it ends.''. If the training step is early, then free-running generation may produce "random" tokens, like $\hat{s}^1=$ ``are'' and $\hat{s}^2=$ ``she''. The information coming from the model trained to maximize $P\left(S^3=\text{``stop''}|\hat{s}^1=\text{``are''},\hat{s}^2=\text{``she''};\theta\right)$ is much more noisy than when a "teacher" forces the model to maximize $P\left(S^3=\text{``stop''}|s^1=\text{``People''},s^2=\text{``often''};\theta\right)$. Besides, even when the training process is advanced, the signal from $P\left(s|\theta\right)=P\left(s^1,s^2,\ldots,s^m|\theta\right)$ requires to compute the sequence $\left(P\left(s^t|\hat{s}^1,\hat{s}^2,\ldots,\hat{s}^{t-1};\theta\right)\right)_{t\in[1..T]}$ one generation step at a time while $\left(P\left(s^t|s^1,s^2,\ldots,s^{t-1};\theta\right)\right)_{t\in[1..T]}$ can be parallelized and computed in a single step.

Sequence-to-sequence learning Sequence-to-Sequence (Seq2Seq) learning is the set of ML approaches that transforms a sequence into another. Sutskever et al. [495] have proposed a neural approach to text-to-text tasks. This work has gotten major impact since it achieved remarkable results by applying three key principles:

1. Train with end-to-end learning, i.e., models (gradient-based) directly learn from inputs associated with targets
2. Optimize auto-regressive Language Model objectives

3. Model distributions with encoder-decoder neural architectures (cf. Section 2.5.4 and Section 2.6)

Sequence-to-Sequence Language Models A Seq2Seq LM is a LM conditioned by some input text $x = (x^1, x^2, \ldots, x^m)$. A sequence sampled from the distribution is called an output $y = (y^1, y^2, \ldots, y^p)$, and a Seq2Seq LM assigns probability $P(y|x;\theta)$. Sutskever et al. [495] proposed a strongly supervised approach, that is to say maximizing the likelihood $P(y|x;\theta)$ from pairs of input and target texts (x_i, y_i). With auto-regressive generation, a convenient way of triggering the generation of the first token $\hat{y}^1$ given the input x is to condition the generation with an initial Beginning-of-Sequence token $y^0 = \texttt{<BOS>}$. This also applies to teacher forcing, in which case $y^0 = \texttt{<BOS>}$ is prepended to the input y.

Prefix Language Models Prefix LMs are neural Seq2Seq LMs modeled with a single-stack structure [112, 367] (cf. 2.5.4).

Encoder-decoder Language Models A Seq2Seq LM modeled with a double-stack encoder-decoder structure [478, 406, 398, 284] (cf. Section2.5.4) is referred to as an encoder-decoder LM [290].

2.5.8 Self-supervised Pretrained Language Models

LMs fit perfectly self-supervision in NLP. Therefore, they have been extensively applied to pretraining systems designed for UNLG (cf. Section 2.5.8), NLU (cf. Section 2.5.8) and test-to-text transfer (cf. Section 2.5.8). We detail below specific approaches to train LMs with self-supervision.

Causal Language Models As we saw in Section 2.5.7, LMs—and in particular auto-regressive LMs—are well suited to NLG. In addition to strong supervision (cf. Section 2.5.7), LMs can be trained with self-supervision in order to generate text automatically. To this end, a sequence s is here split into two subsequences: a prompt text $x = (\texttt{<BOS>}, s^1, s^2, \ldots, s^k)$ and text to generate $y = \left(s^{k+1}, s^{k+2}, \ldots, s^{k+k'}\right)$. A Causal LM [221] is an auto-regressive NLM maximizing the likelihood $P(y|x,\theta)$. Causal LMs are commonly referred to as standard LM. Contrary to Seq2Seq LMs, causal LMs are fully unidirectional. This means their architectures are designed to prevent hidden representations of tokens from processing information coming from subsequent (“future”) tokens. For this reason, causal LMs are sometimes called left-to-right LMs in the context of sinistrodextral (left-to-right) writing systems. When using a single-stack structure, causal LMs are said to use a decoder structure (cf. Section 2.5.4). Self-supervised training consists in teacher forcing with a sliding window of size m over tokens in the corpus w, i.e. at training step $\tau \in \mathbb{N}^*$, $x_\tau = (\texttt{<BOS>})$ and $y_\tau = \left(w^{(\tau-1)\cdot m+1}, w^{(\tau-1)\cdot m+2}, \ldots, w^{\tau\cdot m}\right)$ (cf. Figure 2.5). Pretraining with causal language modeling has been introduced by GPT [401] and declined by subsequent models, often characterized by increasing capacities and/or increasing pretraining dataset sizes [402, 56, 36, 565, 471, 427, 358, 475, 567, 294, 404, 195, 502, 79]. The emergence of these Large Language Models (LLM) led to impressive results in generating fluent and long sequences of coherent and sometimes astonishingly creative text.

Bidirectional Language Models In de Saussure and Baskin [103], the founder of modern linguistics, Ferdinand de Saussure, considered linearity to be a fundamental property of human language:

> "The Linear Nature of the Signifier:
>
> The signifier, being auditory, is unfolded solely in time from which it gets the following characteristics: (a) it represents a span, and (b) the span is measurable in a single dimension; it is a line.
>
> In contrast to visual signifiers (nautical signals, etc.) which can offer simultaneous groupings in several dimensions, auditory signifiers have at their command only the dimension of time. Their elements are presented in succession; they form a chain. This feature becomes readily apparent when they are represented in writing and the spatial line of graphic marks is substituted for succession in time.
>
> Sometimes the linear nature of the signifier is not obvious. When we accent a syllable, for instance, it seems that we are concentrating more than one significant element on the same point. But this is an illusion; the syllable and its accent constitute only one phonational act. There is no duality within the act but only different oppositions to what precedes and what follows. [. . .]
>
> The linear nature of language [. . .] rules out the possibility of pronouncing two elements simultaneously. The elements are arranged in sequence on the chain of speaking."

The linear characteristics of the sound chain can be found in many modern writing systems, particularly in alphabet-based scripts. This motivates the unidirectionality developed in causal LM.

However, Saussure's structural linguistics has been criticized by the more recent founding father of modern linguistics, Noam Chomsky. With the introduction of multi-layered "deep structures" [78], the chomskian generative linguistics implicates the concept of simultaneity observed for instance in non-verbal communication (e.g. sign languages [196]), non-lingual auditory perception (e.g. philharmonic orchestras) or non-verbal visual perception (e.g. photographs).

As a matter of fact, the unidirectional hypothesis is not necessary either in NLU. Bidirectional language modeling [14] is a clever way to build good representations from self-supervision. For example, learning to predict that `‘‘1961’’` precedes `‘‘was the last year in which the majority of Hollywood films were released in black and white.’’` corresponds to memorizing the correct date of a fact, whereas learning that `‘‘black and white.’’` follows `‘‘1961 was the last year in which the majority of Hollywood films were released in’’` amounts to remember attributive information given an incomplete, dated fact. Because a bidirectional LM (BiLM) aims at learning, with self-supervision, semantic representations of input text for NLU, its structure is often referred to as an Auto-Encoder when it is single-stack (cf. Section 2.5.4). However, to avoid confusion with Goodfellow et al. [154]'s more restrictive definition of neural Auto-Encoders[20] (cf. Section 2.5.5), we prefer the general term of "self-supervised encoder".

Forward-backward Language Models [388] introduced bidirectional embeddings from forward-backward LMs. The LM is bidirectional because it is trained to predict a standard "forward" auto-regressive LM $\left(P\left(s^t|s^1, s^2, \ldots, s^{t-1}; \theta\right)\right)$ as well as a "backward" auto-regressive LM $\left(P\left(s^t|s^{t+1}, s^{t+2}, \ldots, s^m; \theta\right)\right)$, in order to learn word representations.

Here, language modeling is said to produce contextualized or dynamic word embedding [338, 387, 336, 388]. The novelty is that the same token can be embedded depending on the context it appears in. It is particularly useful to disambiguate polysemes and homographs[21] in downstream tasks. We give in Table 2.2 an example of the different definitions of the word "bank", illustrating why contextually-meaningful embedding is sometimes needed.

Masked Language Models Beyond forward-backward LMs, subsequent works have proposed additional BiLMs. Actually, self-supervised pretraining for NLU gained a foothold when BERT [108] achieved state-of-the-art results on the GLUE benchmark. Since then, only derivative models have renewed the leaderboard [311, 439, 264, 174, 493]. The pretraining task introduced by BERT is the Masked Language Model (MLM)[22]. If the corpus $\left(w^1, w^2, \ldots, w^q\right)$ is split into n sentences $s_1 \frown s_2 \frown \ldots \frown s_n$, masked language modeling involves corrupting a sentence $s \in (s_i)_{i\in[1..n]}$ like with denoising autoencoding. The noised sentence $\tilde{s} = \left(\tilde{s}^1, \tilde{s}^2, \ldots, \tilde{s}^m\right)$ is corrupted with a specific process originally proposed by Taylor [499] and detailed in equations (2.5). In words, this corresponds to drawing an event C_t that token s^t is corrupted. BERT-style corruption takes the form of a replacement by a special "Mask" token `<M>` with high probability, or by a uniformly random token from V with low probability (explained by the fact that the `<M>` token is reserved for pretraining only).

$$\begin{aligned}
&P\left(C_t\right) = 0.15 \\
&P\left(\tilde{s}^t = s^t|s^t, \bar{C}_t\right) = 1 \\
&P\left(\tilde{s}^t = \texttt{<M>}|s^t, C_t\right) = 0.8 \\
&P\left(\tilde{s}^t = s^t|s^t, C_t\right) = 0.1 \cdot \left(1 + \frac{1}{|V|}\right) \\
&P\left(\tilde{s}^t = w|s^t, C_t\right) = \frac{0.1}{|V|} \;\; \forall w \in V \backslash \left\{s^t\right\}
\end{aligned} \tag{2.5}$$

From the model's perspective, the input x is the corrupted sentence $\tilde{s}$ (cf. Figure 2.5). Though, contrary to DAEs,

[20] "An Auto-Encoder is a neural network that is trained to attempt to copy its input to its output."

[21] Homophones in speech processing

[22] BERT was also trained with contrastive learning, but Liu et al. [311] showed that its impact was limited.

Remark	**Semantic meaning**	**Part of speech**	**Translation in Modern French**
From Old Italian *banca*	An organization where people and businesses can invest or borrow money, change it to foreign money, etc.	Noun	Banque
	A building where the services described above are offered.	Noun	Banque
	To keep or put money in a particular building described above.	Verb	Mettre en banque / de côté, déposer, être titulaire d'un compte
	To win or earn a particular amount of money.	Verb	Gagner, faire (de l'argent)
	To rely upon.	Verb	Compter, miser
	In gambling, money that belongs to the owner and can be won by the players.	Noun	Banque
	A collection of something, such as blood or human organs for medical use, in a place that stores these things for later use.	Noun	Banque
From Old English *hōbanca*	Sloping raised land, especially along the sides of a river.	Noun	Bord, rive, berge
	A pile or mass of earth, clouds, etc.	Noun	Talus, massif, pente, remblai, amoncellement, couche, banc
	To collect in or form into a mass, or to make something do this.	Verb	Amonceler
	(Of an aircraft) To fly with one wing higher than the other when turning.	Verb	Virer
	(Of an aircraft) The inclined turn described above.	Noun	Virage incliné
From Old French *banc*	A row of similar things	Noun	Banc, rangée, batterie
Collocated with "shot"	In pool, a shot in which the player causes the cue ball or an object ball to rebound off a cushion.	Noun	(Coup) par la bande
	In basketball, a shot in which the basketball glances off the backboard before reaching the basket.	Noun	[*Untranslatable*] (Tir) contre la planche

Table 2.2: Homonyms of the English word "bank". Rows sharing the same etymology correspond to polysemes.

masked LMs recover only the masked tokens $\left(s^t\right)_{t\in\{j:\tilde{s}^j=\texttt{<M>}\}}$ rather than the original uncorrupted sentence s. As a result, masked LMs are sometimes referred to as Masked Auto-Encoders. For NLU tasks such as classification, a special `<CLS>` token may be prepended to the input. In this way, the model is able to learn not only contextualized representations of individual tokens but also aggregate representations of the entire text via the representation h_0 of the `<CLS>` token, processed in subsequent classification layers of the model. Another special token, `<SEP>` is also used to separate sentences when the pretraining involves contrastive learning, comparing pairs of sentences.

The second advantage of BERT is the transformer backbone used to compute the masked LM (cf Section 2.6.3).

Unified Language Models Supervised Seq2Seq LMs and self-supervised causal or bidirectional LMs have been developed to train parameters of models specialized on one single type of NLP tasks: CNLG, UNLG or NLU, respectively. A recent trend seeks to unify LM pretraining to address one or more types of tasks.

Permutation Language Models The first attempt to unify causal and bidirectional LM was introduced by Yang et al. [555]. Compared to causal LMs, masked LMs use a special Mask token `<M>`, which causes a discrepancy between pretraining and fine-tuning. To overcome this gap while keeping bidirectionality in pretraining NLU systems, XLNet, and its successor MPNet [479], proposed a generalized self-supervised task, called Permutation LM and consisting in auto-regressive language modeling on a uniformly random permutation $\tilde{s} = \left(\tilde{s}^1, \tilde{s}^2, \ldots, \tilde{s}^m\right)$ of the sequence $s = \left(s^1, s^2, \ldots, s^m\right)$. Formally, let $\mathfrak{S}_m$ be the set of all permutations of $[1..m]$. Permutation LM maximizes the likelihood $P\left(\tilde{s}^t|\tilde{s}^1, \tilde{s}^2, \ldots, \tilde{s}^{t-1}; \theta\right)$ where $P\left(\tilde{s} = \left(s^{\sigma(1)}, s^{\sigma(2)}, \ldots, s^{\sigma(m)}\right) \mid \left(s^1, s^2, \ldots, s^m\right)\right) = \frac{1}{m!}\ \forall \sigma \in \mathfrak{S}_m$. Permutation LM led to improvement on several NLU tasks compared to the original masked LM.

Sequence-to-Sequence Masked Language Models Song et al. [478] and Raffel et al. [406] have adapted bidirectional pretraining to an encoder-decoder structure with the goal of improving NLU and NLG in addition to the text-to-text abilities of Seq2Seq LMs. When pretraining with Sequence-to-Sequence masked LM (Seq2Seq MLM), the encoder is fed with the corrupted input $\tilde{x}$ and the decoder is trained to predict the pseudo-labels in an auto-regressive way. Additional Seq2Seq MLMs have been proposed, such as the summarization-specialized PEGASUS [569].

Multi-task self-supervised Pretrained Language Models Recent LMs have been proposed to pretrain systems with a unified self-supervised LM based on controlling the information flow (causal, bidirectional or Seq2Seq) in its backbone neural architecture, via dynamic attention masking (cf. Section 2.6.3). Pseudo-mask LM Dong et al. [112], Bao et al. [19] enables pretraining a unique model to gain both NLU and UNLG abilities. In addition to these two types of tasks, General LMs [114] can also address CNLG elegantly.

2.5.9 Transfer Learning via Knowledge Adaptation to Downstream Tasks

Pretrained models reveal their full potential when adapted to specific—and possibly many—problems, called downstream tasks, and evaluated by labeled datasets (cf. Section 2.3). As mentioned in Section 2.5.6, strongly supervised fine-tuning has been the standard knowledge adaptation procedure after a model has been pretrained with self-supervision.

LMs have an essential characteristic: they can represent task descriptions as well as targets in the same semantic space as inputs. GPT-2 [402] and T5 [406] were among the first systems to solve multi-NLU-and-CNLG tasks in a full text-to-text format after the model had been trained. The control of the task to solve was done by conditioning task-specific text prefixes prepended to the input. Keskar et al. [232] used these prefixes, called "control codes", to steer UNLG in text subspaces characterized by specific style or content attributes. Raffel et al. [406] showed the mutual benefit resulting from gradient-based fine-tuning of a single model on a mixture of tasks and illustrated transfer learning between downstream tasks (in addition to transfer learning between the pretraining and fine-tuning phases).

Fine-tuning often requires a large amount ($\sim 10^4$ to $\sim 10^6$) of labeled examples per downstream task. It also hinders the sought-after capability of alternating the resolution of several—possibly unrelated—tasks without further parameter updates. These reasons have motivated the development of techniques leveraging the pretraining phase (denoted meta-learning [56] in this context) to address transfer learning directly. Besides fine-tuning, recent works [402, 56] have shown that task-specific knowledge can be acquired by manipulating causal LMs' behaviors (cf. Section 2.5.9). Plus, these methods paved the way for efficient few-shot learning in NLP.

Few-shot learning ML traditionally requires at least hundreds of data points for training a system to solve a complex task. However, at a certain cognitive developmental stage, the biological brain has sufficient innate and acquired knowledge[23] to learn new tasks with few or even no examples. For instance, if a human is asked to identify whether two different sentences have the same meaning, they will perform pretty well without any example and whether or not they solved a similar problem before. Sometimes, an example is needed to understand the task. If an English speaker is asked to identify pleonasms, an example such as "to exit outside" may be needed if the person does not know the word "pleonasm". More difficult tasks may require more than one example. Asking to recognize zeugmas or semantic syllepsis may require positive examples like "They covered themselves with dust and glory" [509], "On their flippant way through Italy, the French carelessly picked up Genoa, Naples and syphilis" [24], as well as a few negative examples. The setup where a pretrained system has access to a minimum description of a task and/or a small amount of annotated data, typically less than 100, is called few-shot learning. In ML, if a pretrained system has access to a single annotated example, the setup is called one-shot learning [344, 126]. For

[23] On the nature versus nurture debate, see Coll et al. [84], Goldhaber [150] and Keller [230]

[24] Quote attributed to Voltaire

instance, a system that has properly clustered the classes seen during pretraining may be able to classify inputs with regard to the previous classes plus an additional class to whose the single annotated example belongs. In the extreme, zero-shot learning [67, 265, 259, 476, 368] refers to the case where the pretrained model cannot see any labeled example to solve a task it has not been explicitly pretrained on.

Contrary to BERT [108] and T5 [406] which require strongly supervised gradient-based fine-tuning of their parameters with a significant number of labeled examples to solve one or several downstream task(s), GPT-2's breakthrough was to propose a clever way to address several downstream tasks with a single frozen model [402]. The passable-but-promising results showed that a self-supervised causal LM could learn new tasks in a few-shot learning setup. The method, described in the next section, was later called "in context-learning" by Brown et al. [56]. Self-supervised pretraining on a dataset covering a tremendous amount of information, including domain knowledge, could theoretically suffice to address in a few-shot learning setup, NLP tasks where targets can be expressed with text, provided the model has enough capacity. For instance, the unlabeled sentence "I loved it because I thought she was a great director" might participate in indicating that appreciating a director means the opinions on their movies is positive. Radford et al. [402] provide additional examples of naturally occurring demonstrations of translation found in large pretraining corpora. Recent evaluations on NLP benchmarks [56, 79] tend to show that scaling the parameters of LMs (cf Section 2.5.9) does indeed close the gap between state-of-the-art fine-tuned LMs and large-scale causal LMs used in few-shot learning setups. Whether this means that Artificial general intelligence [132] or human-level AI will eventually be achieved with large-scaled models is an open debate. Controlling causal LMs with natural language descriptions of downstream tasks (and potentially a couple of labeled examples) is done with methods based on clever input tuning.

Few-shot and zero-shot transfer with input tuning Early few-shot learning involved gradient-based parameter fine-tuning [130, 33, 223] though input tuning is the current state-of-the-art approach. Prompt augmentation is the method consisting of feeding pretrained causal LMs with information necessary to solve a task given an input text. The model is stimulated with a prompt (or context), like ``The director is great. I thought the movie was ''. Then, probabilities of a pre-defined set of texts, called verbalizers and corresponding to the classes of the problem (e.g., in this situation ``good'' for positive sentiment and ``bad'' for a negative one) are compared. A prompt (cf. Section 2.5.8) x' is formed with the input text x and a "template" [307]. The process of transforming x into x' is called the prompting function. The template consists of a prefix and/or a suffix. Prefixes and suffixes may take the form of explicit text instructions for zero-shot transfer [440, 528], like the prefix ``Give me the sentiment of '' or the suffix ``What is the sentiment of the previous sentence? ''. Additionally, with recent architectures being able to process long sequences of text without long-term dependency problems, it is possible to pass a few labeled examples in the prompt [272]. This priming-based few-shot learning [253] may concatenate pairs using inter-separation (between pairs) and intra-separation (between the input and the target) symbols, for instance, “`I loved this. => Positive \n Great director! => Positive \n I really hated it. => Negative \n This was the worst movie I have ever seen! =>` ”. The power of prompt-based learning for multi-task learning [60, 402] comes from the ease in of switching from one application to a totally different one with a single change in the template. Instructions and examples can be mixed in many ways for the same task. Actually, prompt template engineering has recently become an active research topic.

Prompt Engineering

Manual template engineering A straightforward way of exploring how linguistic and world knowledge acquired by LMs are affected by prompt templates is to manually craft prefix prompts [389, 444, 445, 446].

Automated prompt learning Besides manually designed prompts, templates can be optimized with automated methods [214, 470, 142]. Further works have proposed to transform prompt learning into continuous prompt tuning [399, 588, 165, 292, 310, 280, 161]. Whereas templates (hard prompts) are discrete instructions written in natural language, it is possible to prepend dense representations (soft prompts) to the input right after it has been embedded by the initial neural layer (cf. Section 2.5.2). Besides offering a finer-grained optimization framework, this tuning strategy requires far fewer parameters to train than what is required by traditional full-model tuning, hence producing major savings in time and space.

Large Pretrained Language Models A major development currently happening in neural NLP is the scaling of Pretrained Language Models (PLMs). Indeed, we observe a race to develop and use resources to pretrain increasingly deeper models on ever larger datasets. This is justified by empirical studies showing that not only LMs' capacities (reflected by performances on downstream NLP tasks and text generation) are smoothly improved by scaling up the parameters and training datasets [228], but also it seems no upper bound has (yet?) been reached [79]. At the time of writing this dissertation, models have regularly grown from the 100-million-parameter GPT-1 [401] to the 300-million-parameter BERT [108], the 1.5-billion-parameter GPT-2 [402], the 8.3-billion-parameter Megatron [471, 358], the 11-billion-parameter-T5 [406], the 17.2-billion-parameter Turing NLG [427], the 70-billion-parameter Chinchilla [195], the 137-billion-parameter LaMDA [502], the 175-billion-parameter GPT-3 [56], the 178-billion-parameter Jurassic-1 [294], the 200-billion-parameter PanGu-α [567], the 280-billion-parameter Gopher [404], the 530-billion-parameter Megatron-Turing NLG [475], the 540-billion-parameter PaLM [79], and finally the 1.6-trillion-parameter Switch Transformer [125]. The crux of the problem lies in parallel computing hardware and software. It was enabled by advances in model architectures (cf. Section 2.6) and AI accelerators like Tensor Processing Units (TPUs) and Graphics Processing Units (GPUs).

Large self-supervised Pretrained LMs are currently considered as driving a paradigm shift, similar to the sea changes observed when ML algorithms—and later DL architectures—emerged and became dominant in AI. These

Model	**GLUE Score** $\uparrow$	**SuperGLUE Score** $\uparrow$
CBoW [343] (Baseline)	58.6	44.5
GPT [401, 56]	72.8	71.8[a]
BERT-Large [108]	80.5	69.0
RoBERTa-Large [311]	84.6	88.5
T5-XXL [406]	90.3	89.3
DeBERTa-TuringNLRv4 [174]	90.8	90.3
ERNIE [582, 493]	$\mathbf{91.1}$	$\mathbf{90.6}$
Human	87.1	89.8

[a] Few-shot learning.

Table 2.3: GLUE and SuperGLUE scores of high-impact LMs on the test sets according to the leaderboards. Except for the score of GPT(-3) on the SuperGLUE benchmark, all models are fine-tuned on the downstream tasks.

seismic changes are characterized by quick standardization of approaches that are eventually applied to real-life problems and directly—or indirectly—impact citizens' daily lives. The current AI transition stemming from Pretrained LMs has been named the revolution of "Foundation Models" [46]. Not only have these models become the "substrate of NLP", showing astonishing capabilities in world knowledge, creativity as well as common sense, logical and arithmetical few-shot reasoning (examples of complex joke explanation are provided by Brown et al. [56]), but similar ideas have been applied to further modalities [498] such as images [73, 412, 435], tables [558] or proteins [423]. Pretrained LMs made of numbers of parameters exceeding the number of neurons in human brains [179] are now considered as "foundations for a wide range of downstream applications". It is not well understood yet whether their aptitudes are based on actual generalization, as in human intelligence, or on powerful memorization [402]. Opportunities and risks have been extensively explored by Bommasani et al. [46], including potential harms concerning bias, ethics, fairness, privacy, safety, security, or natural environment.

Table 2.3 shows the average GLUE and SuperGLUE test scores (cf. Section 2.3) of high-impact LMs as well as a baseline (cf. Section 2.5.3) and human performances. Apple-to-apple comparison of released or published models on benchmarks is difficult [12]. Indeed, models often differ on several aspects such as objective functions, sizes, pretraining datasets, or knowledge adaptation strategy.

NLP methods have been developed in light of ML, with old systems relying on handcrafted features. Then, given the powerful abilities of a particular class of end-to-end ML models—called deep Artificial Neural Networks (ANN) [362]—in other supervised AI problems such as computer vision, research has introduced Deep Learning architectures tailored to language characteristics [149]. It appears that ANNs are now widely used in all sorts of NLP tasks, given they achieve state-of-the-art results on many benchmarks.

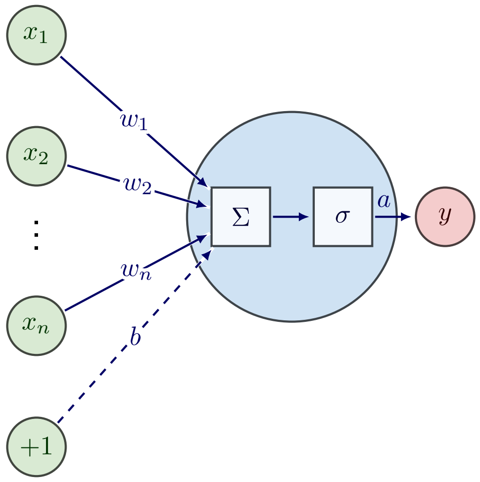


Figure 2.6: Computations made by a single artificial neural unit. Figure best viewed in color.

2.6 Neural network architectures

ANNs [426] are loosely inspired—though not being replicas[25]—of biological neural networks described by neuroscientists [337]. From the theoretical computer science perspective, an ANN can be seen as a circuit [531], i.e., a model made of a computation graph processing information as it flows through it. This concept is specific when compared to the more traditional procedural (e.g., programming in C++) or functional (e.g., programming in OCaml) approaches of computer science. The simplest kind of neural network is called the Feedforward Neural Network, and we describe its structure below.

2.6.1 Feedforward neural networks

Compared to other ML systems, ANNs can be characterized by their cognitive-inspired structure consisting of layer stacks, each made of many artificial neural units making small computations in parallel.

Artificial neural units The building block of ANNs is simple parametric and generic computing units called artificial neurons and inspired by the mathematical modeling of the molecular biology observed in nerve cells [226]. A neural unit is fed with some vector of n input features $\mathbf{x}$, and outputs a scalar y (also denoted a when part of a network). The transformation computed by the unit is a non-linear function of the weighted sum of the input features plus a bias term (cf. Figure 2.6). The computation can be formally written $y = \sigma\left(\mathbf{w} \cdot \mathbf{x} + b\right)$.

The weight vector $\mathbf{w}$ and the bias scalar b are the neuron's parameters. The non-linear function σ is called an activation function. As training ANNs relies on gradient-descent-based optimization algorithms, activation functions are often chosen to be continuously differentiable (e.g. hyperbolic tangent, sigmoid [168], softmax [45], Gaussian Error Linear Unit [178], Swish [411]) though the REctified Linear Unit [353] is a popular activation function not

[25] The design of modern ANNs should not be confused with the modeling of biological brains.

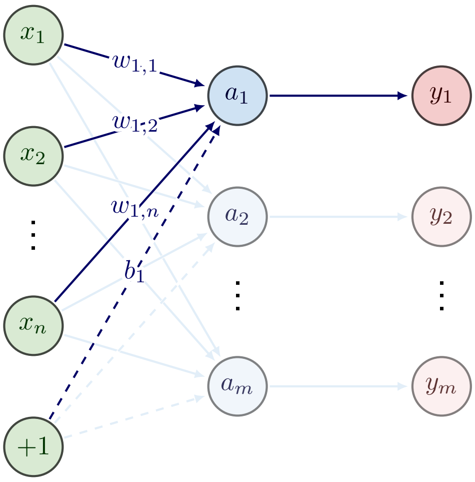


Figure 2.7: Computations made by a single feedforward neural layer. Figure best viewed in color.

differentiable at 0. An intuitive way to apprehend the activation's operation is to see it as an indicator triggered by specific linear combinations of the input features $\mathbf{x}$. For instance, a specific neural unit in a system might be "activated" when it detects the presence of the adverb ``not'' in a sentence. It will then pass this output signal to the input of subsequent units that may take this information into account if they take part in the process of determining whether the sentence contains some semantic negation and how to process it.

Feedforward neural layers A feedforward neural layer is a set of neural units, each with its own set of parameters, making similar computations in parallel on the same input vector $\mathbf{x}$. The layer produces an output vector $\mathbf{y}$ and is parameterized by a $m \times n$ weight matrix $\mathbf{W}$ and a bias vector $\mathbf{b}$. The computation can be written $\mathbf{y} = \sigma\left(\mathbf{W} \cdot \mathbf{x} + \mathbf{b}\right)$ assuming the activation function σ applies to a vector element-wise (cf. Figure 2.7). As each neural unit is connected to all the input features in $\mathbf{x}$, this layer is sometimes qualified as "fully-connected" as opposed to layers whose neural units are only partially connected to the input features (cf. Section 2.6.2). As an example, natural language embeddings (cf. Section 2.5.2) are modeled with fully-connected feedforward layers.

Fully-connected neural networks We can go one step further and compute the output **y** as a non-identity function of the activation values. This can be achieved with a second neural layer with its own set of parameters called the output layer (cf. Figure 2.8). Now, $\mathbf{y} = \sigma^{(2)}\left(\mathbf{W}^{(2)} \cdot \sigma^{(1)}\left(\mathbf{W}^{(1)} \cdot \mathbf{x} + \mathbf{b}^{(1)}\right)\right)$. The first layer is called the hidden layer because it has direct access neither to the output nor to the input. The input features $\mathbf{x}$ is sometimes referred to as the input layer, even if it has no parameter. Note that the output layer does not require bias as in the hidden layer.

Actually, we just described the architecture of a standard multi-layer Feedforward Neural Network (FFNN), also known as Fully-Connected Neural Network (FCNN). FCNN made of one hidden layers are used to pretrain non-contextual word embeddings like Word2vec (cf. Section 2.5.3). In NLP, the input layer is the one-hot encoding of words in the vocabulary V. The idea of sequential neural layers comes from the neocortical layering observed

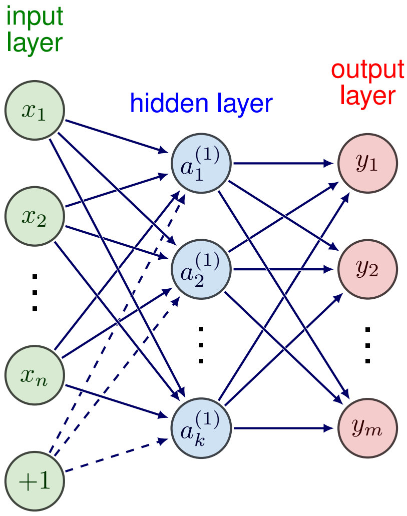


Figure 2.8: Computations made by a fully-connected feedforward two-layer neural network. Figure best viewed in color.

by neuroscientists [256]: for instance, layer IV acts as the recipient of incoming sensory signal later processed by subsequent layers.

The network's trainable parameters θ is the set of all parameters found in each layer. This architecture defines a mapping f_θ between the input and the output space. The non-linear activation functions enable this multi-layer structure. Plus, non-linearities are desired when it comes to making predictions on non-linearly separable data.

Universal approximation theorem The main theoretical result in the mathematical theory of ANNs is that they are universal function approximators.

Theorem 1 *Let $S \subseteq \mathbb{R}^n$ be compact. Then, $\forall \left(g, \epsilon\right) \in \mathcal{C}\left(S, \mathbb{R}\right) \times \mathbb{R}_+^*$, there exists a FCNN f_θ with one hidden layer and a finite number of neural units such that $\sup\limits_{\mathbf{x} \in S} \| g\left(\mathbf{x}\right) - f_\theta\left(\mathbf{x}\right) \|$.*

Even if the result does not tell how many weights are needed or how to find them, it grounds ANNs as compositional function approximators.

Deep neural networks A characteristic of ANNs is that they can stack many hidden layers, each learning specific functionalities. The use of deep neural networks is often called Deep Learning (cf. Figure 2.9). Besides depth, the other hyperparameter characteristic of ANNs' capacities is their width, that is, the number of units within a layer. It has been shown that deep and wide ANNs can approximate functions of increasing complexity. Of course, the design of ANNs is tied with the optimization strategy used to train them as well as their connectionist structures and the mathematical operations performed by units or cells.

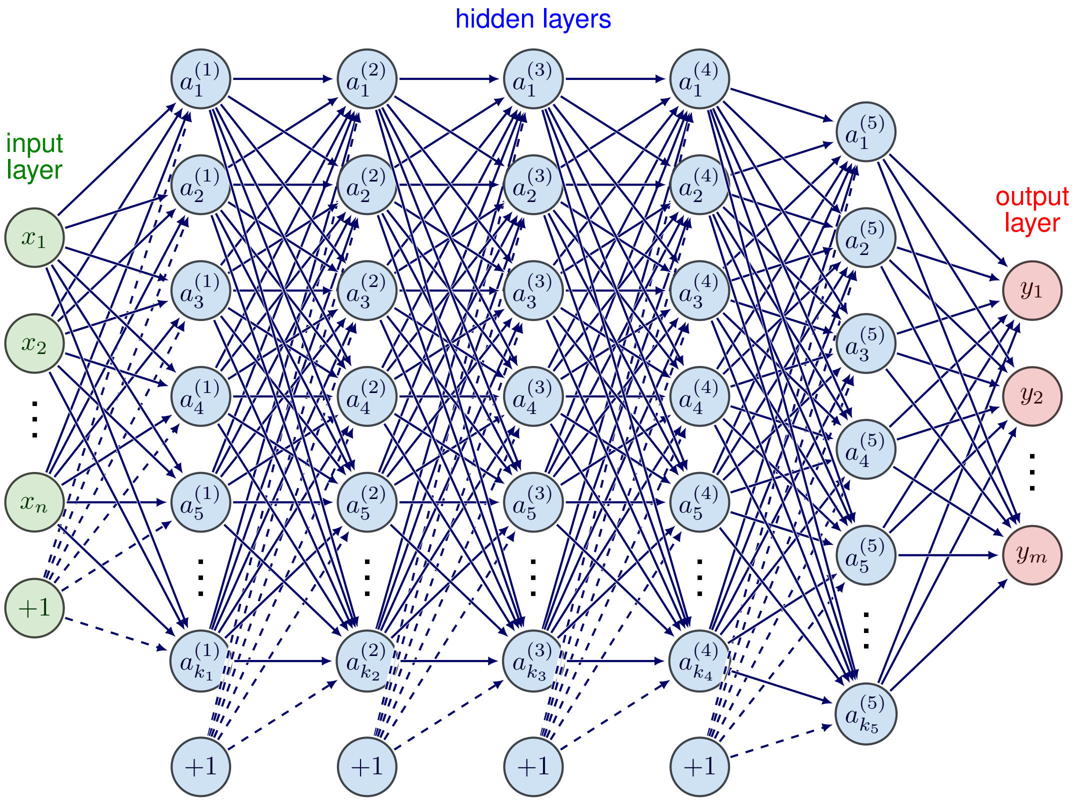


Figure 2.9: Standard structure of a deep Feedforward Neural Network. Figure best viewed in color.

Learning via ANN training After their parameters θ are initialized, for instance randomly, ANNs usually learn to approximate functions or probability distributions with gradient-descent-based optimization [61]. In supervised learning, differentiation algorithms tune the parameters θ with the goal of minimizing the prediction error computed with a loss function or, alternatively, maximizing the probability distribution of the observed data. Efficient training of the network's parameters is enabled by the back-propagation algorithm [430, 431], leveraging the chain rule [278, 285] and dynamic programming [24] to make fast updates of each unit's parameters. Additional advances in hardware and training techniques led computer scientists to develop larger ANNs [187, 27].

ANNs have been studied as efficient representation learners for a long time [189]. Not only have deep ANNs demonstrated better learning performances than previous ML models [250, 472, 497], but they do so by automatically learning salient features with end-to-end supervised training [87, 28]. This means that there is no other signal coming to the network than the input features x processed by the input layer and the target y used to backpropagate errors from the output layer. In other words, even if no agent tells the model *how* to solve the task, the network learns suitable representations in its hidden layers. Indeed, studies on both neural Computer Vision [473, 563, 480] and neural NLP[26] [500, 82, 211, 425] showed that the hidden layers learn hierarchical structures of the feature distributions, similar to traditional pipelines based on manual feature engineering. For instance, in NLU, BERT's lower-level layers tend to be dedicated to syntactic processing while the higher-level layers focus on semantic understanding.

[26] Sometimes called "BERTology" when investigating modern systems

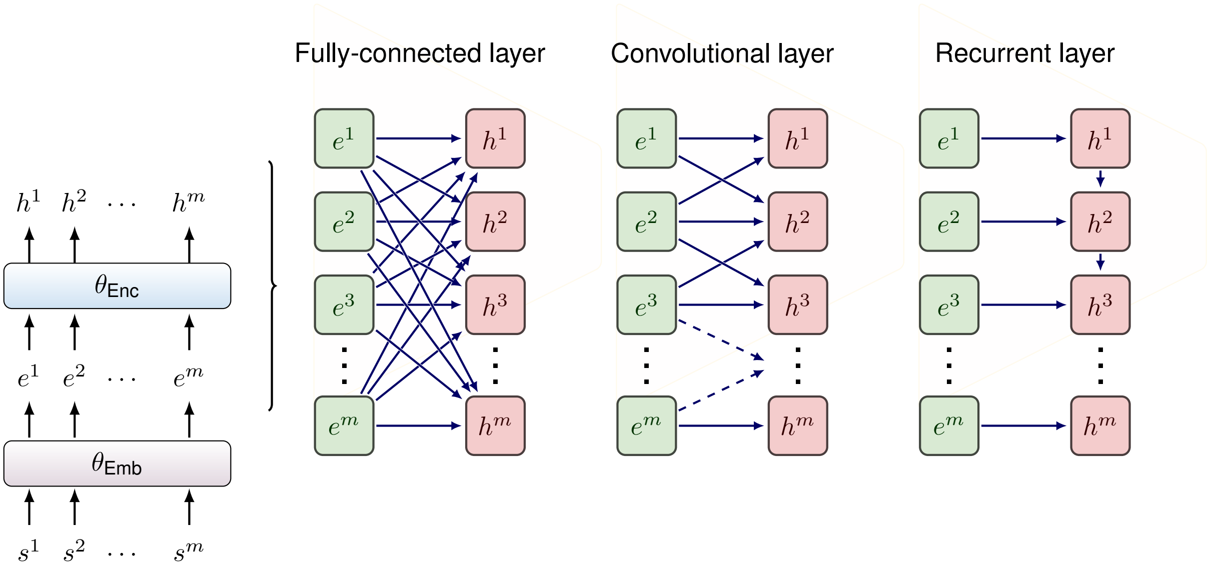


Figure 2.10: Popular alternative architectures of layers found in neural NLP models. s^i represents the one-hot encoding of the i^{th} token. e^i is the non-contextual embedding of the i^{th} token. h^i is the contextualized neural representation of the i^{th} token. Figure best viewed in color.

This echoes the observations of Broca's area in the frontal cortex, broken into two component regions. Brodmann's areas 44 and 45 are assumed to be respectively involved in syntactic and semantic tasks [156].

Feedforward Neural Language Models LMs were modeled by ANNs first by Bengio et al. [26]. Their Feedforward Neural Language Model (FNLM) was made of an embedding layer, a hidden layer, whose inputs are the concatenation of the word embeddings, and an output layer whose activation is softmax.

2.6.2 Alternative neural architectures

Advances in HPC (HPC) devices and optimization algorithms have driven the development of large ANNs. Their expressiveness has also been improved by architecture engineering, which contributes to the scaling of the models' parameters as well. We present here a set of popular architectures for NLP, illustrated in Figure 2.6. See Goodfellow et al. [154] for further reference on Deep Learning architectures.

Convolutional layers Besides FCNN, another major family of FFNN has been introduced in computer vision. Inspired by neurophysiology research on the receptive fields in the visual cortex [205], Convolutional Neural Networks (CNNs) apply convolution filters to local features from the previous layer [137, 520, 274, 275]. These convolving operations have later been transposed to neural NLP [236, 224, 208]. In NLP-oriented CNNs, the first hidden layer makes convolving operations with a window moving on the stacked word embeddings (cf. Figure 2.10). The Universal Sentence Encoder [63] used to embed sentences in Chapter 5 is a CNN.

FFNNs are good encoders of fixed-length data. In particular, CNNs were originally developed to process images. Yet, non-feedforward ANNs have been proposed to encode and generate variable-length sequential data, such as text.

Recurrent layers Inspired by cognitive modeling of temporal structures [432, 433], recurrent layers are neural layers whose units process inputs seen as time series [118]. At each time step t, the neural unit computes a hidden state h^t by applying operations to the t^{th} input x^t and the $t-1^{\text{th}}$ hidden state h^{t-1}. Figure 2.10 (right) shows an unfolded representation of a recurrent layer. Hidden states are expected to act as—theoretically infinite—memory, computed from the past inputs $x^{1:t}$ (or context). Actually, some important characteristic of recurrent layers is that the units share their parameters. This is equivalent to introduce cycles in ANN computation graphs, while FFNN's graphs are acyclic. Parameter sharing is motivated by better generalization on variable-length sequences and reduces space complexity since the model width is independent of the input sequence lengths. In practice, the training of Recurrent Neural Networks (RNNs) is done by unrolling the cycles, which represents high time complexity in the case of deep RNNs processing long sequences, compared to parallelized architectures. For this reason, RNNs tend to be shallower than other neural architectures, despite evidences showing that RNNs also benefit from deeper architectures [159, 375].

RNNs have been widely employed in NLP. Mikolov et al. [341] introduced recurrent LMs. RNN-based encoder-decoders are used in Seq2Seq learning [495, 75]. BiLMs (cf. Section 2.5.8), and more generally NLU systems, have often been implemented with bidirectional RNNs [448] while auto-regressive NLG has been modeled with unidirectional RNNs.

Variants of the vanilla RNN, called gated RNNs, introduced units—sometimes referred to as cells—performing more complex operations whose goal is to better model the context to address the long-term dependency problem. Long Short-Term Memories (LSTMs) [192, 145] and Gated Recurrent Units (GRUs) [80] are popular cells. In Chapter Chapter 4, we used Bidirectional LSTMs (BiLSTMs) to detect toxic spans in online posts.

Further details on architectures, training, and comparisons of RNNs, see Graves [158]. The main drawback of deep RNNs is the unstable training known as the vanishing and exploding gradient problem [25].

2.6.3 Attention neural networks

Information storage is a concept essential to understanding the mind as well as for the development of AI. The nature of traditional computer memories differs significantly from the human memory studied by cognitive neuropsychology. Yet, cognitive and neurocomputing architectures have spurred effective mechanisms improving ANNs' abilities [488, 530, 160]. In particular, attention is the core building block of a prominent neural architecture that has gradually become the de-facto standard in state-of-the-art NLP: the transformer [514].

Attention mechanism Because the human working memory (or short-term memory) has a limited capacity [348] (7 ± 2 elements), it uses a behavioral process called attention to prioritizing the information. Attention consists of identifying salient stimuli, i.e., signals relevant to the task, and ignoring the remaining incoming sensory data, considered distraction [562]. In speech, sentence stress can draw attention to a specific word in order to adapt its meaning. For instance, the message in the sentence “I never said she stole my money.” varies according to the word(s) stressed [397]. “I never said **she** stole my money.” means that somebody else stole my money, while “I never said she stole my **money**.” means that she stole something else, etc. Visual memory is also known to rely on selective attention[27]. When reading, a human is assumed to focus attention on the words most useful to understand the text. A symbolic way of visualizing this would be to transcribe text like in the following example:

> “1961 was the last year in which the majority of Hollywood films were released in black and white.”

The integration of attention into ANNs was first proposed by Graves [159] and Bahdanau et al. [16]. Their work improved recurrent layers with a mechanism enabling the model to focus on specific tokens of the input sentences. In their famous publication entitled “‘Attention Is All You Need”, Vaswani et al. [514] proposed an attention neural network, called the transformer, without recurrent connections. Its key component is the attention head.

Attention heads Attention heads look like the fully-connected feedforward layers described in Section 2.6.1. Attention head inputs are the d-dimensional representations of a sequence of n tokens (e.g. the initial embeddings of the one-hot encoded tokens, cf. Section 2.5.2). The first step consists in projecting with weight matrices $\mathbf{W}^Q \in \mathbb{R}^{d_k \times d}$, $\mathbf{W}^K \in \mathbb{R}^{d_k \times d}$ and $\mathbf{W}^V \in \mathbb{R}^{d_v \times d}$ the inputs into vectors respectively called queries $\mathbf{Q} \in \mathbb{R}^{n \times d_k}$, keys $\mathbf{K} \in \mathbb{R}^{n \times d_k}$ and values $\mathbf{V} \in \mathbb{R}^{n \times d_v}$. Then, the attention head performs dot-product-based pairwise comparisons on the queries and keys. The attention head outputs the sum of the values, weighted by the comparison scores. In other words, the queries are said to “attend” the values. The formula is given in Equation (2.6).

$$\text{Attention}\left(\mathbf{Q}, \mathbf{K}, \mathbf{V}\right) = \text{softmax}\left(\frac{\mathbf{Q} \cdot \mathbf{K}^\intercal}{\sqrt{d_k}}\right) \cdot \mathbf{V} \tag{2.6}$$

$\mathbf{W}^V$, $\mathbf{W}^Q$ and $\mathbf{W}^V$ are the attention head’s trainable parameters θ_{Att}. The principle of the attention head is that during training, the ANN learns which token pairs should be attended to perform best on the task; the intuition being that the weighted sum summarizes the contextual information contained in the input tokens, independently of their distance to the query. This progress in addressing the long-term dependency problem of RNNs has contributed to the success of transformers. The second advantage proposed by attention heads is that the computations are highly parallelizable and can leverage the capabilities of AI accelerators. Yet, the bottleneck to the scaling of transformers

[27] The Invisible Gorilla Test [65] is a well-known study on intentional blindness

is the quadratic complexity of the attention heads, while RNNs' sequential operations are linear in the input length. Though, the current trend is to deploy large transformers on increasing HPC resources.

Actually, there are two kinds of attention heads found in transformers. First, self-attention heads compute comparisons between each pair of tokens in the input text. Besides, in strongly supervised text-to-text, teacher forcing feeds the network with pairs of input and target texts. In this case, cross-attention heads compare pairs made of one token from the input and one token from the target. By telling which words were the most attended, attention heads contribute to increasing the explainability of ANNs, often described as black boxes. We leveraged this property in Chapter 4.

In order to better model the many ways words can interact with each other, transformers use multi-head attention layers whose output representations are concatenated and linearly transformed with a matrix parameterized by $\theta_{\text{Concatenation}}$. This enables the ANN to capture all the possible semantic and syntactic relationships.

Transformer blocks Transformers are made of stacks of identical blocks including multi-attention heads as well as attention-less fully-connected neural layers, residual connections [173], normalizing layers [13] (cf. Figure 2.11). These additional attention-less layers and operations contribute to improving the transformer training.

Transformers The original transformer architecture is made of a stack of transformer blocks on top of a position-wise feedforward embedding layer, whose role is to encode the order of the words (cf. Figure 2.11). As a matter of fact, the transformer initially referred to the double-stack encoder-decoder structure introduced by Vaswani et al. [514]. However, single-stack transformers underpin many NLP systems. To avoid confusion, we refer to single-stack transformers as unitransformers while encoder-decoder transformers—using both self-attention and cross-attention in the decoder—are called here bitransformers.

The expressiveness of transformers is the reason why they have become the backbone of many neural NLP models. Even though several variants of the original transformer have been explored, Narang et al. [356] did not find modifications which significantly improved the transformer performance on a wide range of applications.

The premise of foundation models is based on the self-supervised pretraining of large transformer-based LMs, equivalently referred to as pretrained transformers, Pretrained LMs, or large LMs in the literature. Further details on transformers can be found in Vaswani et al. [514] and Lin et al. [299].

Attention masks Section 2.5 presented strategies for training ML systems. Even though training objectives are mainly implemented with loss functions, attention neural network stacks trained with teacher forcing (cf. Section 2.5.7) utilize masking patterns to control the context that tokens are allowed to attend (cf. Figure 2.11). In other words, these attention masks condition the representation of query tokens to a specific set of key tokens. For instance, causal LM is implemented with causal attention masks that restrict the information flow used to represent

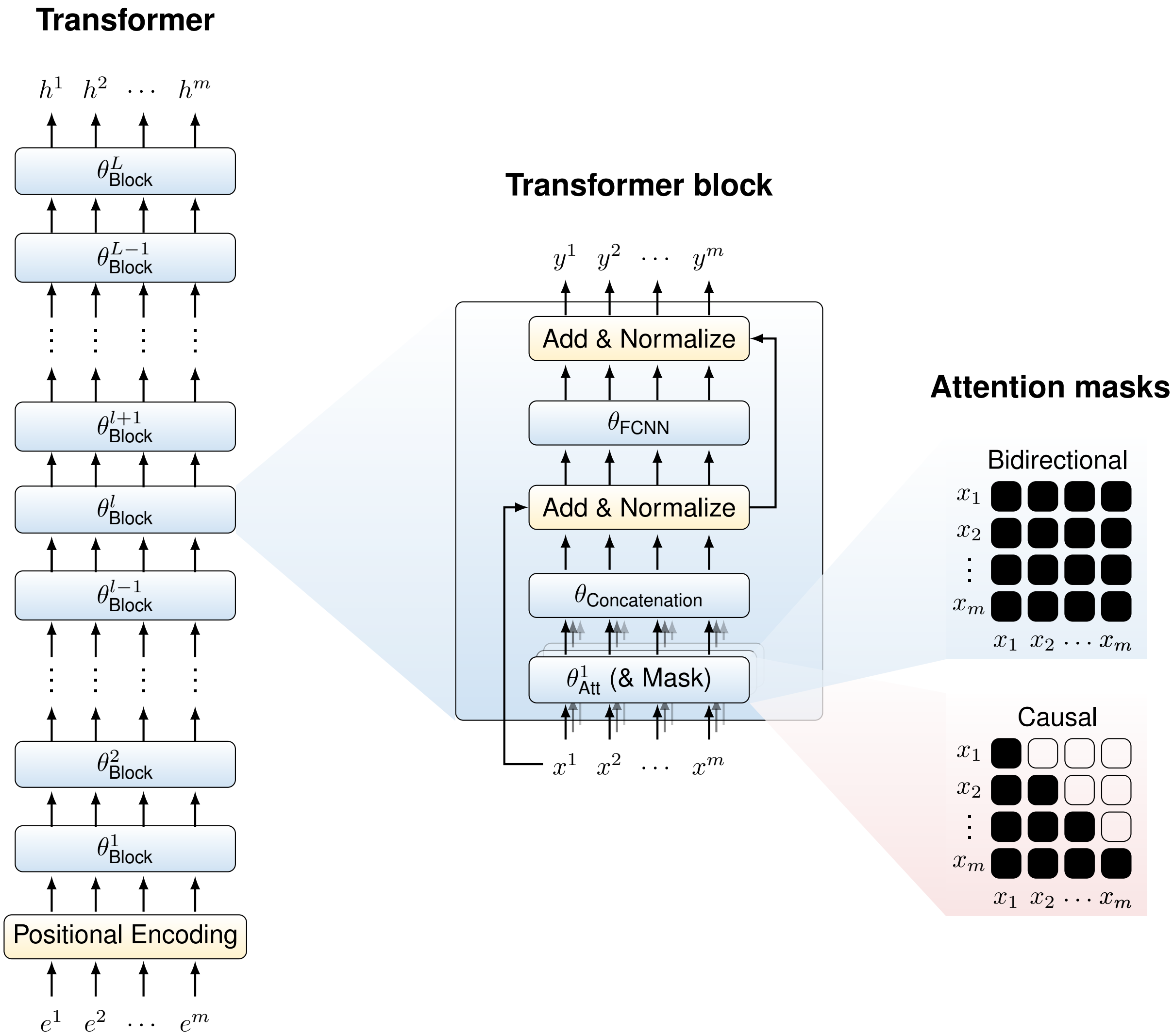

Transformer
h^1 h^2 $\cdots$ h^m
θ^L_{Block}
$\theta^{L-1}_{\text{Block}}$
$\theta^{l+1}_{\text{Block}}$
$\theta^{l}_{\text{Block}}$
$\theta^{l-1}_{\text{Block}}$
$\theta^{2}_{\text{Block}}$
$\theta^{1}_{\text{Block}}$
Positional Encoding
e^1 e^2 $\cdots$ e^m
Transformer block
y^1 y^2 $\cdots$ y^m
Add & Normalize
θ_{FCNN}
Add & Normalize
$\theta_{\text{Concatenation}}$
θ^1_{Att} (& Mask)
x^1 x^2 $\cdots$ x^m
Attention masks
Bidirectional
x_1
x_2
x_m
x_1 x_2 $\cdots$ x_m
Causal
x_1
x_2
x_m
x_1 x_2 $\cdots$ x_m

Figure 2.11: Illustration of the transformer architecture. θ indicates that the layer (in blue) is parametric. **Left**: Unitransformer stack. **Middle**: A detailed transformer block. **Right**: Matrix representation of possible alternative attention masking patterns. The input token at row i is allowed to attend the input token at column j if and only if the cell at (i, j) is shaded. Figure best viewed in color.

the target token to the past tokens.

Chapter 2 conclusion

NLP has studied tasks of gradually increasing difficulty, successively tacked by symbolic AI, statistical ML and deep ANNs. At the crossroads of computer science and linguistics, the field has become enriched by neuropsychology and cognitive concepts. As a matter of fact, NLP is not just about language; it is deeply related to automatic reasoning and world knowledge. We presented in this chapter the winning combination of training strategies, self-supervised objectives, and HPC-powered DL architectures constituting modern neural NLP basis. In light of the recent advances in NLP described in this chapter, subsequent chapters introduce novel methods to apply these “foundation models” to improve online user interactions.

Chapter 3

Civil Rephrases Of Toxic Texts With Self-Supervised Transformers

This chapter presents research conducted with John Pavlopoulos, Jeffrey Sorensen, and Lucas Dixon. It led to an article published in the Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2021) [267].

Chapter 3 abstract

Platforms that support online commentary, from social networks to news sites, are increasingly leveraging Machine Learning to assist their moderation efforts. But this process does not typically provide feedback to the author that would help them contribute according to the community guidelines. This is prohibitively time-consuming for human moderators to do, and computational approaches are still nascent. This work focuses on models that can help suggest rephrasings of toxic comments in a more civil manner. Inspired by recent progress in unpaired sequence-to-sequence tasks, a self-supervised learning model is introduced, called CAE-T5[a]. CAE-T5 employs a pretrained text-to-text transformer, which is fine-tuned with a denoising and cyclic Auto-Encoder loss. Experimenting with the largest toxicity detection dataset to date (Civil Comments) our model generates sentences that are more fluent and better at preserving the initial content compared to earlier text style transfer systems which we compare with using several scoring systems and human evaluation.

[a] The code is publicly available at `github.com/LeoLaugier/conditional-auto-encoder-text-to-text-transfer-transformer` with an Apache License 2.0.

INPUT OFFENSIVE COMMENT	you now have to defend this clown along with his russian corruption.
GENERATED CIVIL COMMENT	**you now have to defend this guy from his russian ties........**
INPUT OFFENSIVE COMMENT	blaming trudeau and the government is just stupid.
GENERATED CIVIL COMMENT	**blaming trudeau and the liberal government is just wrong.**
INPUT OFFENSIVE COMMENT	dubya[a] was a moron.
GENERATED CIVIL COMMENT	**dubya was a republican.**

[a] A nickname for George W. Bush.

Table 3.1: Examples of offensive sentences from the Civil Comments test set and the more civil rephrasing generated by our model. The third example shows that its strategy may involve shifting the original intent since "republican" is not a non-offensive synonym of "moron".

3.1 Introduction

There are many ways to express our opinions. When we exchange views online, we do not always immediately measure the emotional impact of our message. Even when the opinions expressed are legitimate, well-intentioned, and constructive, poor phrasing may make the conversation go awry [568]. Recently, NLP research has tackled the problem of abusive language detection by developing accurate classification models that flag toxic (or abusive, offensive, hateful) comments [101, 377, 543, 139, 134, 568, 513, 560].

The prospect of healthier conversations, nudged by ML systems, motivates the development of NLU and NLG models that could later be integrated into a system suggesting alternatives to vituperative comments before they are posted. A first approach would be to train a text-to-text model [15, 514] on a corpus of parallel comments where each offensive comment has a courteous and fluent rephrasing written by a human annotator. However, such a solution requires a large paired labeled dataset, in practice difficult and expensive to collect (see Section 3.7). Consequently, we limit our setting to the self-supervised case where the comments are only annotated in attributes related to toxicity, such as the Civil Comments dataset [49]. We summarize our investigations with the following research question:

RQ: *Can we fine-tune end-to-end a pretrained text-to-text transformer to suggest civil rephrasings of rude comments using a dataset solely annotated in toxicity?*

Answering this question might provide researchers with an engineering proof-of-concept that would enable further exploration of the many complex questions that arise from such a tool being used in conversations. The main contributions of this work are the following:

- We addressed for the second time the task of unsupervised civil rephrases of toxic texts, relying for the first time on the Civil Comments dataset and achieving results that reflect the effectiveness of our model over baselines.

- We developed a non-task-specific approach (i.e., with no human hand-crafting in its design) that can be generalized and later applied to related and/or unexplored attribute transfer tasks.

While several of the ideas we combine in our model have been studied independently, to the best of our knowledge, no existing self-supervised models combine sequence-to-sequence bitransformers, transfer learning from large pretrained models, and self-supervised fine-tuning (Denoising Auto-Encoder and Cycle Consistency). We discuss the related work introducing these tools and techniques in the following section.

3.2 Related work

Unsupervised complex text attribute transfer (like civil rephrasing of toxic comments) remains in its early stages, and our particular applied task has only a single antecedent [364]. There is a great variety of useful works to tackle the task and this section attempts to summarize the vast majority of these works. We describe below the recent strategies in style transfer when applied to image-to-image tasks. Then, we present the most related lines of work in unsupervised—or more precisely self-supervised—text-to-text tasks.

3.2.1 Style transfer in Computer Vision

The first successful application of Deep Learning to style transfer dates back to the work of Gatys et al. [143] who relied on the supervised pretraining of a CNN to synthesize paintings in a specific style. Then, Zhu et al. [594] proposed a more general approach to image-to-image translation using Generative Adversarial Network (GAN) [153]. CycleGAN is an unsupervised learning method. This means it does not need any correspondence between paired images with the exact same content (e.g., a pair of images with three Equidae in a specific position with a specific background) and different attributes (on the first image, the Equidae are horses, while on the second they are zebras). For training, CycleGAN only needs two "collections" of images, each collection having a known specific attribute or style (e.g., a collection of horses and a collection of zebras). The goal is to make CycleGAN learns the mapping between the first collection (made of examples in the "source" space, with the source attribute) and the second collection (made of examples in the "destination" space, with the destination attribute).

A GAN consists in a two ANNs: a generator (G) and a discriminator (D). The generator aims to transfer images from the source space (X) to the destination space (Y). The discriminator computes the probability that its input image belongs to the destination space. The adversarial training consists in optimizing the discriminator's weights in order for it to correctly classify images in the destination space while maximizing the probability that the generator outputs images considered as destination images by the discriminator. The GAN loss function is describe in Equation (3.1). Its formal optimization will be given in Equation (3.4).

$$\mathcal{L}_{\text{GAN}}\left(G,D,X,Y\right)=\mathbb{E}_{y\sim Y}\ \left[\log P\left(y\in Y|\theta_D\right)\right]+\mathbb{E}_{x\sim X}\ \left[\log\left(1-P\left(G\left(x\right)\in Y|\theta_D\right)\right)\right] \tag{3.1}$$

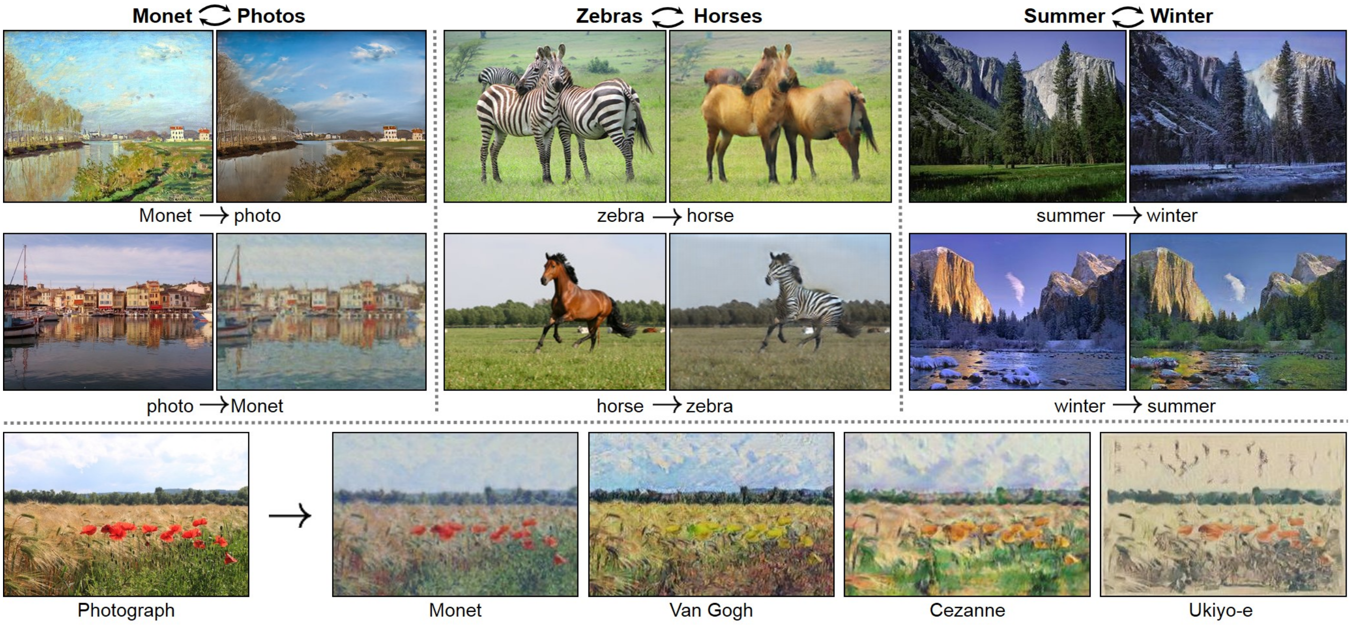


Figure 3.1: Examples of image-to-image transfer with CycleGAN. Figure from Zhu et al. [594].

In order to learn the mapping in both ways, CycleGAN actually involves two generators (one for attribute$_A$ to attribute$_B$, G_{A2B}, and one for attribute$_B$ to attribute$_A$, G_{B2A}) and two discriminators (one for attribute$_A$, D_A, and one for attribute$_B$, D_B). Is an adversarial loss enough, though? Not if the generator's capacity is large enough to map the entire source collection to a random permutation of the destination collection. This is where the Cycle Consistency loss will play its regularization role by constraining the "round trip" image translation to be close to the source image. It is essential to ensure that the content is not altered. CycleGAN's Cycle Consistency loss is given in Equation (3.2).

$$\mathcal{L}_{\text{CYC}}\left(G_{A2B}, G_{B2A}\right) = \mathbb{E}_{x\sim X}\left[\left\|G_{B2A}\left(G_{A2B}\left(x\right)\right) - x\right\|_1\right] + \mathbb{E}_{y\sim Y}\left[\left\|G_{A2B}\left(G_{B2A}\left(y\right)\right) - y\right\|_1\right] \tag{3.2}$$

The CycleGAN model can then be trained by optimizing the loss $\mathcal{L}_{\text{CycleGAN}}$ (cf. Equation (3.3)) being the sum of the adversarial loss and the Cycle Consistency loss (cf. Equation (3.4)).

$$\mathcal{L}_{\text{CycleGAN}}\left(G_{A2B}, G_{B2A}, D_A, D_B\right) = \mathcal{L}_{\text{GAN}}\left(G_{A2B}, D_B, A, B\right) + \mathcal{L}_{\text{GAN}}\left(G_{B2A}, D_A, B, A\right) + \mathcal{L}_{\text{CYC}}\left(G_{A2B}, G_{B2A}\right) \tag{3.3}$$

$$\hat{G}_{A2B}, \hat{G}_{B2A} = \underset{G_{A2B}, G_{B2A}}{\arg\min}\ \max_{D_A, D_B} \mathcal{L}_{\text{CycleGAN}}\left(G_{A2B}, G_{B2A}, D_A, D_B\right) \tag{3.4}$$

A natural question arises when we investigate unsupervised text-to-text transfer: could CycleGAN be directly applied to text? Unfortunately, the answer is no, and the reason is mainly due to the discrete nature of text. Indeed as we saw in Section 2.5.7, search strategies involved in decoding the representations—after the softmax layer—often break the chains of differentiability. Therefore, gradient-based training cannot use the standard backpropagation algorithm. Yet, popular workarounds have been proposed. Besides uses of reinforcement learning to solve the non-differentiability problem when sampling in NLG, efforts to generate text in a continuous space only have been proposed [23, 254].

However, these methods being unstable, hard to train in practice and/or nascent at the time of our work, we decided not to apply CycleGAN, but rather seek ideas in the then recently promising unsupervised neural machine translation.

3.2.2 End-to-end training of text-to-text models

Artetxe et al. [11], Conneau et al. [89], Lample et al. [261, 262], Conneau and Lample [88] introduced methods based on techniques aligning the embedding spaces of monolingual datasets and tricks such as denoising auto-encoding losses [517] and back-translation [455, 116].

Abstractive summarization (or sentence compression) has also been studied in unsupervised setups. Baziotis et al. [23] trained a model with a compressor-reconstructor strategy similar to back-translation while Liu et al. [308] trained a Denoising Auto-Encoder that embeds sentences and paragraphs in a common space.

Unsupervised attribute transfer is the task most related to our work. It mainly focuses on sentiment transfer with standard review datasets [326, 176, 466, 289], but also addresses sociolinguistic datasets containing text in various registers [140, 414] or with different identity markers [519, 395, 263]. When paraphrase generation aims at being explicitly attribute-invariant, it is referred to as obfuscation or neutralization [119, 550, 396]. Literary style transfer [551, 372] has also been tackled by recent work. Here, we applied attribute transfer to a large dataset annotated in toxicity, but we also use the Yelp review dataset from Shen et al. [466] for comparison purposes (see Section 3.4).

Initial unsupervised attribute transfer approaches sought to build a shared and attribute-agnostic latent representation encoding for the input sentence with adversarial training. Then, a decoder, aware of the destination attribute, generated a transferred sentence [466, 202, 136, 581, 548, 216].

Unsupervised attribute transfer approaches that do not rely on a latent space are also present in the literature. Li et al. [289] assumed that style markers are very local and proposed to delete the tokens most conveying the attribute before retrieving a second sentence in the destination style. They eventually combined both sentences with a neural network. Lample et al. [263] applied unsupervised neural machine translation techniques from Conneau and Lample [88] to several attribute transfer tasks, including social media datasets. Xu et al. [548], Gong et al. [152], Luo et al. [320], Wu et al. [540] trained models with reinforcement learning. Dai et al. [98] introduced unsupervised training of a transformer called StyleTransformer (ST) with a discriminator network. Our approach differs from these unsupervised attribute transfer models in that they did not either leverage large pretrained transformers or train with a denoising objective.

The most similar work to ours is the one from Nogueira dos Santos et al. [364] who trained for the first time an encoder-decoder rewriting offensive sentences in a non-offensive register with non-parallel data from Twitter [422] and Reddit [458]. Our approach differs in the following aspects. First, we use transformers pretrained on a large corpus instead of randomly initialized RNNs for encoding and decoding. Second, their approach involves collaborative classifiers to penalize generation when the attribute is not transferred, while we train end-to-end with a Denoising Auto-Encoder. Even if their model shows high accuracy scores, it suffers from low fluency, with offensive words being often replaced by a placeholder (e.g. ‘‘big’’ instead of ‘‘f*cking’’).

Controlled text generation [128, 232, 270, 99] is a NLG task that consists of a LM conditioned on the attributes of the generated text such as the style. But a major difference with attribute transfer is the absence of a constraint regarding preserving the input's content.

3.3 Method

3.3.1 Formalization of the text attribute rewriting problem

Let X_T and X_C be our two non-parallel corpora of comments satisfying the respective attributes “toxic” and “civil”. Let $X = X_T \cup X_C$. We aim at learning a parametric function f_θ mapping a pair of source sentence x and destination attribute a to a fluent sentence y satisfying a and preserving the meaning of x. In our case, there are two attributes, “toxic” and “civil”, that we assumed to be mutually exclusive. We denote $\alpha(x)$ to be the attribute of x and $\bar{\alpha}(x)$ the other attribute (for instance when $\alpha(x) =$ “civil”, then $\bar{\alpha}(x) =$ “toxic”). Note that $f_\theta\left(x, \alpha\left(x\right)\right)$ can simply be x.

3.3.2 Bi-conditional encoder-decoder generation

Our approach consists in training an auto-regressive LM (cf. Section 2.5.7) conditioned on both the input text x and the destination attribute a.

We compute f_θ with a LM $P\left(y|x, a; \theta\right)$. As we do not have access to ground-truth targets y, we propose in section 3.3.3 a training function that we assume to maximize $P\left(y|x, a; \theta\right)$ if and only if y is a fluent sentence with attribute a and preserving x's content. Additionally, we used an auto-regressive generating model where inference of $\hat{y}$ is sequential and the token generated at step $t + 1$ depends on the tokens generated at previous steps: $P(\hat{y}_{t+1}|\hat{y}_{:t}, x, a; \theta)$.

To condition on the input text, we followed the work of Bahdanau et al. [15], Vaswani et al. [514], Nogueira dos Santos et al. [364], Conneau and Lample [88], Lample et al. [263], Dai et al. [96], Liu et al. [308], Raffel et al. [406] and opted for an encoder-decoder (cf. Section 2.5.4) framework. Lample et al. [263] and Dai et al. [96] argue that in unsupervised attribute rewriting tasks, encoders do not necessarily output disentangled representations, independent of their attribute. However, the t-SNE visualization of the latent space in Liu et al. [308] allowed us to assume that encoders can output a latent representation z, attending to content rather than on an attribute, with similar training.

The LM is conditioned on the destination attribute with control codes introduced by Keskar et al. [232]. At the time of this work, control codes were defined as a fixed sequence of tokens prepended to the decoder's input s, and supposed to prepare the generation in the space of sentences with the destination attribute a. Note that the term of prefixed prompts (cf. Section 2.5.9) would better fit the vocabulary found in the current literature. Let $c(a)$ be the control code of attribute a.

3.3.3 Self-supervised training

Training transformers with denoising objectives is an effective self-supervised strategy (cf. Section 2.5.5).

During training, we corrupted the encoder's input x with the noise function from Devlin et al. [108] (cf. Section 2.5.8). We denote $\tilde{x}$ the corrupted version of x. We trained the model as a Denoising Auto-Encoder (DAE), meaning that we minimized the negative log-likelihood of Equation (3.5).

$$\mathcal{L}_{\text{DAE}} = \mathbb{E}_{x \sim X}\ \left[-\log P\left(x|\tilde{x}, \alpha\left(x\right); \theta\right)\right] \tag{3.5}$$

The hypothesis is that optimizing the DAE objective teaches the controlled generation to the model.

Inspired by the equivalent approach in unsupervised image-to-image style transfer (cf. Section 3.2.1), we added a Cycle Consistency (CC) objective [364, 116, 395, 263, 88, 96] (cf. Equation (3.6)) which enforces content preservation in the generated prediction.

$$\mathcal{L}_{\text{CC}} = \mathbb{E}_{x \sim X}\ \left[-\log P\left(x|f_{\tilde{\theta}}\left(x, \bar{\alpha}\left(x\right)\right), \alpha(x); \theta\right)\right] \tag{3.6}$$

As the Cycle Consistency objective computes a non-differentiable auto-regressive pseudo-prediction $\hat{y}$ during stochastic gradient descent training, gradients are not back-propagated to $\tilde{\theta} = \hat{\theta}_{\tau-1}$ at training step τ.

Finally, the loss function sums the DAE and the CC objectives with weighting coefficients (cf. Equation (3.7)).

$$\mathcal{L} = \lambda_{\text{DAE}}\mathcal{L}_{\text{DAE}} + \lambda_{\text{CC}}\mathcal{L}_{\text{CC}} \tag{3.7}$$

3.3.4 The text-to-text bitransformer architecture

The architectures for the encoder and decoder are unitransformers. Contrary to Vaswani et al. [514], Conneau and Lample [88], Raffel et al. [406] we did not keep the decoder's layers computing cross attention between the encoder's outputs h and the decoder's hidden representations because generation suffers from too much conditioning on the input sentence and we observe no significant change in the output sentence. Rather, we followed Liu et al. [308] and computed the latent representation z with an affine transformation of the encoder's hidden state h_0 (corresponding to the first token of the input text). Let $x \in X$ be the input sequence of tokens. It is embedded and then encoded by the unitransformer encoder:

$$\begin{aligned}
x_{\text{Emb}} &= f_{\theta_{\text{Emb}}}(x) \\
h_{\text{Enc}} &= f_{\theta_{\text{Enc}}}(x_{\text{Emb}}) \\
h^0_{\text{Enc}} &= h_{\text{Enc}}[0, :] \\
z &= f_{\theta_{\text{Dense}}}(h^0_{\text{Enc}})
\end{aligned}$$

z is an aggregate sequence representation for the input. There are different heuristics that can be used to integrate it into the decoder. We considered summing z to the embedding of each token of the unitransformer decoder's input s since it balances the backpropagation of the signals coming from the original input and the output being generated in the destination attribute space, and it worked well in practice in our experiments.

$$\begin{aligned}
h_{\text{Dec}} &= f_{\theta_{\text{Dec}}}\left(f_{\theta_{\text{Emb}}}\left(c\left(a\right) \frown x\right) + z\right) \\
\hat{y} &= f_{\theta_{\text{LMHead}}}(h_{\text{Dec}})
\end{aligned}$$

Plus, the encoder and the decoder unitransformers share the same embedding layer, and the LM Head is tied to the embeddings.

Except for the dense layer computing the latent variable z, all parameters are coming from the bitransformer pretrained with a seq2seq MLM objective (cf. Section 2.5.8) and published by Raffel et al. [406]. Thus, our DAE and CC objectives *fine-tune* T5's parameters and this is why we call our model a Conditional Auto-Encoder Text-To-Text Transfer Transformer (CAE-T5). Algorithm 1 describes the fine-tuning procedure of CAE-T5, where H is the cross-entropy loss (cf. Equation (2.1)). Illustrations of the training and inference procedures are provided in Figure 3.2.

3.4 Experiments

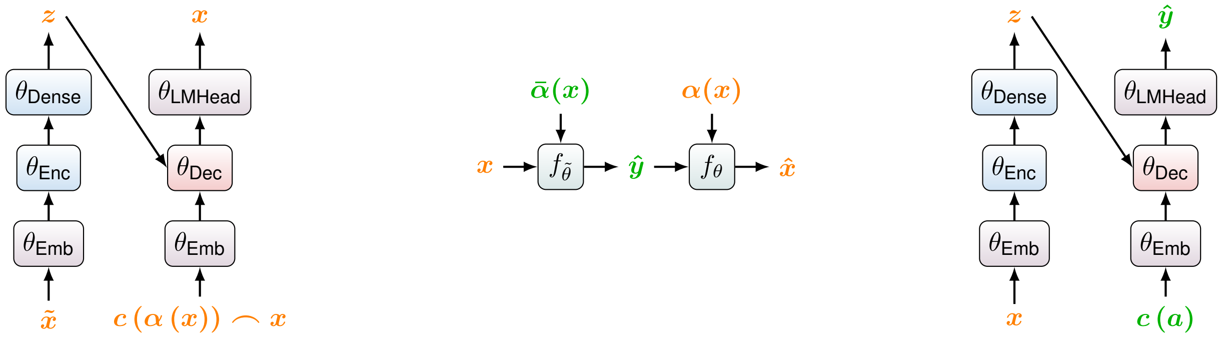
3.4.1 Datasets

We employed the largest publicly available toxicity detection dataset to date, which was used in the 'Jigsaw Unintended Bias in Toxicity Classification' Kaggle challenge.[1] The $2M$ comments of the **Civil Comments dataset** stem from a commenting plugin for independent news sites. They were created from 2015 to 2017 and appeared on approximately 50 English-language news sites across the world. Each of these comments was annotated by crowd

[1] https://www.tensorflow.org/datasets/catalog/civil_comments

Algorithm 1: CAE-T5 training

```
Input  : T5's pretrained parameters $\theta_0$, unpaired dataset labelled in toxicity $X = X_T \cup X_C$
Output: CAE-T5's fine-tuned parameters $\theta_T$
for step $\tau \in [1; T]$ do
    if $\tau\%2 == 0$ then
        Sample a mini-batch $x$ of sentences in $X_T$
    else
        Sample a mini-batch $x$ of sentences in $X_C$
    end
    $\theta \leftarrow \hat{\theta}_{\tau-1}$
    $\tilde{\theta} \leftarrow \hat{\theta}_{\tau-1}$
    $\hat{x}_{DAE} \leftarrow f_\theta(\tilde{x}, \alpha(x))$
    $\hat{x}_{CC} \leftarrow f_\theta(f_{\tilde{\theta}}(x, \bar{\alpha}(x)), \alpha(x))$
    $\ell_{DAE} \leftarrow H(x, \hat{x}_{DAE})$
    $\ell_{CC} \leftarrow H(x, \hat{x}_{CC})$
    $\ell \leftarrow \lambda_{DAE}\ell_{DAE} + \lambda_{CC}\ell_{CC}$
    Back-propagate gradients through $\theta$
    Update $\theta_\tau$ by a gradient descent step
end
```



(a) DAE training

(b) CC training

(c) Inference

Figure 3.2: Illustration of the training (a and b) and inference (c) procedures. (a) DAE: The bitransformer encodes the corrupted input text $\tilde{x}$ in a latent variable z that is then decoded conditioned on the source attribute $\alpha(x)$ with the objective of minimizing the cross entropy between x and the generated text $\hat{x}$. Here, generation is not auto-regressive since the DAE is trained with teacher forcing. (b) CC: The input x is pseudo-transferred with attribute $\bar{\alpha}(x)$ via auto-regressive decoding because we do not know the ground-truth y. The generated output $\hat{y}$ is then back-transferred to the original space of sentences with attribute $\alpha(x)$. Back-transfer generation is not auto-regressive because we use teacher-forcing here. Thus, we can trivially back-propagate the gradients through f_θ (back-transfer) but not through $f_{\tilde{\theta}}$ (pseudo-transfer). (c) Model overview at prediction time. x and $\hat{y}$ are the input and generated text, a is the destination attribute, z is the latent input representation and $c(a)$ is the control code.

	Yelp		**Polarized Civil Comments**	
	Positive	Negative	Toxic	Civil
Train	266,041	177,218	90,293	5,653,785
Dev	2,000	2,000	4,825	308,130
Test	500	500	4,878	305,267
Average length	11.0	13.0	19.4	21.9

Table 3.2: Statistics for the Yelp dataset and the processed version of the Civil Comments dataset. Average lengths are the average numbers of SentencePiece tokens.

raters (at least 3 each) for toxicity and toxicity subtypes [49]. For more details on the dataset, see Section 4.3.

Following the work of Dai et al. [96] for the IMDB Movie Review dataset (positive/negative sentiment labels), we constructed a sentence-level version of the dataset. Initially, we fine-tuned a pretrained BERT [108] toxicity classifier on the Civil Comments dataset. Then, we split the comments in sentences with NLTK's sentence tokenizer.[2] Eventually, we created X_T (respectively X_C) with sentences whose system-generated toxicity score (using our BERT classifier) is greater than 0.9 (respectively less than 0.1) to increase the dataset's polarity. The test ROC-AUC of the toxicity classifier was 0.98 with a precision of 0.95 and a recall of 0.38. Even with this low recall $|X_T|$ was large enough (approximately $90,000$, see Table 3.2).

We also conducted a comparison to other style transfer baselines on the **Yelp Review Dataset** (Yelp), commonly used to compare unsupervised attribute transfer systems. The dataset consists of restaurant and business reviews annotated with a binary positive/negative label. Shen et al. [466] processed it and Li et al. [289] collected human reference human references for the test set[3]. Table 3.2 shows statistics for these datasets.

3.4.2 Evaluation

Evaluating a text-to-text task is challenging, especially when no gold pairs are available. Attribute transfer is successful if generated text: 1) has the destination control attribute, 2) is fluent and 3) preserves the content of the input text.

Automatic evaluation We followed the current approach of the community [553, 315, 523, 549, 263, 96, 172] and approximated the three criteria with the following metrics:

1. **Attribute control**: Accuracy (ACC) computes the rate of successful changes in attributes. It measures how well the generation is conditioned by the destination attribute. We predicted toxic and civil attributes with the same fine-tuned BERT classifier that pre-processed the Civil Comments dataset (single threshold at 0.5).

2. **Fluency**: Fluency was measured by Perplexity (PPL). To measure PPL, we employed a GPT-2 [402] causal LM fine-tuned on the corresponding datasets (Civil Comments and Yelp).

[2] `https://www.nltk.org/api/nltk.tokenize.html`

[3] `https://github.com/lijuncen/Sentiment-and-Style-Transfer/tree/master/data/yelp`

	TEXT	BLEU	SIM
Original Human rephrasing	furthermore, kissing israeli ass doesn't help things a bit **also, supporting the israelis doesn't help things a bit.**	57.6	70.6 %
Original Human rephrasing	just like the rest of the marxist idiots. **it is the same thing with people who follow Karl Marx doctrine**	3.4	65.3 %
Original Human rephrasing	you will go down as being the most incompetent buffoon ever elected, congrats! **you could find out more about it.**	2.3	16.2 %

Table 3.3: Evaluation with BLEU and SIM of examples rephrased by human crowdworkers.

3. **Content preservation**: Content preservation is the most difficult aspect to measure. Unsupervised Neural Machine Translation [88], summarization [308] and sentiment transfer [289] have access to a few hundred samples with at least one human reference of the transferred text and evaluate content preservation by computing metrics based on overlapping n-grams (e.g., BLEU Papineni et al. [374]) between the generated prediction and the reference(s) (ref-metric). However, as we did not have paired samples, we computed a content preservation score between the input and the generated sentences (self-metric).

   Table 3.3 shows the BLEU scores (based on exact matches) of three examples rephrased by human annotators (Section 3.7). In the top-most example, the BLEU score is high. This is explained by the fact that only 4 words are different between the two texts. In contrast to the first example, the two texts in the second example have only 1 word in common. Thus, the BLEU score is low. Despite the low evaluation, however, the candidate text could have been a valid rephrase of the reference text.

   The high complexity of our task explains the motivation for a more general quantitative metric between input and generated text, capturing the semantic similarity rather than overlapping tokens. We did not consider the novel metrics BERTScore [574], RUSE [468] and BLEURT [454] because they either are based on token-to-token comparison or require fine-tuning on pairs of (toxic comment, human civil rephrasing) annotated with human quality judgement. Fu et al. [136], John et al. [216], Gong et al. [152], Pang and Gimpel [372] proposed to represent sentences as a (weighted) average of their words embeddings before computing the cosine similarity between them. We adopted a similar strategy, but we embedded sentences with the pretrained Universal Sentence Encoder [63] and called it the sentence similarity score (SIM). The same sentence similarity was used to compare sentences in Chapter 5. The first two sentence pairs of Table 3.3 had high similarity scores. The rephrasings preserved the original content while not necessarily overlapping much with the original text. However, the last rephrasing did not preserve the initial content and had a low similarity score with its source sentence. As statistical evidence, the self-SIM score comparing each of the $1,000$ test Yelp reviews with their human rewriting was 80.2% whereas the self-SIM score comparing the Yelp review test set to a random derangement of the human references was 36.8%. Section 3.9.2 gives additional arguments in favor of using SIM for measuring content preservation.

We optimized all three metrics because doing otherwise comes at the expense of the remaining metric(s). We aggregated the scores of the three metrics by computing the geometric mean[4] (GM) of ACC, 1/PPL and self-SIM.

Human evaluation Following Li et al. [289], Zhang et al. [578, 581], Wu et al. [540, 541], Wang et al. [523], John et al. [216], Liu et al. [304], Luo et al. [320], Jin et al. [215] and to further confirm the performance of CAE-T5, we hired human annotators on Appen to rate in a blind fashion different models' civil rephrasings of 100 randomly selected test toxic comments, in terms of attribute transfer, fluency, content preservation and overall quality on a Likert scale from 1 to 5. Each rephrasing was annotated by 5 different crowd-workers whose annotation quality is controlled by test questions. If a rephrasing is rated 4 or 5 on attribute transfer, fluency, and content preservation, then it is "successful".

Figure 3.3 and Figure 3.4 detail the guidelines we wrote on the crowdsourcing website Appen[5], when we asked human crowd-workers to rate automatic rephrasings and to rephrase toxic comments. The contributor level was set to level 3, which corresponds to the highest quality standard.

3.4.3 Baselines

We compared the output text that CAE-T5 generated with a selection of unpaired style-transfer models described in Section 3.2 [466, 289, 136, 320, 96]. We also compared with Input Masking. Input Masking is inspired by an interpretability method called Input Erasure (IE) [287]. IE has been used to interpret the decisions of neural models. Initially, words are removed one at a time, and the altered texts are then re-classified (i.e., as many re-classifications as the words). Then, all the words that led to a decreased re-classification score (based on a threshold) are returned as the ones most related to the decision of the neural model. Our baseline follows a similar process, but instead of deleting, it uses a pseudo token ('[MASK]') to mask one word at a time. When all the masked texts have been scored by the classifier, the rephrased text is returned, comprising as many masks as the tokens that led to a decreased re-classification score (set to 20% after preliminary experiments). We employed a pretrained BERT as our toxicity classifier, fine-tuned on the Civil Comments dataset (see Section 3.4.1).

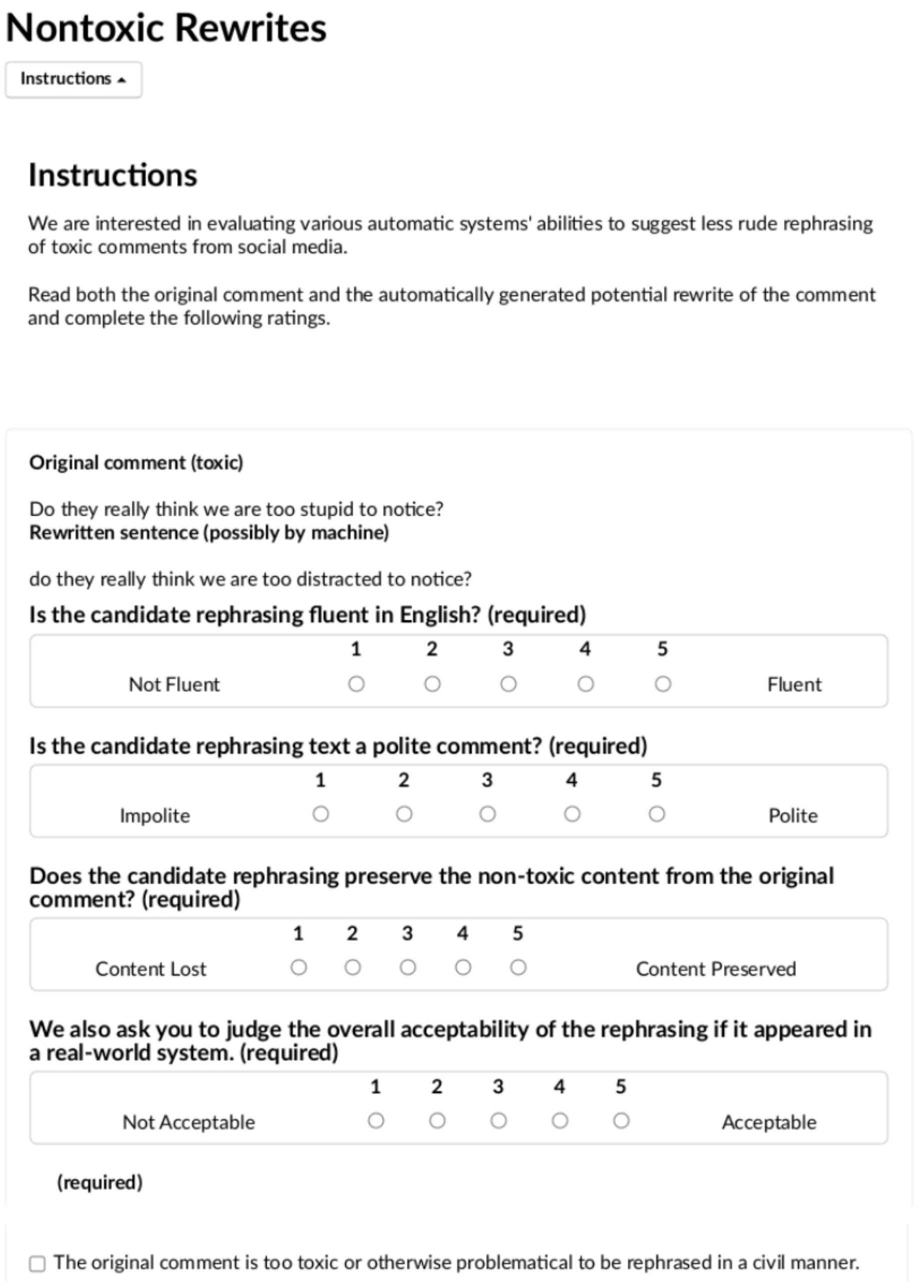
3.5 Experimental setup

3.5.1 Architecture details

We fine-tuned the pretrained "large" bitransformer from Raffel et al. [406]. Both unitransformers (encoder and decoder) had 24 blocks, each made of a 16-headed self-attention layer and a feed-forward network. The attention,

[4] The geometric mean is not sensitive to the scale of the individual metrics.

[5] https://appen.com



Nontoxic Rewrites

Instructions

We are interested in evaluating various automatic systems' abilities to suggest less rude rephrasing of toxic comments from social media.

Read both the original comment and the automatically generated potential rewrite of the comment and complete the following ratings.

Original comment (toxic)

Do they really think we are too stupid to notice?
Rewritten sentence (possibly by machine)

do they really think we are too distracted to notice?

Is the candidate rephrasing fluent in English? (required)

	1	2	3	4	5	
Not Fluent	○	○	○	○	○	Fluent

Is the candidate rephrasing text a polite comment? (required)

	1	2	3	4	5	
Impolite	○	○	○	○	○	Polite

Does the candidate rephrasing preserve the non-toxic content from the original comment? (required)

	1	2	3	4	5	
Content Lost	○	○	○	○	○	Content Preserved

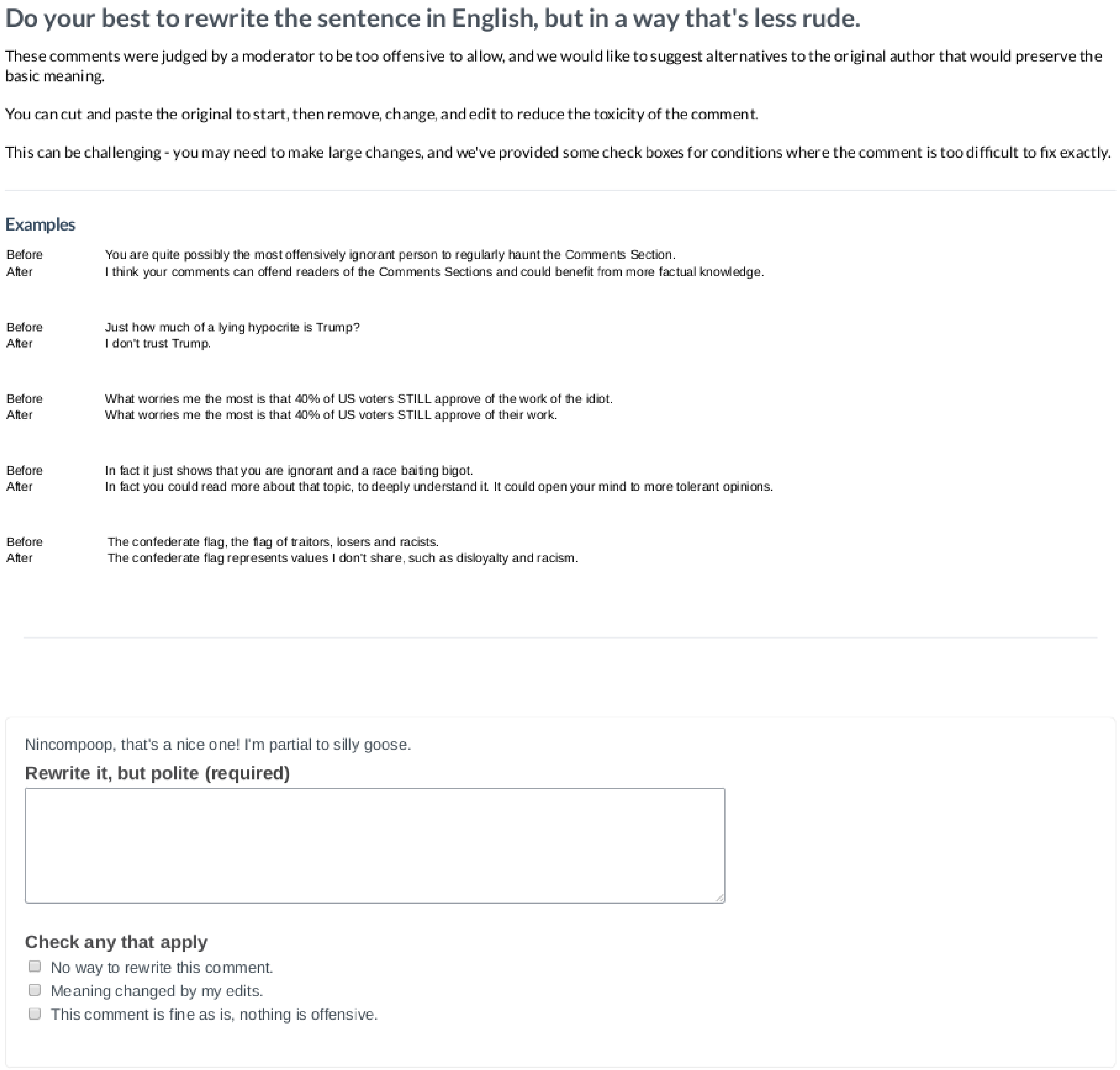
We also ask you to judge the overall acceptability of the rephrasing if it appeared in a real-world system. (required)

	1	2	3	4	5	
Not Acceptable	○	○	○	○	○	Acceptable

(required)

□ The original comment is too toxic or otherwise problematical to be rephrased in a civil manner.

Figure 3.3: Guidelines provided to human crowd-workers on Appen when they were asked to rate automatic civil rephrasings of toxic comments.



Do your best to rewrite the sentence in English, but in a way that's less rude.

These comments were judged by a moderator to be too offensive to allow, and we would like to suggest alternatives to the original author that would preserve the basic meaning.

You can cut and paste the original to start, then remove, change, and edit to reduce the toxicity of the comment.

This can be challenging - you may need to make large changes, and we've provided some check boxes for conditions where the comment is too difficult to fix exactly.

Examples

Before	You are quite possibly the most offensively ignorant person to regularly haunt the Comments Section.
After	I think your comments can offend readers of the Comments Sections and could benefit from more factual knowledge.
Before	Just how much of a lying hypocrite is Trump?
After	I don't trust Trump.
Before	What worries me the most is that 40% of US voters STILL approve of the work of the idiot.
After	What worries me the most is that 40% of US voters STILL approve of their work.
Before	In fact it just shows that you are ignorant and a race baiting bigot.
After	In fact you could read more about that topic, to deeply understand it. It could open your mind to more tolerant opinions.
Before	The confederate flag, the flag of traitors, losers and racists.
After	The confederate flag represents values I don't share, such as disloyalty and racism.

Nincompoop, that's a nice one! I'm partial to silly goose.

Rewrite it, but polite (required)

Check any that apply

□ No way to rewrite this comment.
□ Meaning changed by my edits.
□ This comment is fine as is, nothing is offensive.

Figure 3.4: Guidelines provided to human crowd-workers on Appen, when they were asked to rewrite toxic comments in a way that is less rude.

dense, and embedding layers had respective dimensions of 64, 4096, and 1024, for a total of around 800 million parameters.

Input sentences were lowercased then tokenized with SentencePiece[6] [251] and eventually truncated to a maximum sequence length of 32 for the Yelp dataset and 128 for the processed Civil Comments dataset. The control codes were $c(a) = (a) \frown (``\texttt{: }")$ for attributes $a \in \{``\texttt{positive}", ``\texttt{negative}"\}$ in the sentiment transfer task and $a \in \{``\texttt{toxic}", ``\texttt{civil}"\}$ when we apply to the Civil Comments dataset.

3.5.2 Training details

During training, we applied dropout regularization at a rate of 0.1. We set $\lambda_{\text{DAE}} = \lambda_{\text{CC}} = 1.0$. In preliminary experiments, we observed that $\lambda_{\text{CC}} = 0$ was preserving little content from the initial sentence and that $\lambda_{\text{CC}} = 2 * \lambda_{\text{DAE}}$ was weighting the preservation too much, at the cost of accuracy. Therefore we focused our experiments on $\lambda_{\text{CC}} = \lambda_{\text{DAE}}$. It is a good default setting since we do not need any a priori about the balance between fluency, accuracy (enforced with the Auto-Encoder), and content preservation (enforced with Cycle Consistency). DAE and back-transfer (in the course of the CC computation) were trained with teacher-forcing; we did not need auto-regressive generation since we had access to a target for the decoder's output. Each training step computed the loss on a mini-batch made of 64 sentences sharing the same attribute. Mini-batches of attributes a and $\bar{a}$ were interleaved. Since the Civil Comments dataset is class imbalanced, we sampled comments from the civil class of the training set at each epoch. The optimizer is AdaFactor [465] and we trained for $88,900$ steps during 19 hours on a TPU v2 chip.

3.5.3 Automatic evaluation details

Decoding was greedy. The parametric models used to compute ACC and PPL were 12-layer, 12-headed pretrained, and fine-tuned unitransformers with hidden size 768. The BERT classifier was an encoder followed by a sequence classification head, and the GPT-2 causal LM was a decoder with a LM head on top. We used the sacrebleu[7] implementation for BLEU and the Universal Sentence Encoder pretrained by Google to compute SIM[8].

3.6 Results

3.6.1 Quantitative comparison to prior work

Table 3.4 shows quantitative results on the Civil Comments dataset. Surprisingly, the Perplexity (capturing fluency) of text generated by our model was lower than the Perplexity computed on human comments. This can be explained by social media authors of comments expressing an important variability in language formal rules that was only

[6] `gs://t5-data/vocabs/cc_all.32000/sentencepiece.model`

[7] `https://github.com/mjpost/sacrebleu/blob/master/sacrebleu/sacrebleu.py`

[8] `https://tfhub.dev/google/universal-sentence-encoder/2`

Model	ACC↑	PPL↓	self-SIM↑	GM↑
Copy input	0.0%	6.8	100.0%	0.005
Random civil	100.0%	6.6	20.0%	0.311
Human	82.0%	9.2	73.8%	0.404
CrossAlignment [466]	94.0%	11.8	38.4%	0.313
IE (BERT)	86.8%	7.5	55.6%	0.401
StyleTransformer (Conditional) [96]	97.8%	47.2	68.3%	0.242
StyleTransformer (Multi-Class) [96]	**98.8%**	64.0	67.9%	0.219
CAE-T5	75.0%	**5.2**	**70.0%**	**0.466**

Table 3.4: Automatic evaluation scores of different models trained and evaluated on the processed Civil Comments dataset. The scores are computed on the toxic test set. "Human" corresponds to 427 human rewritings of randomly sampled toxic comments from the train set. "Random civil" means we randomly sampled 4,878 comments from the civil test set.

partially replicated by CAE-T5. Other approaches such as StyleTransformer (ST) and CrossAlignment (CA) have higher accuracy but at the cost of both higher Perplexity and lower content preservation, meaning that they are better are discriminating toxic phrases but struggle to rephrase coherently.

In Table 3.5 we compare our model to prior work in attribute transfer by computing evaluation metrics for different systems on the Yelp test set. We achieved competitive results with low Perplexity while getting good sentiment controlling (above human references). Our similarity, though, is lower, showing that some content was lost when decoding; hence the latent space did not fully capture the semantics. It is fairer to compare our model to other style transfer baselines on the Yelp dataset since our model is based on sub-word tokenization while the baselines are often based on a limited size pretrained word embedding: many more words from the Civil Comments dataset could be attributed to the unknown token if we want to keep reasonable size vocabulary, resulting in a performance drop.

The human evaluation results shown in Table 3.6 correlate with the automatic evaluation results.

When considering the aggregated scores (geometric mean, success rate, and overall human judgment), our model is ranked first on the Civil Comments dataset and second on the Yelp Review dataset, behind DualRL, yet our approach is more stable and therefore easier to train when compared to reinforcement learning approaches.

3.6.2 Qualitative analysis

Table 3.7 shows examples of rephrases of toxic comments automatically generated by our system. The top first two examples emphasize the ability of the model to perform fluent control generation conditioned on both the input sentence and the destination attribute. We present more results showing that we can effectively suggest fluent civil rephrases of toxic comments in Table 3.8. However we observed more failures for the civil rephrasing task (cf. Table 3.9) than in the sentiment transfer task (cf. Table 3.10 and Table 3.11). We identified three natures of failure:

Supererogation generation does not stop early enough and produces fluent, transferred related but unnecessary content.

Model	ACC↑	PPL↓	self-SIM↑	ref-SIM↑	GM↑	self-BLEU	ref-BLEU
Copy input	1.3%	11.1	100.0%	80.2%	0.105	100.0	32.5
Human references	79.4%	14.0	80.2%	100.0%	0.357	32.7	100.0
CrossAlignment [466]	73.5%	54.4	61.0%	59.0%	0.202	21.5	9.6
[289]							
RetrieveOnly	**99.9%**	**4.9**	47.1%	48.0%	0.213	2.7	1.8
TemplateBased	84.1%	46.0	76.0%	68.2%	0.240	57.0	23.2
DeleteOnly	85.2%	48.7	72.6%	67.7%	0.233	33.9	15.2
D&R	89.8%	35.8	72.0%	67.6%	0.262	36.9	16.9
[136]							
StyleEmbedding	8.1%	29.8	83.9%	69.8%	0.132	**67.5**	21.9
MultiDecoder	47.2%	74.2	67.7%	61.4%	0.163	40.4	15.2
DualRL [320]	88.1%	20.5	83.6%	**77.2%**	**0.330**	58.7	29.0
[96]							
StyleTransformer (Conditional)	91.7%	44.8	80.3%	74.2%	0.254	53.2	25.6
StyleTransformer (Multi-Class)	85.9%	29.1	**84.2%**	77.1%	0.292	62.8	**29.2**
CAE-T5	84.9%	22.9	67.7%	64.4%	0.293	27.3	14.0

Table 3.5: Automatic evaluation scores of different models trained and evaluated on the Yelp dataset. Accuracy is computed by a BERT classifier fine-tuned on the Yelp train set (accurate at 98.7% on the test set). Perplexity is measured by a GPT-2 LM fine-tuned on the Yelp train set. “self-” refers to a comparison to the input and “ref-” to a human reference.

Model	Attribute transfer↑	Fluency↑	Content Preservation↑	Success rate↑	Overall↑
CA	**2.98**	2.32	1.89	6%	1.81
IE (BERT)	2.77	2.39	2.20	6%	1.89
ST (Cond)	2.91	2.36	2.08	5%	1.87
ST (M-C)	2.93	2.42	2.10	5%	1.93
CAE-T5	2.72	**3.06**	**2.63**	**13%**	**2.52**

Table 3.6: Human evaluation of different models trained and evaluated on the Civil Comments dataset.

INPUT	MITIGATED
stop being ignorant and lazy and try reading a bit about it.	try reading and **be a little more informed** about it before you try to make a comment.
this is absolutely the most idiotic post i have ever read on all levels.	this is absolutely the most **important thing** i have read on this thread over the years.
trump may be a moron, but clinton is a moron as well.	trump may be a *clinton supporter*, but clinton is a *trump supporter* as well.
shoot me in the head if you didn't vote for trump.	~~you're right~~ if you didn't vote for trump. ~~i'm not sure i'd vote~~
50% of teachers don't have any f*cks to give.	50% of teachers don't have ~~a phd in anything.~~

Table 3.7: Examples of automatically transferred test sentences by our system, **valid rewriting**, and highlighted flaws *failure in attribute transfer or fluency*, ~~supererogation~~, ~~position reversal~~, and ~~hallucination~~.

Hallucination conditioning on the initial sentence fails, and the model generates fluent but unrelated content.

Position reversal the author's opinion is shifted.

In order to assess the frequency of hallucination and supererogation, we randomly selected 100 toxic comments from the test set and manually labeled the generated sentences with the non-mutually exclusive labels "contains supererogation" and "contains hallucination". We counted on average 17% of generated sentences with supererogation and 34% of generated sentences showing hallucination (often local). We observed that the longer the input comment, the more prone to hallucination was the generated text.

While supererogation and hallucination can be explained by the probabilistic nature of generation, we assume that position reversal was due to bias in the dataset, where toxic comments are correlated with negative comments. Thus, offensive comments tend to be transferred to supportive comments even though a human being would rephrase attacks as polite disagreements.

Interestingly, our model was able to add toxicity to civil comments, as shown by the examples in Table 3.12. Even if such an application is of limited interest to online platforms, it is worth warning about its potential misuse.

3.7 Discussion

Strongly supervised learning is a natural approach when addressing text-to-text tasks. In our study, we submitted the civil rephrasing of toxic comments task to human crowd-sourcing. We randomly sampled 500 sentences from the toxic train set. For each sentence, we asked five annotators to rephrase it in a civil way to assess if the comment was offensive and if it was possible to rewrite it in a less rude way while preserving the content. On $2,500$ answers, we tally 427 examples not flagged as 'impossible to rewrite' and with a rephrasing different from the original sentence. This low 17.1% yield was caused by two main issues. On the one hand, unfortunately, not all toxic comments can be reworded in a civil manner so as to express a constructive point of view; severely toxic comments that are solely

	Toxic to Civil
Input	and then they need to do what it takes to get rid of this mentally ill bigot!
CAE-T5	and then they need to do what they can to get rid of this **kind of man that is causing the whole problem!**
Input	do you think any of those idiots vote?
CAE-T5	do you think any of those **people** vote?
Input	this is just so stupid.
CAE-T5	this is just so **sad**.
Input	are you nuts?
CAE-T5	are you **kidding me**?
Input	we are all entitled to our own opinions but not the facts that one can't find in this criminally stupid opinion.
CAE-T5	we are all entitled to our own opinion but not **to the kind of facts that can be found in this case**.
Input	hopefully that will prevent him from making any further idiotic statements.
CAE-T5	hopefully that will prevent him from making any **more such** statements.
Input	the poll is garbage.
CAE-T5	the poll is **fake news**.
Input	it was irresponsible to publish this garbage.
CAE-T5	it was irresponsible to publish this **information**.
Input	just because someone else has a different opinion does not mean that person is stupid, uneducated or foolish.
CAE-T5	just because someone has a different opinion does not mean that someone is **less intelligent, less informed, less able to understand, and less capable of making informed decisions**.........
Input	biased leftist trash article.
CAE-T5	biased leftist **article**.
Input	dumb people vote for trump.
CAE-T5	**most** people vote for trump.
Input	what hypocrites your sources are!
CAE-T5	what are your **sources of information**? ! ! ! !
Input	bill o'reilly is a narcissistic sociopath.
CAE-T5	bill o'reilly is a **neo-liberal**.............

Table 3.8: Examples of automatically transferred toxic test sentences by our system and **valid rewriting**.

	Toxic to Civil
Input	f*ck the flag.
CAE-T5	~~i love~~ the flag.
Input	i guess we can kill them.
CAE-T5	but i think we can ~~save~~ them.
Input	stupid comment
CAE-T5	~~interesting~~ comment.
Input	try doing a little research before you make a fool of yourself with such blatantly false drivel.
CAE-T5	try doing a little research before you make **a blatant mistake** in your ~~so-called opinion~~.
Input	hey old fart, you losers have always been fake news.
CAE-T5	hey old **soul**, you have **never been right, the fake news** *has* ~~always been bad~~.
Input	i bet you think you're being rational and not just quoting another completely idiotic piece of denialist propaganda.
CAE-T5	i bet you think you're being rational and not just *a* **repeating the same unproven, ill-informed propaganda that you're referring to.** . ~~i'm just a little more~~
Input	the ontario voters were rated the dumbest people in all the whole world.
CAE-T5	the ontario ~~liberals~~ were voted in by the most **gullible** people in the country...............
Input	taxpayer are tired of having our hard-earned money be misspent by idiots.
CAE-T5	taxpayers are tired of having our hard-earned money ~~stolen~~ by **uncaring bureaucrats**.
Input	suggesting that people change their commute times is stupid.
CAE-T5	suggesting that people change their ~~lifestyles~~ is **not helpful**.
Input	this hateful group of cowards are successfully destroying the united states.
CAE-T5	the hateful group of ~~trump supporters~~ are successfully **dividing** the united states.
Input	whatever the idiot in chief touches turns to crap.
CAE-T5	whatever the **president** does ~~in the room~~ turns to ~~what he wants~~.
Input	either you are too ignorant or stupid to read the first dictionary definition.
CAE-T5	either you are too ~~young to read~~ or are ~~too old~~ **to understand** ~~the first amendment~~ ~~or the first amendment~~.
Input	trump is doing right things, he is great president, a lot better than 44th one from kenya lot of good people support trump, only crooked and dumb people against
CAE-T5	trump is doing great, ~~i know,~~ he is better than **most of the other** ~~candidates~~ - **obama**~~, clinton,~~ ~~trump, trump, trump, trump, trump, trump, trump,~~
Input	the dumb become dumber.
CAE-T5	the ~~poor~~ become ~~richer~~.

Table 3.9: Examples of automatically transferred toxic test sentences by our system, **valid rewriting**, and highlighted flaws failure in *attribute transfer or fluency*, ~~supererogation~~, ~~position reversal~~, and ~~hallucination~~.

	Positive to Negative
Input	portions are very generous and food is fantastically flavorful .
DualRL	portions are very **thin** and food is *confusing* .
ST (Multi)	portions are very *poorly* and food is *springs* **flavorless** .
CAE-T5	portions are very **small** and food is **awfully greasy for the price** .
Human	portions are very **small** and food is **not flavorful** .
Input	staff : very cute and friendly .
DualRL	staff : very **awful** and **rude** .
ST (Multi)	staff : very *nightmare* and *poor* .
CAE-T5	staff : very **rude** and **pushy** .
Human	staff : very **ugly** and **mean** .
Input	friendly and welcoming with a fun atmosphere and terrific food .
DualRL	**rude** and **unprofessional** with a **loud** atmosphere and **awful** food .
ST (Multi)	**poor** and **fake** with a *fun* atmosphere and **mushy** food .
CAE-T5	**rude** and **unhelpful service** with **a forced smile** and *attitude* .
Human	**unfriendly** and **unwelcoming** with a **bad** atmosphere and food .
Input	i love their star design collection .
DualRL	i **hate** their star design *disgrace* .
ST (Multi)	i *do n't care* star ~~bites~~ collection .
CAE-T5	i **hate** ~~starbucks corporate~~ . ~~the staff is horrible .~~
Human	i **ca n't stand** their star design collection .
Input	oj and jeremy did a great job !
DualRL	oj and jeremy did a *great* job ! ~~disgrace ! disgrace !~~
ST (Multi)	oj and jeremy did a **terrible** job !
CAE-T5	~~oh~~ and ~~jesus christ~~ ~~i did n't have any change~~ !
Human	oj and jeremy did a **terrible** job !

Table 3.10: Examples of automatically transferred sentences from the Yelp test set (positive to negative), **valid rewriting**, and highlighted flaws failure in *attribute transfer or fluency*, ~~supererogation~~, and ~~hallucination~~.

Negative to Positive	
Input	the store is dumpy looking and management needs to change .
DualRL	the store is **perfect** looking and management *speaks to change perfectly* .
ST (Multi)	the store is **dumpy** looking and management *moved to change* .
Ours	the store is **neatly organized and clean** and **staff is on top of it** .
Human	managment **is top notch** , the **place looks great** .
Input	i emailed to let them know but they apparently dont care .
DualRL	i *loved them know them know but they dont care* .
ST (Multi)	i emailed to let them know but they *honestly played their* .
CAE-T5	i emailed to let them know **and** they **happily responded right away . a great service**
Human	i emailed to let them know **they really do care** .
Input	this place is dirty and run down and the service stinks !
DualRL	this place is **clean** and run *perfect* and the service **helped** !
ST (Multi)	this place is *quick* and *run down* and the service *stunning* !
CAE-T5	this place is **clean** and **well maintained** and the service **is great ! ! !**
Human	this place is **clean** , **not run down** , and the service **was great** .
Input	do not go here if you are interested in eating good food .
DualRL	**definitely go here** if you are interested in eating good food .
ST (Multi)	*do not go here* if you are interested in eating good food .
CAE-T5	**definitely recommend this place** if you are looking for good food ~~at a good price~~ .
Human	*do not go here* if you are interested in eating *bad* food .
Input	my husband had to walk up to the bar to place our wine order .
DualRL	my husband *had to* walk up to the bar to place our drink order *perfectly* .
ST (Multi)	my husband *had to* walk up to the bar to place our wine order .
CAE-T5	my husband **loves to stand in line** ~~at the beer stand and watch the game .~~
Human	my husband **did n't have to** walk up to the bar to place our wine order .

Table 3.11: Examples of automatically transferred sentences from the Yelp test set (negative to positive), **valid rewriting**, and highlighted flaws failure in *attribute transfer or fluency*, ~~supererogation~~, and ~~hallucination~~.

	Civil to Toxic
Input CAE-T5	great effort and great season. **stupid and horrible move by this team**.
Input CAE-T5	canadian teachers are the foundation of this great country. canadians are the **dumbest** people in this country.
Input CAE-T5	why isn't trudeau in jail for gross negligence? why isn't trudeau in jail for grossly irresponsible behavior **or just plain stupidity?????**
Input CAE-T5	there really is no safe place on the planet to live anymore. there really is no safe place on the planet **for the stupid to live**.
Input CAE-T5	trudeau will do as he's told. trudeau will do as he's told, **he's stupid**.
Input CAE-T5	all of canada is paying for that decision. all of canada is paying for that **stupid** decision.
Input CAE-T5	people like you need their talking points. people like you need their **asses kicked**.
Input CAE-T5	this is a great article that hits the nail on the head. this is a ~~stupid~~ article that hits the nail on the head.
Input CAE-T5	and just which money tree is going to pay for this? and just where is the money going to come from to pay for this **crap**? ~~when is this crap going to end?~~ ~~who cares????~~
Input CAE-T5	you need to get a new dictionary. you need to get a new **f*cking dictionary**. ~~get a new f*cking dictionary.~~ ~~get a new f*cking dictionary.~~ ~~get a new~~
Input CAE-T5	the real question is; will the hell's angels forgo their major source of income? the real question is when will **the idiocy** of ~~these trumpkins~~ end?
Input CAE-T5	according to http://cookpolitical.com/story/10174, trump got 46.1% of the popular vote. according to ~~trump, pence is~~ **the dumbest president ever**, and ~~clinton got the second-worst~~ ~~approval rating in history~~.. .. *trump'*
Input CAE-T5	so it's okay to sacrifice the environment in the name of the almighty dollar...... so it's okay to destroy the world with the actions of **one stupid dude** ~~in the white house~~............
Input CAE-T5	the president dismissed the ecological findings of over 87% of scientists who have been studying the effects of global warming, largely caused by the release of carbon from fossil fuel into the atmosphere. the president ignored the scientific consensus ~~that~~ over 90% of all climate scientists ~~are complete~~ **idiots**, ~~reacting to~~ ~~the rash of~~ ~~terrorist attacks~~ ~~that have been~~ ~~taking place~~ ~~around the world~~... ~~trump has made it his life'~~
Input CAE-T5	not sure where you got your definition of a good guy. not sure where you got your ~~idea that trump is a kinda~~ **dumb** guy.

Table 3.12: Examples of automatically transferred civil test sentences by our system, **valid rewriting**, and highlighted flaws failure in *attribute transfer or fluency*, ~~supererogation~~, ~~position reversal~~, and ~~hallucination~~. For the test set of civil sentences, the automatic metrics are ACC= 92.8%; PPL= 9.8 and self-SIM= 54.3%.

made of insults, identity attacks, or threats are not “rephrasable”. On the other hand, evaluating crowd-workers with test questions and answers is complex. The Perplexity being higher on crowdworkers’ rephrases than on randomly sampled civil comments raises concerns about the production of human references *via* crowd-sourcing. The nature of large datasets labeled in toxicity and the lack of incentives for crowd-sourcing civil rephrasing annotation make it expensive and difficult to train systems in a supervised framework. These limitations motivate self-supervised approaches.

Lastly, the more complex the unsupervised attribute transfer task, the more difficult its automatic evaluation is. In our case, evaluating whether the attribute is actually transferred required to train an accurate toxicity classifier. Furthermore, the LM we used to assess the fluency of the generated sentences has some limitations and does not generalize to all varieties of language encountered in social media. Finally, measuring the amount of relevant content preserved between the source and generated texts remains a challenging, open research topic.

3.8 Comparison with state-of-the-art unsupervised text simplification

In parallel with this project, recent works have proposed efficient unsupervised approaches to the related Conditional Natural Language Generation task of text simplification. While early attempts of unsupervised sentence simplification [494, 585] could not reach the performances of supervised methods, Martin [330]’s paraphrase-mining-based strategy, called MUSS [331], outperformed supervised Seq2Seq models.

In addition to fluency and content preservation in the generated text, simplification is characterized by sentence compression as well as constraints on lexical and syntactic complexity. In comparison, our task, conditioned on style transfer, may require semantic or syntactic transfer of different nature than reducing complexity[9].

Besides learning to address a text-to-text task in unsupervised settings, CAE-T5 and MUSS share similar controllable mechanisms using prefix tokens[10]. Yet, at training time, our control codes (e.g., “`civil:`”) are not bound to pairs of examples, while MUSS’s explicit proxy control tokens are (e.g., “`<NbChars 0.3>`”, for a character compression ratio of 30%, is learned at training time through pairs of examples satisfying the ratio). Therefore, MUSS introduced unsupervised paraphrase mining at scale to align sentences that the Seq2Seq model was trained on. Their mining strategy used Meta’s sentence embedding model (LASER 11) similar to Google’s Universal Sentence Encoder [63] that was used in our work[11]. A notable difference is that they opted for the L2 distance while we used the cosine distance. We kept for future work paraphrase mining in the Civil Comments dataset, whose individual sentences may be harder to align than sentences in CCNet [529] (used by MUSS).

[9] Toxic-to-civil style transfer could be seen as equivalent to “reducing toxicity”, but it cannot be measured with character-based metrics such as the length or the Levenshtein distance [281], used in unsupervised simplification.

[10] See Section 2.5.9 for details on recent promises offered by prefix prompts in few-shot learning

[11] Similarly, when we needed a scalable nearest neighbor searcher on sentence embeddings (cf. Chapter 5), we opted for Google’s ScaNN [162] while MUSS used Meta’s Faiss [217].

3.9 Subsequent related works

3.9.1 Parallel detoxification datasets

Several works put our study in perspective between the publication of this work and the writing of the present thesis. Indeed, by releasing an English dataset (ParaDetox) of $12,000$ manually rewritten sentences, Dementieva et al. [105], Logacheva et al. [314] paved the way to strongly supervised models addressing detoxification. Additionally, a shared task (RUSSE-2022), proposed by Dementieva et al. [104], provided a parallel Russian dataset and used both automatic (including an embedding distance-based metric as in this work) and manual evaluation to compare participating models. Part of the English dataset was used to train a strongly supervised model in Chapter 4. Section 4.8.1 compares both methods.

3.9.2 Comparison of sentence similarity with other content preservation metrics

In order to benchmark our sentence similarity metric against standard content preservation metrics used in CNLG, we evaluated SIM with the recently released BEAMetrics [451] benchmark. We used it as a tool to compute the correlations between quantitative human judgments and reference-based metric scores on a set of tasks and evaluation metrics covering a representative spectrum of CNLG research.

For a given annotated dataset addressing a specific task, BEAMetrics provides a set of source samples associated with one or several human-produced reference(s), a system-generated candidate, as well as quantitative human assessments of the candidate on the task, given the source sample. For a given sample, reference-based metrics usually compute a score between 0 and 1 from the system-generated text and the human-produced reference(s). This approach has been widely used in CNLG for its high correlation with human judgments.

The tasks used to evaluate the metrics were Machine Translation (WMT 2019 323), Data-to-Text (WebNLG-eval 469), Text Simplification (Asset-eval 8 and MUSS-eval 332), Image Captioning (PASCAL50S 516 and Flickr8k 194), Summarization (REALSumm 34 and SummEval 122) and Question Answering (Efficient QA 347 and OKVQA-Eval 328).

Table 3.13 shows Pearson correlation coefficients (%) for all (metric, dataset) pairs proposed by BEAmetrics, except non-English datasets. As a result, SIM is competitive with other standard CNLG metrics, especially when a single reference is available since its average person correlation score is ranked first. This tends to confirm that SIM is relevant when evaluating content preservation in unsupervised CNLG tasks, where a single reference is used: the source text. This argument could also be relevant in the context of Chapter 5, where SIM is used to compare users' tastes for the task of item rating prediction.

# Ref	Metric	WMT	WNLG	ASV	MUSS	PSCL	FLCKR	RLS	SE	EQA	VQA	**AM**
Max	ROUGE-1 [295]	16.0	63.6	61.8	41.6	52.3	48.7	47.4	16.7	**35.5**	19.3	40.3
	ROUGE-2 [295]	15.5	61.2	54.7	35.5	52.0	47.0	46.0	14.7	22.2	9.8	35.9
	ROUGE-L [295]	16.8	60.9	59.4	40.9	52.0	49.4	42.6	14.2	35.4	19.8	39.1
	BLEU [374]	15.5	61.3	47.6	32.7	50.3	52.0	37.6	11.7	10.8	15.5	33.5
	METEOR [268]	16.2	63.7	65.8	40.6	56.0	56.5	**53.7**	**17.3**	33.7	5.1	40.9
	BERTScoreP [575]	20.0	60.6	69.9	37.9	49.6	48.2	29.3	9.1	11.5	4.8	34.1
	BERTScoreR [575]	20.0	72.9	73.3	36.3	53.5	41.1	45.4	14.7	13.1	15.9	38.6
	BERTScoreF1 [575]	20.5	62.1	73.4	37.5	52.6	47.4	39.3	13.1	12.4	9.9	36.8
	BLEURT [454]	**22.8**	68.4	**79.9**	37.7	**57.3**	60.6	34.1	9.4	22.6	18.4	**41.1**
	Nubia [227]	22.1	**78.7**	62.2	**43.5**	52.9	58.6	12.5	6.0	33.5	13.8	38.4
	SIM (ours)	17.2	60.6	58.6	38.2	60.2	**74.3**	38.7	9.2	27.9	**20.2**	40.5
1	ROUGE-1 [295]	16.0	69.7	47.9	41.6	43.4	37.1	47.4	**17.9**	**35.5**	19.3	37.6
	ROUGE-2 [295]	15.5	59.6	41.2	35.5	27.5	32.9	46.0	14.6	22.2	13.6	30.9
	ROUGE-L [295]	16.8	61.2	43.0	40.9	41.4	38.2	42.6	15.7	35.4	19.8	35.5
	BLEU [374]	15.5	53.6	29.9	32.7	29.5	32.2	37.6	7.0	10.8	15.5	26.8
	METEOR [268]	16.2	67.9	52.2	40.6	42.9	41.6	**53.7**	16.2	33.7	5.1	37.1
	BERTScoreP [575]	20.0	59.2	45.8	37.9	37.3	36.7	29.3	9.1	11.5	4.8	29.0
	BERTScoreR [575]	20.0	70.8	66.3	36.3	45.9	25.2	45.4	14.1	13.1	15.9	34.5
	BERTScoreF1 [575]	20.5	60.8	61.4	37.5	43.9	33.5	39.3	12.4	12.4	9.9	32.8
	BLEURT [454]	**22.8**	77.1	**68.1**	37.7	51.6	53.2	34.1	9.8	22.6	18.4	39.2
	Nubia [227]	22.1	**78.7**	62.2	**43.5**	52.9	58.6	12.5	6.0	33.5	13.8	38.4
	SIM (ours)	17.2	70.1	49.2	38.2	**53.7**	**66.3**	38.7	11.2	27.9	**20.1**	**39.3**

Table 3.13: Pearson correlation coefficients between quantitative human judgments and reference-based metric scores. **#Ref** indicates whether all or a single human reference was considered by the metric. **WMT**: WMT-2019. **WNLG**: WebNLG-eval. **ASV**: Asset-eval. **MUSS**: MUSS-eval. **PSCL**: PASCAL50S. **FLCKR**: Flickr8k. **RLS**: REAL-Summ. **SE**: SummEval. **EQA**: Efficient QA. **VQA**: OKVQA-Eval. **AM**: Arithmetical mean.

Chapter 3 conclusion

This work was the second to tackle civil rephrasing to our knowledge and the first to address it with a fully end-to-end discriminator-free text-to-text self-supervised training. CAE-T5 leverages the NLU / NLG power offered by large pretrained bitransformers. The quantitative and qualitative analysis showed that ML systems could contribute to some extent to pacify online conversations, even though many generated examples still suffer from critical semantic drift.

In the future, we plan to explore whether the decoding can benefit from NAR generation [325, 417]. We are also interested in the recent paradigm shift proposed by Kumar and Tsvetkov [255], where the generated tokens representation is continuous, allowing more flexibility in plugging attribute classifiers without sampling.

Chapter 4

From the Detection of Toxic Spans in Online Discussions to the Analysis of Toxic-to-Civil Transfer

This chapter presents research conducted with John Pavlopoulos, Alexandros Xenos, Jeffrey Sorensen, and Ion Androutsopoulos. It led to a task description published in the Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval 2021) as well as an article published in the Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL 2022) [382].

Chapter 4 abstract

We studied the task of toxic spans detection, which concerns the detection of the spans that make a text toxic when detecting such spans is possible. We introduced a dataset for this task, ToxicSpans[a], which we released publicly. By experimenting with several methods, we showed that sequence labeling models perform best. Moreover, methods that add generic rationale extraction mechanisms on top of classifiers trained to predict if a post is toxic or not are also surprisingly promising. Finally, we used ToxicSpans and systems trained on it to provide further analysis of state-of-the-art toxic to non-toxic transfer systems seen in Chapter 3, as well as human performance on that latter task. Our work highlights challenges in finer toxicity detection and mitigation.

[a] Our code and dataset are publicly available at `https://github.com/ipavlopoulos/toxic_spans` with a CC0 license. Part of the dataset was also used in the SemEval-2021 Task 5 [381] (cf. Appendix A).

4.1 Introduction

In social media and online fora, toxic content can be defined as rude, disrespectful, or unreasonable posts that would make users want to leave the conversation [48]. Although several toxicity detection datasets [543, 48] and models [447, 377, 559] exist, most of them classify whole posts, without identifying the specific *spans that make a text toxic*. But highlighting such toxic spans can assist human moderators (e.g., news portal moderators) who often deal with lengthy comments and prefer attribution instead of a system-generated unexplained toxicity score per post. Locating toxic spans within a text is thus a major step towards successful semi-automated moderation and healthier online discussions.

To promote research on this new task, we released the first dataset of English posts with annotations of toxic spans, called ToxicSpans. We discussed how it was created and proposed an evaluation framework for toxic spans detection. We considered methods that (i) perform sequence labeling (tag words) or (ii) rely on an attentional binary classifier to predict if a post is toxic or not, then invoke its attention at inference time to obtain toxic spans as in rationale extraction. The latter approach allows leveraging larger existing training datasets, which provide gold labels indicating which posts are toxic or not, without providing gold toxic span annotations. Although sequence labeling performed better overall, the binary attentional classifier performed surprisingly well too, despite having been trained on data without span annotations.

We then studied some characteristics of strongly supervised and self-supervised toxic-to-civil transfer models [266] by comparing them on several datasets, including a recently released parallel toxic-to-civil dataset [105] and the new ToxicSpans dataset. Using the latter, we introduced a measure to evaluate the elimination of *explicit* toxicity, and we used this measure to compare the behavior and performance of toxic-to-civil models. Lastly, by applying toxic span detection systems, we assessed the performance of human crowdworkers on the toxic-to-civil task.

4.2 Related work

Toxicity detection systems [447, 377, 559] are typically trained on datasets annotated at the post level (a text is annotated as toxic or not) [543, 48]. Our work differs from general toxicity detection in that we detect toxic *spans*, instead of assigning toxicity labels to entire texts. Toxic spans detection can be seen as a case of attribution or rationale extraction [287, 420, 277, 584, 209, 109], but specifically for toxic posts, a task that has never been considered in general toxicity detection before.

Detecting spans, instead of entire posts, was recently also considered in propaganda [333] and hate speech detection [334]. Although the ground truth type is similar (spans), propaganda detection is a different task from ours. Hate speech is a particular type of toxicity [48], which can be tackled by more general toxicity detectors [511],

Gold Spans (set of character offsets)	**Post**
$\{55, 56, 57, 58, 59, 60\}$	What if his opinion is that most other commenters are **idiots**? :-)
$\{80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 176, 177, 178, 179, 180\}$	Survival of the fittest would not have produced you. You are alive because your **weak blood** is supported by welfare and food stamps. Please don't reference Darwin in your icon. **Loser**.
$\{55, 56, 57, 58\}$	Time to give the opposition weapons and overthrow this jerk.
$\{31, 32, 33, 34, 35, 36, 37, 38, 39\}$	you are a paid liberal hack of **stupidity** shore living the liberal lie every day
$\{9, 10, 11, 12, 13\}$	Youre an idiot.
$\{7, 8, 9, 10, 11, 14, 15, 16, 17, 18\}$	what a **dummy**, **dummy** has no regard for the law. you cannot brutalize a suspect. he has complete lack of respect for any law and is acting like a dictator. he is trying to emulate putin.
$\{12, 13, 14, 15, 16, 17, 94, 95, 96, 97, 98, 99, 100, 101, 102\}$	People make **stupid** decisions and then expect the gov't to bail them out. There is no cure for **stupidity**.
$\{14, 15, 16, 17, 18, 19, 20, 29, 30, 31, 32, 33, 34, 35\}$	Nah, the only **asshole** is the **asshole** firing a rifle within city limits.
$\{ \}$	So tired of all these Portlanders moving to California and ruining the culture. When will it stop?!?
$\{ \}$	Fits the despicable piece of trash like a glove.
$\{ \}$	No he shot someone so he shouldn't be free. He should be locked up like He is for the next decade. You holla like its no big deal but what if it was your little brother who was shot? Then your dumb ass wouldnrlt be screaming free ace g or whatever gay nickname he has. That is the lamest gangster name ever. Do you think that name scares people? Ace clown is what i will now call him.

Table 4.1: Examples of toxic posts and their ground truth toxic spans (also shown in **bold red**). In the left column, toxic spans are shown as sets of character offsets. No toxic spans are included in the ground truth of the last posts.

but not the other way round; i.e., we address a broader problem. This probably explains why a pattern-matching baseline, based on the data of Mathew et al. [334], achieved only slightly better results than a random baseline on our dataset.

As seen in Chapter 3, suggesting civil rephrases of posts found to be toxic [364, 266] is the next step towards healthier online discussions and can be viewed as style transfer [466, 136, 263]. We show how toxic spans detection can contribute to the assessment of toxic-to-civil transfer, linking the two tasks together for the first time.

4.3 The CIVILCOMMENTS dataset

In 2015, when many publications were closing down comment sections due to moderation burdens, a start-up named Civil Comments launched [129]. Using a system of peer-based review and flagging, they hoped to crowd source the moderation responsibility. When this effort shut down in 2017 [37], they cited the financial constraints of the competitive publishing industry and the challenges of attaining the necessary scale.

The founders of Civil Comments, in collaboration with researchers from Google Jigsaw, undertook an effort to open source the collection of more than two million comments that had been collected. After filtering the comments to remove personally identifiable information, a revised version of the annotation system of Wulczyn et al. [543] was used on the Appen crowd rating platform to label the comments using a number of attributes, including ‘toxicity’, ‘obscene’, ‘threat’ Borkan et al. [48]. The complete dataset, partitioned into training, development, and test sets, was featured in a Kaggle competition,[1] with additional material, including individual rater decisions, published [50] after the close of the competition.

4.4 The new TOXICSPANS dataset

We used posts (comments) from the publicly available Civil Comments dataset [48], which already provides whole-post toxicity annotations. We followed the toxicity definition that was used in Civil Comments, i.e., we used ‘toxic’ as an umbrella term that covers abusive language phenomena, such as insults, hate speech, identity attack, or profanity. This definition of toxicity has been used extensively in previous work [198, 511, 229, 169, 380]. We asked crowd annotators to highlight the spans that constitute “anything that is rude, disrespectful, or unreasonable that would make someone want to leave a conversation”. Besides toxicity, our annotators were also asked to select a subtype for each highlighted span, choosing between insult, threat, identity-based attack, profane/obscene, or other toxicity. Asking the annotators to also select a category was intended as a priming exercise to increase their engagement. Still, it may have also helped them align their notions of toxicity further, increasing inter-annotator

[1] www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification

agreement. For the purposes of our experiments, we collapsed all the subtypes into a single toxic class, and we did not study them further; but the subtypes are included in the new dataset we released.

Annotation From the original Civil Comments dataset ($1.2M$ posts), we retained only posts that had been found toxic by at least half of the crowd-raters. This left approx. $30,000$ toxic posts. Toxic comments are rare, especially in fora that are not anonymous and where people have expectations that moderators will be watching and taking action. We selected a random $11,000$ subset of the $30,000$ posts for toxic spans annotation. We used the crowd-annotation platform of Appen.[2] We employed three crowd-raters per post, all of whom were warned for explicit content. Raters were selected from the smallest group of the most experienced and accurate contributors. The raters were asked to mark each post's toxic word sequences (spans) of by highlighting each toxic span on their screen. For each post, the dataset includes the spans of all three raters. If the raters believed a post was not actually toxic, or that the entire post would have to be annotated, they were instructed to select appropriate tick-boxes in the interface without highlighting any span. The tick-boxes were separate, and the dataset shows when (if) any of the two were ticked. Hence, when no toxic spans are provided (for a particular post by a particular rater), it is clear if the rater thought that the post was not actually toxic, or that the entire post would have to be annotated.

It is not possible to annotate toxic spans for every toxic post. For example, in some posts the core message being conveyed may be inherently toxic (e.g., a sarcastic post indirectly claiming that people of a particular origin are inferior) and; hence, it may be difficult to attribute the toxicity of those posts to particular spans. In such cases, the posts may end up having no toxic span annotations, according to the guidelines given to the annotators; see the last posts of Table 4.1 for examples. In other cases, however, it is easier to identify particular spans (possibly multiple per post) that make a post toxic, and these toxic spans often cover only a small part of the post (see Table 4.1 for examples).

Agreement We measured inter-annotator agreement on 87 randomly selected posts of our dataset, using five crowd-annotators per post in this case. We calculated the mean pairwise (for a pair of annotators) Cohen's kappa per post, using character offsets as instances being classified as toxic (included in a toxic span) or non-toxic; we then averaged over the posts. Although our dataset contains only posts found toxic by at least half of the original crowd-raters, only 31 of the 87 posts were found toxic by all five of our annotators, and 51 were found toxic by the majority of our annotators; this is an indicator of the well-known subjectivity of toxicity detection. On the 31, 51, and 87 posts, the average kappa score was 65%, 55%, 48%, respectively, indicating that when the raters agree (at least by a majority) about the toxicity of the post, there is also reasonable agreement regarding the toxic spans. Note that the toxic spans are typically short. This leads to class imbalance (most offsets are marked as non-toxic), increases agreement by chance (on the non-toxic offsets), and leads to low kappa scores (kappa adjusts for chance

[2] https://appen.com/

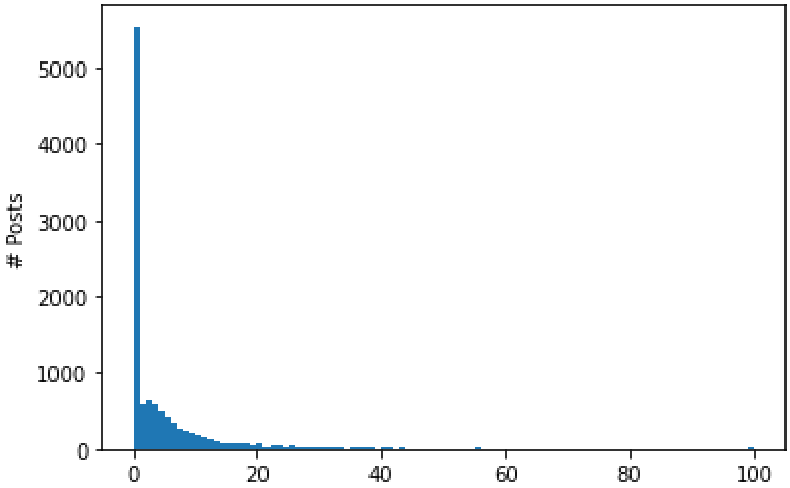


Figure 4.1: Distribution of the percentage of characters of each post that are covered by the ground truth spans.

agreement). Another reason behind this modest (compared to other tasks) inter-annotator agreement is the inherent subjectivity of deciding if a post is toxic or not. Our kappa score is in fact, slightly higher than in previous work on toxicity detection, classifying posts as toxic or not [442, 378], and in that sense, our inter-annotator agreement can be seen as an improvement.

Ground truth To obtain the ground truth of our dataset, we averaged the labels per character of the annotators per post. We used the following process: for each post t, first, we mapped each annotated span of each rater to its character offsets. We then assigned a toxicity score to each character offset of t, computed as the fraction of raters who annotated that character offset as toxic (included it in their toxic spans). We retained only character offsets with toxicity scores higher than 50%; i.e., at least two raters must have included each character offset in their spans. Table 4.1 shows examples.

The dataset ToxicSpans contains the $11,035$ posts we annotated for toxic spans. The unique posts are actually $11,006$ since a few were duplicates and were removed in subsequent experiments. A few other posts were used as quiz questions to candidate annotators' reliability and were also discarded in subsequent experiments.

Exploratory analysis Figure 4.1 shows the distribution of the percentage of character offsets of each post that are included in toxic spans. Figure 4.2 illustrates the distribution of dense toxic spans per post. Figure 4.3 shows the most frequent toxic spans in the dataset (after lower-casing each post) and their frequencies. Figure 4.4 shows the most frequent multi-word toxic spans (again after lower-casing). Figure 4.5 illustrates the distribution of the size (in words) of those posts whose ground truth covers the whole post.

Although we instructed the crowd-raters to click the appropriate tick-box and not highlight any span when the whole post would have to be highlighted, the ground truth of 34 out of the $11,000$ posts covers the entire post.

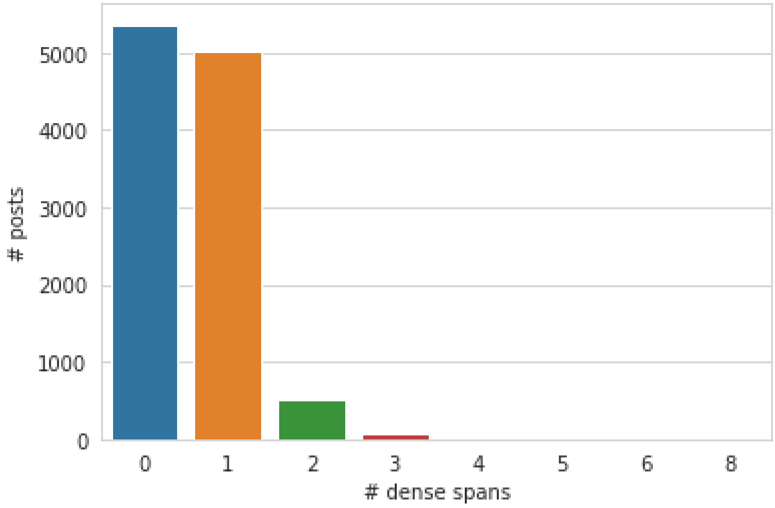


Figure 4.2: Distribution of the number of dense ground truth toxic spans per post in ToxicSpans.

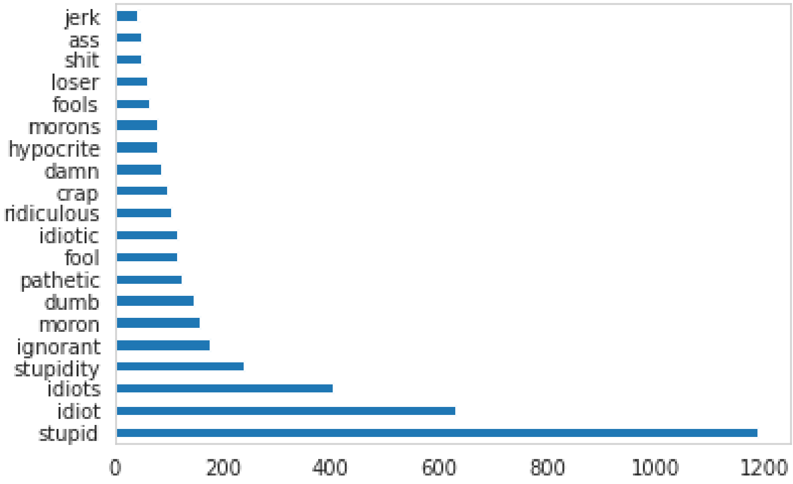


Figure 4.3: Most frequent toxic spans in ToxicSpans.

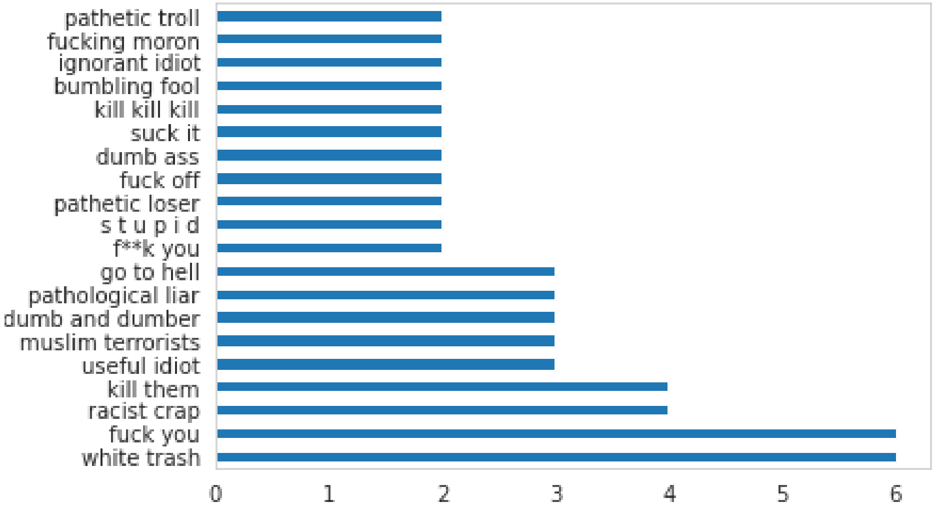


Figure 4.4: Most frequent multi-word toxic spans.

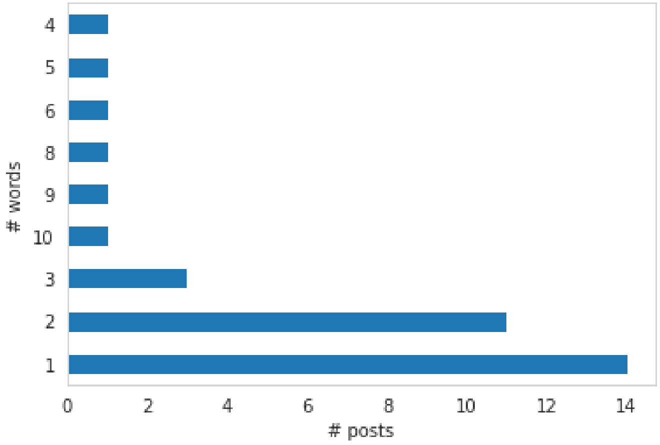


Figure 4.5: Distribution of size (in words) of posts whose ground truth covers the whole post.

However, 14 out of the 34 posts are single-word texts, while the other posts are very short; it seems that in very short posts the raters sometimes did not realize they ended up highlighting the entire post. Furthermore, about $5,000$ of the $11,000$ posts have an empty ground truth set of toxic character offsets (as in the last posts of Table 4.1), even though all the posts of our dataset had been found toxic by the original raters. This is partly due to the fact that we include in the ground truth only character offsets that were included in the toxic spans of the majority of our annotators. It also confirms it is not always possible to attribute (at least not by consensus) the toxicity of a post to particular toxic spans. In almost all posts, the ground truth covers less than half of the post; and in the vast majority, less than 20% of the post. A *dense toxic span* of a post is a maximal sequence of contiguous toxic characters. There exist posts with more than one dense toxic span, but most posts include only one. Table 4.2 provides further statistics.

4.5 Evaluation framework for toxic spans

For the newly introduced toxic spans detection task, we evaluated systems in terms of F_1 score, as in the work of Da San Martino et al. [92]. Given the i^{th} test post, let $\hat{Y}_i$ be the predicted set of character offsets and Y_i the ground truth character offsets. The per-post F_1 score is defined by Equation (4.1).

$$F_1^i = \frac{2 \cdot P^i \cdot R^i}{P^i + R^i} \text{ where } P^i = \frac{|\hat{Y}_i \cap Y_i|}{\hat{Y}_i} \text{ and } R^i = \frac{|\hat{Y}_i \cap Y_i|}{Y_i} \tag{4.1}$$

If Y_i is empty (no gold spans are given), we set $F_1^i = 1$ if $\hat{Y}_i$ is also empty, and $F_1^i = 0$ otherwise. The final test F_1 score is the average of the per-post F_1^i scores over all test posts. We used F_1 as the main evaluation measure in the experiments reported below.

	Mean	Min	Max
Post length	208.14	4	1,000
Dense toxic span length	7.01	3	87
# Dense toxic spans	0.58	0	8

Table 4.2: ToxicSpans statistics. Lengths in characters.

4.6 Methods for toxic spans detection

4.6.1 Simplistic baselines

TRAIN-MATCH, is a simple lookup-based model that classifies as toxic any tokens encountered inside toxic spans of the training data. HATE-MATCH operates similarly, but the lookup is within the hateful/offensive spans of the data of Mathew et al. [334]. A naive baseline, RAND-SEQ, randomly classifies tokens as toxic or not.

4.6.2 Supervised sequence labelling

Toxic spans detection can be seen as sequence labeling (tagging words). As a baseline of this kind, we employed Spacy's CNN (cf. Section 2.6.2), which is pretrained for tagging, parsing, entity recognition [197]. We called this model CNN-SEQ and fine-tuned it on dense toxic spans, treated as 'entities'. We also trained a bidirectional LSTM (BILSTM-SEQ, cf. Section 2.6.2),[3] and fine-tuned BERT [107] (cf. Section 2.5.8) and SPAN-BERT [219] for toxic spans (BERT-SEQ, SPAN-BERT-SEQ). These methods require training data manually annotated with toxic spans, so we refer to them as strongly supervised as opposed to weakly supervised methods.

4.6.3 Weakly supervised learning

We trained binary classifiers to predict the toxicity label of each post, and we employed attention as a rationale extraction mechanism at inference to obtain toxic spans, an approach Pavlopoulos et al. [376] found to work reasonably well in toxicity detection.[4] We experimented with two classifiers: a BILSTM with deep self-attention as in the work of Pavlopoulos et al. [376], but training with a regression objective and probabilistic labels following D'Sa et al. [113] and Wulczyn et al. [543]; and BERT with a dense layer and sigmoid on the `<CLS>` embedding. To detect toxic spans, we used the attention scores of the BILSTM and the attention scores from the heads of BERT's last layer averaged over the heads, respectively. In both cases, we obtained a sequence of binary decisions (toxic, non-toxic) for the *tokens* of the post (inherited by their character offsets) by using a probability threshold (tuned on development data) applied to the attention scores. We refer to these two attention-based rationale extraction methods as

[3] We used the probabilistic ground truth for training and mean squared error as the loss function of BILSTM-SEQ, which yielded best results in preliminary experiments.

[4] Similar attention-based rationale-extraction methods have been used, e.g., by DeYoung et al. [109] and Jain et al. [209], but not in toxicity detection. See also Wiegreffe and Pinter [532], Kobayashi et al. [240], Ferrando and Costa-jussà [127] for a broader discussion of attention as an explainability mechanism.

SQUAD 2.0	TOXICSPANS
Context	Post
Question	Empty string
is_impossible boolean	toxic_spans_is_empty boolean
Answer	Toxic span

Table 4.3: Mapping between the SQUAD 2.0 format and TOXICSPANS examples.

BILSTM+ARE and BERT+ARE, respectively. These methods require training posts annotated only with toxicity labels per *post* (no toxic span annotations).

4.7 Experimental Settings

4.7.1 Sequence labelling

BILSTM-SEQ was implemented in KERAS 2.7.0.[5] We used word embeddings of size 200 and hidden states of size 128; mean squared error (MSE) loss; the Adam optimiser; learning rate 0.001; post padding; maxlen and batch size 128; training for max. 100 epochs. We used early stopping with 5 epoch patience, monitoring the validation loss. The classification threshold was set to 0.5. CNN-SEQ was trained for 30 epochs; we used 0.5 recurrent dropout; progressively increasing batch size from 4 to 32 with step 1. All the other hyper-parameters were set to their default values. BERT-SEQ was implemented using the Huggingface Transformers library.[6] We used the BERT-base-cased model, binary cross entropy loss; the Adam optimiser; learning rate $2 \cdot 10^{-5}$; maxlen 128; batch size 32; training for max. 100 epochs; early stopping with 5 epoch patience, monitoring validation loss. The classification threshold was 0.5.

SPAN-BERT base (cased) was fine-tuned in the same way that Joshi et al. [219] fine-tuned it on SQUAD 2.0 [410] with the format mapping presented in Table 4.3. At training time, we ignored posts with more than one dense toxic span since the SQUAD 2.0 format allows for only one dense answer span in the context. We trained with a learning rate $2 \cdot 10^{-5}$, for 4 epochs with training batches of size 32.

Post-level classifiers with attribution BILSTM+ARE was implemented in KERAS, like BILSTM-SEQ. We used maxlen of 128; post padding; early stopping with patience 5 epoch, monitoring the validation loss; Adam optimizer with 0.001 learning rate; MSE loss. The text classification threshold was 0.5. BERT+ARE was implemented with Huggingface Transformers similarly to BERT-SEQ. We used maxlen of 128; post padding; early stopping with patience 5 epoch, monitoring the validation loss; Adam optimizer with $2 \cdot 10^{-5}$ learning rate; binary cross-entropy loss. The text classification threshold was 0.5. In both models, the attention threshold (above which a token is considered

[5] https://keras.io/

[6] https://huggingface.co/transformers/

		F_1 (%)	P (%)	R (%)
Baselines	RAND	7.3	5.3	25.4
	TRAIN-MATCH	41.0	39.1	48.7
	HATE-MATCH	10.6	7.1	43.7
Strong supervision	BILSTM-SEQ	58.9	59.8	58.9
	CNN-SEQ	59.3	60.7	59.0
	BERT-SEQ	59.7	60.7	60.0
	SPAN-BERT-SEQ	$\mathbf{63.0}$	$\mathbf{63.8}$	$\mathbf{62.8}$
Weak supervision	BILSTM+ARE	57.7	58.4	57.3
	BERT+ARE	49.1	49.4	49.5

Table 4.4: F_1, Precision (P), Recall (R) of sequence labeling (1st zone), attentional (2nd), and look-up methods (3rd) in toxic spans detection. Average scores of a 5-fold Monte Carlo cross-validation are shown. The standard error of the mean is always lower than a percentage point. The ROC AUC scores of BILSTM and BERT (the attention-based rationale extraction methods) in *toxic/non-toxic text classification* were 90.9% and 96.1%, respectively.

toxic) was fine-tuned on the development set of each Monte Carlo cross-validation fold.

Further implementation details can be found in our code repository (cf. the chapter's abstract).

4.7.2 Experimental results for toxic spans

We used a 5-fold Monte Carlo cross-validation (5 random training/development/test splits) on the $11,000$ posts of TOXICSPANS. In each fold, we use 10% of the data for testing, 10% for development, and 80% for training. In ARE-based methods, which rely on an underlying classifier to predict if a post is toxic or not, the classifier is trained on the training part of the fold (which contains only toxic posts, ignoring the toxic span annotations) and a randomly selected equal number of non-toxic posts from Civil Comments that are not included in our dataset. When measuring the (binary) classification performance of the underlying classifier, the classifier is evaluated on a new equally balanced test set of $3,000$ randomly sampled unseen posts from Civil Comments.

Both look-up methods (TRAIN-MATCH, HATE-MATCH) outperform the random baseline (Table 4.4). However, TRAIN-MATCH performs much better, which agrees with our hypothesis that toxicity detection is a broader problem than hate speech detection. Both look-up methods are outperformed by the sequence labeling models (-SEQ), especially SPAN-BERT-SEQ, which is pretrained to predict spans. These results show that the tokens of toxic spans are context-dependent, and their meaning is not captured well by context-unaware look-up lexicons. BERT+ARE performs worse than BILSTM+ARE, despite the fact that the underlying BERT classifier is much better (ROC AUC 96.1%) at separating toxic from non-toxic *posts* than the underlying BILSTM (90.9%). Interestingly, the BILSTM binary toxicity classifier with the attention-based toxic span detection mechanism [376] is close in performance with BILSTM-SEQ, despite the fact that the latter is directly trained on toxic span annotations, whereas the former is trained with binary post-level annotations only (toxic, non-toxic *post*).

You can stick your d**k up anyone's butt. Why have any laws at all?
Of course they do. Stupid people really have to meet everyone else half way if they don't want to be called stupid, starting with not saying stupid things.

Table 4.5: Examples of posts where SPAN-BERT-SEQ incorrectly predicted no spans. Ground truth in red.

Play **stupid** games, win stupid prizes.
I always smile when I've been called **stupid** by a fool.

Table 4.6: Examples of posts where SPAN-BERT-SEQ predicted some, but not all of the gold spans. Ground truth in red. Predictions of SPAN-BERT-SEQ in **bold**.

Error analysis of SPAN-BERT-SEQ We performed an error analysis on our best toxic spans detector (SPAN-BERT-SEQ). We analyzed its predictions on the first fold of the Monte Carlo Cross-Validation, which comprises 10% of the dataset or $1,001$ posts. We identified three main types of errors. The first type of error, which is the most frequent one occurring in 235 out of $1,001$ posts (23.5%), comprises posts for which SPAN-BERT-SEQ failed to find all toxic spans. This type of error can be divided in two sub-types: the first sub-type comprises posts for which SPAN-BERT-SEQ predicted no spans at all (Table 4.5), while the second sub-type comprises posts for which SPAN-BERT-SEQ predicted some, but not all of the gold spans (Table 4.6). The first sub-type occurs more often, with 217 out of the 235 total occurrences of the first error type, while the second sub-type occurs only a few times (18 out of 235). The second type of error, which is the second most frequent one, occurred in 173 out of the $1,001$ posts (17.3%). It occurs when the ground truth of a post is empty, but SPAN-BERT-SEQ predicts at least one toxic span (Table 4.7). The last type of error occurs rarely (only 10 out of $1,001$ posts) when the ground truth of a post is not empty, and SPAN-BERT-SEQ predicts more (or larger) toxic spans than it should (Table 4.8).

Improving attribution-based detection Several large datasets with *post-level* toxicity annotations are publicly available [379]. Therefore, attribution-based toxic span detectors, such as BILSTM+ARE, can, in principle, perform even better if the underlying binary classifier is trained on a larger existing dataset. To investigate this, we increased

Not sure if "people are **dumb**" is the best descriptor, but you are correct that we tend to seek out and grasp at anything that supports our beliefs and hopes. Hence the proliferation of "fake news", which feeds those wants.
They can shuffle the cabinet seven ways from Sunday and it's still a cabal of **losers**.
This outlet should hire some editors. Nobody I've crossed paths with would green light this **crap**.
Actually, Seaton is a wealthy man and can do without his day job quite easily. If he would just get rid of that friggin' **stupid** cap....
In other word, blah, blah, blah, blah. It's **bullshit**. Deal with it. No proof=doesn't exist.
Or maybe we should place a tax on **stupid** ideas like yours

Table 4.7: Examples of posts where the ground truth was empty, but SPAN-BERT-SEQ incorrectly predicted at least one span. Predictions of SPAN-BERT-SEQ in **bold**.

People don’t normally take it to heart when an **idiot calls someone stupid**.
$10B a GW avg compared to $2.5B a GW for a 2nd Candu nuke at LePreau. **Stupid is as stupid** does I guess.
All **useless piles of crap**.
oh no, this isn’t even in the top 10 **moronic statements by this babbling fool**.

Table 4.8: Examples of posts where the ground truth was not empty, and SPAN-BERT-SEQ incorrectly predicted more (or larger) toxic spans. Ground truth in red. Predictions of SPAN-BERT-SEQ in **bold**.

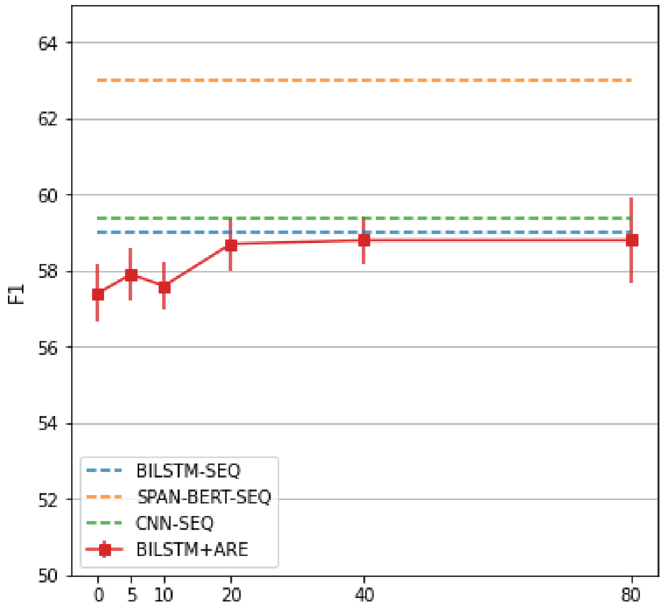


Figure 4.6: Improvement in the F1 of BILSTM+ARE when increasing the training set of its underlying BILSTM with posts tagged at the post-level (toxic/non-toxic, no toxic spans). Standard error of mean shown as error bars.

the training set of the underlying BILSTM classifier of BILSTM+ARE. We added to the training set of each cross-validation fold $80,000$ further toxic and non-toxic posts (still equally balanced, without toxic spans) from the dataset of Borkan et al. [48], excluding posts used in TOXICSPANS. The ROC AUC score of the underlying BILSTM (in the task of separating toxic from non-toxic posts) improved from 90.9% to 94.2%, and the F_1 score of BILSTM+ARE (in toxic spans detection) improved from 57.7% to 58.8%, almost reaching the performance of BILSTM-SEQ.

Figure 4.6 shows the improvement in the F_1 score of BILSTM+ARE when increasing the training set of the underlying BILSTM with 5000, 10000, 20000, 40000, 80000 more posts (always balanced toxic/non-toxic) with post-level annotations only (no toxic span annotations). The dashed lines represent the sequence labeling methods, which cannot benefit directly from training data without toxic span annotations. Similarly, Figure 4.7 shows the corresponding improvement in the ROC AUC score of the underlying BILSTM in the toxic/non-toxic text classification task.

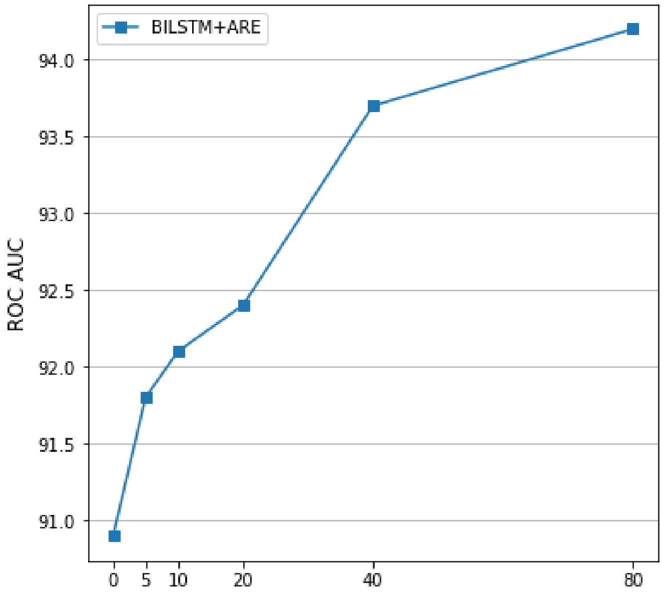


Figure 4.7: Improvement in the ROC AUC of BILSTM+ARE in the toxic spans detection task, when increasing the training set of its underlying BILSTM with posts tagged at the post-level (no toxic spans).

4.8 Toxic spans in toxic-to-civil transfer

As shown in Section 4.7.2, a toxic span detection method can be used to highlight toxic parts of a post, to assist, for instance, human moderators. The new ToxicSpans dataset and toxic span detection methods, however, can assist in more ways. This section describes how we combined the new dataset and the best-performing toxic span detector (SPAN-BERT-SEQ) to show how they can be useful in *toxic-to-civil text transfer* (cf. Chapter 3). In the context of detoxifying comments to nudge users toward healthier conversations online, this task aims at suggesting civil rephrasings of toxic posts. More specifically, we studied the following research question:

RQ: *Can* ToxicSpans *data and toxic span detectors be used to assess the mitigation of* explicit toxicity *in toxic-to-civil transfer?*

To answer this question, we proceeded in two ways:

(i) Evaluating the transfer of toxic spans in *system*-detoxified posts

(ii) Studying any remaining toxic spans in *human*-detoxified posts.

4.8.1 System-detoxified posts

We first compared the performance of two toxic-to-civil transfer models, CAE-T5 and SED-T5, both based on the T5 encoder-decoder transformer architecture [406]; they both fine-tune the weights of the same pretrained model, namely T5-large. CAE-T5 [266] is the self-supervised Conditional Auto-Encoder presented in Chapter 3. It had

	Parallel (P)	**Non-Parallel (NP)**	
	Toxic-to-Civil pairs	Toxic	Civil
Train	2,222	90,293	5,653,785
Dev	278	4,825	308,130
Test	278	4,878	305,267
Average length	19.8[a]	19.4	21.9

[a] Toxic posts

Table 4.9: Statistics for the parallel (P) and non-parallel (NP) datasets, used to train the SED-T5 and CAE-T5 toxic-to-civil models, respectively. Average lengths are reported by counting SentencePiece [251] tokens.

Civil to toxic (P)	
Input	stop being ignorant and lazy and try reading a bit about it.
CAE-T5	try reading and **be a little more informed** about it before you try to make a comment.
SED-T5	stop being *lazy and lazy* and try reading a bit about it.
Input	this is absolutely the most idiotic post i have ever read on all levels.
CAE-T5	this is absolutely the most **important thing** i have read on this thread over the years.
SED-T5	this is ~~not~~ the most **useless** *post* post I have ever read on all levels
Input	trump may be a moron, but clinton is a moron as well.
CAE-T5	trump may be a *clinton supporter*, but clinton is a *trump supporter* as well.
SED-T5	trump may be a *badon*, but clinton is a *badon* as well.
Input	shoot me in the head if you didn't vote for trump.
CAE-T5	~~you're right~~ if you didn't vote for trump. ~~i'm not sure i'd vote~~
SED-T5	*shoot me the head* if you didn't vote for Trump
Input	50% of teachers don't have any f*cks to give.
CAE-T5	50% of teachers don't have ~~a phd in anything.~~
SED-T5	50% of teachers don't have any *tofls* to give.

Table 4.10: Examples of automatically transferred sentences from the NP test set (toxic to civil), **valid rewriting**, and highlighted flaws failure in *attribute transfer or fluency*, ~~supererogation~~, ~~position reversal~~, and ~~hallucination~~.

been fine-tuned on a large non-parallel (NP) dataset based on pre-processed posts from the Civil Comments (CC) dataset, the dataset (with post-level annotations) that TOXICSPANS was also based on. SED-T5 is a Supervised Encoder-Decoder; we fine-tuned it on a smaller parallel (P) dataset created by Dementieva et al. [105], consisting of pairs of comments: a toxic comment and a detoxified paraphrase written by a crowdworker.

Table 4.9 summarizes statistics of the two datasets (P, NP) and highlights a trade-off between the level of supervision and number of samples: there is a $1:40$ ratio between toxic comments in P (direct supervision, parallel data) and NP (indirect supervision, no parallel data). Table 4.11 shows our experimental results. We report accuracy (ACC), Perplexity (PPL), similarity (SIM), and the geometric mean (GM) of ACC, $1/$PPL, SIM. The systems computing these metrics are the same as in Chapter 3. As a reminder, accuracy measures the rate of successful transfers from toxic to civil. It computes the fraction of posts whose civil version is classified as non-toxic by a BERT toxicity classifier. Perplexity is used here as a measure of fluency. Similarity measures content preservation between the original toxic text and its system-rephrased civil version (self-SIM) or the gold (human) civil rephrasing (ref-SIM, only for P).

As can be seen in Table 4.11, CAE-T5 has better aggregated results (higher GM) than SED-T5 in all three datasets, which are due to lower Perplexity and (in NP and TOXICSPANS) higher accuracy. However, SED-T5 learned to preserve content better (higher SIM in all three datasets) because of the parallel data (P, with gold rephrases) it was trained on. By contrast, CAE-T5 was trained without parallel data (NP) using a cycle-consistency loss, which leads to more frequent hallucinations of content that was not present in the original post [266]. These hallucinations may also help CAE-T5 obtain better Perplexity scores by generating fluent civil 'rephrases' that do not preserve, however, the original semantics. Examples shown in Table 4.10 illustrate the tendency of CAE-T5 to remove toxicity with hallucination and position reversal compared to SED-T5's rephrasings, often more faithful to the input comment (rare cases of hallucination), but at the cost of fluency and detoxification accuracy. Also, although the general trends are similar in all three datasets (SED-T5 preserves content better, CAE-T5 is better in Perplexity and GM), there are several differences too across the three datasets. For example, CAE-T5 is much better than SED-T5 in accuracy (posts detoxified) on NP and TOXICSPANS, but both systems have the same accuracy on P; and the scores of the systems vary a lot across the three datasets.

These considerations motivated us to seek ways to analyze the behavior of toxic-to-civil transfer models further. TOXICSPANS and toxic span detectors are an opportunity to move towards this direction by studying how well transfer models cope with *explicit toxicity*, i.e., spans that can be explicitly pointed to as sources of toxicity. We leave for future work the flip side of this study, i.e., studying cases where transfer models rephrase spans not explicitly marked (by toxic span detectors or human annotators) as explicitly toxic.

4.8.2 Explicit Toxicity Removal Accuracy

Recall that the accuracy (ACC) scores of Table 4.11 measure the percentage of toxic posts that the transfer models (CAE-T5, SED-T5) rephrased to forms that a (BERT-based) toxicity classifier considered non-toxic. One could question, however, if it is possible (even for humans) to produce a civil rephrase of a toxic post when it is impossible to point to particular spans of the post that cause its toxicity (as in the last posts of Table 4.1). Detoxifying posts of this kind may constitute a mission impossible for most models (possibly even for humans); the only way to produce a non-toxic 'rephrase' may be to change the original post beyond recognition, which may be rewarding systems like CAE-T5 that often hallucinate in their rephrases, as already discussed.

Hence, it makes sense to focus on posts that contain explicit toxic spans, marked by human annotators (for TOXICSPANS) or our best toxic span detector (SPAN-BERT-SEQ). Using these toxic spans, we defined three additional variants of accuracy: ACC2 is the same as ACC, but ignores posts that do not contain at least one toxic span; ACC3 also considers (in its denominator) only posts that contained at least one toxic span, but computes the fraction of these posts that had all of their toxic spans rephrased (even partly) by the transfer model; ACC4 is a stricter version

Evaluation Dataset	Metric	CAE-T5	SED-T5
Non-Parallel (NP)	ACC ↑	**75.0 %**	52.2 %
	ACC2 ↑	**83.4 %**	67.3 %
	PPL ↓	**5.2**	11.8
	self-SIM ↑	70.0 %	**87.9 %**
	GM (self) ↑	**0.466**	0.338
	ACC3 ↑	**86.7 %**	64.1 %
	ACC4 ↑	**83.2 %**	59.5 %
Parallel (P)	ACC ↑	94.3 %	94.3 %
	ACC2 ↑	**94.7 %**	94.3 %
	PPL ↓	**9.1**	38.3
	ref-SIM ↑	27.6 %	**65.3 %**
	self-SIM ↑	32.6 %	**65.6 %**
	GM (ref) ↑	**0.306**	0.252
	GM (self) ↑	**0.323**	0.252
	ACC3 ↑	**98.8 %**	94.3 %
	ACC4 ↑	**94.7 %**	91.9 %
TOXICSPANS	ACC ↑	**92.9 %**	65.6 %
	ACC2 ↑	**92.5 %**	63.7 %
	PPL ↓	**7.2**	24.9
	self-SIM ↑	34.5 %	**82.1 %**
	GM (self) ↑	**0.355**	0.279
	ACC3 ↑	**96.9 %**	62.0 %
	ACC4 ↑	**92.0 %**	54.7 %

Table 4.11: Automatic evaluation scores of CAE-T5 (trained on NP's training subset) and SED-T5 (trained on P's training subset), when the test sets are from NP, P, and TOXICSPANS. ACC2, ACC3, ACC4 also consider toxic spans (Section 4.8.2).

of ACC3 that requires the posts to also be judged non-toxic by the (BERT-based) toxicity classifier.

Table 4.11 shows that restricting ACC to consider only posts with at least one toxic post (ACC2) substantially improved the performance of both models on the NP dataset, indicating that it contains many 'mission impossible' instances (posts with no toxic spans) that the original ACC considers. By contrast, switching from ACC to ACC2 led to mostly negligible changes on the P and TOXICSPANS datasets, which is in accordance with the fact that they contain fewer posts with no toxic spans (11.5% and 48.7%, respectively, compared to 67.4% for NP). Another interesting observation is that ACC4 was always substantially lower than ACC3 (for both systems, on all three datasets), indicating that the models often successfully detected toxic spans and try to rephrase them, but the rephrases were still toxic, at least according to the toxicity classifier.

4.8.3 Human-detoxified posts

In this experiment, we wished to study the extent to which *humans* rephrase known toxic spans when asked to produce civil rephrases of toxic posts. We used the P dataset, the only one of the three considered that contains human rephrases.[7] Since P does not contain gold toxic spans, we again employed SPAN-BERT-SEQ to add toxic

[7] We used all the P data since no training was involved.

spans to the source posts and retained only the $1,354$ (out of $2,778$ in total) source-target pairs of posts with at least one toxic span in their source post.[8] In all but 6 of the $1,354$ posts, the humans have rephrased (in the gold target post they provided) all the toxic spans of the source post. The 6 posts were mainly cases where the human changed the context to mitigate toxicity, while retaining the original toxic span. For example, ‘‘he’s not that stupid’’ became ‘‘he’s not stupid’’ (original toxic span shown in bold); in this case removing the ‘‘that’’ from the context arguably makes the post less offensive. Overall, we conclude that humans did rephrase almost all cases of explicit toxicity in the toxic posts they were given.

We also applied SPAN-BERT-SEQ to the gold target (rephrased) posts that the humans provided to check if any explicit toxicity remained or was introduced by the rephrases. This flagged 93 gold target posts as comprising at least one toxic span. A manual inspection of the 93 posts revealed that they fall into two main categories. The first category comprises cases where a toxic span of the source post was rephrased, but the rephrase might not be considered totally civil; e.g., ‘‘how freaking narcissistic do you have to be?’’ became ‘‘how narcissistic do you have to be?’’, where SPAN-BERT-SEQ marked the ‘‘narcissistic’’ of the rephrase as a toxic span. The second category comprises cases where SPAN-BERT-SEQ produced false positives; e.g., the source post ‘‘most of the information is total garbage’’ became ‘‘most of the information is totally useless’’, but SPAN-BERT-SEQ marked (arguably incorrectly) ‘‘useless’’ as a toxic span.

4.8.4 Toxicity scores of posts with and without explicit toxicity

We also applied the BERT-based text toxicity classifier of Chapter 3 to the $2,778$ posts of the P dataset, dividing them into two sets: posts that comprised at least one toxic span detected by SPAN-BERT-SEQ ($1,354$ posts with explicit toxicity) and the rest (implicit toxicity). The BERT-based toxicity classifier considered more toxic (higher average toxicity score) the $1,354$ posts of the first set compared to the second one, i.e., it was more confident that the posts of the first set (explicit toxicity) were toxic, as one might expect. By resampling $1,000$ subsets (of 50 posts each) from the two sets, we confirmed that this is a statistically significant difference ($P = 0.001$). The difference in the average predicted toxicity score between the two sets is 14% (from 0.94 down to 0.80).

4.9 Discussion

The posts we annotated for toxic spans were extracted from an already heavily studied public domain benchmark dataset (Civil Comments) that has been examined by thousands of teams in a Kaggle competition,[9] and that has been cited in over 50 academic publications. The Civil Comments dataset was filtered to remove any potential personally identifiable information before it was released. Our annotation cost was $\$21,089$ for $59,486$ judgements,

[8] The most frequent spans were ‘‘sh*t’’, ‘‘st*p*d’’, ‘‘f*ck’’.

[9] shorturl.at/hqEJ3

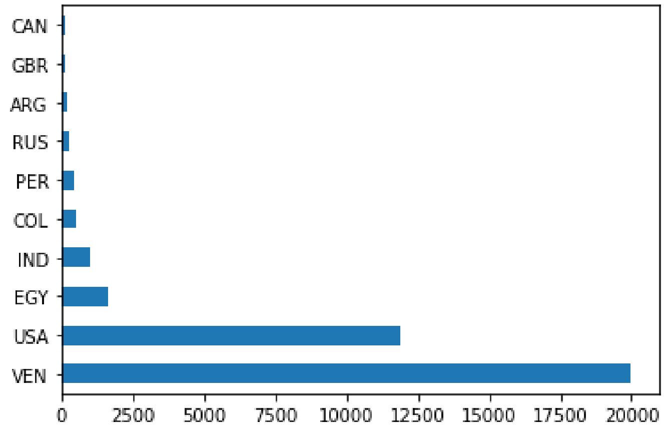


Figure 4.8: Frequency of annotations based on the country of origin of the crowd-annotators.

paying $\$0.30$ per item. All raters were warned for the explicit content of the job, and only high accuracy raters were selected $(70+\%)$ based on performance on quiz questions. The most common countries of origin of our crowd-annotators were Venezuela and USA (cf. Figure 4.8). In the contributor satisfaction survey, 51 participants gave an overall task rating of $3.6/5.0$, with pay and test question fairness rated slightly higher than ease of job and clarity of instructions.

We note that it is more difficult and costly (approximately 3 times more) to manually annotate toxic spans, instead of just labeling entire posts as toxic or not. This is why we also explored adding rationale extraction components on top of toxicity classifiers trained on existing much larger datasets. We showed that BILSTM+ARE has the potential to reach the performance of BILSTM-SEQ, which is important for future work aiming to build toxic span detectors without any toxic span annotations in the training data. This may be particularly useful in low-resourced languages with limited resources for text toxicity [561].

Having two separate systems, one for toxicity detection and one for toxic spans identification, is more easily compatible with existing deployed toxicity detectors. One can simply add a component for toxic spans at the end of a pipeline for toxicity detection, and the new component would be invoked only when toxicity would be detected, leaving the rest of the existing pipeline unchanged. Since the vast majority of posts in real-world applications are non-toxic [48], this pipeline approach would only increase the computational load for the relatively few posts classified as toxic. Using only toxic posts in this study was also a way to simplify this first approach to toxic spans detection, assuming an oracle system achieved the first step (deciding which posts are toxic). However, we note that future work could study adding non-toxic posts to our dataset and require systems to first detect toxic posts, then extract toxic spans for toxic posts.

A direct comparison (in terms of size) of TOXICSPANS with other existing toxicity datasets is only possible if one focuses on the toxic class, typically the minority one, since our dataset contains only toxic posts. By adding non-toxic posts, much larger versions of our dataset can be compiled, of sizes similar to those of existing previous datasets (that provide post-level annotations only). Hence, our ToxicSpans dataset is accessible with the following versions: First, only toxic posts are included ($11,006$ posts), which is the version we discuss in this work. Second, the previous version will be augmented with the same number of randomly selected non-toxic Civil Comments posts. Third, a version similar to the previous one, but where the ratio of toxic to non-toxic posts will be $1:40$ to be closer to that of real-world datasets ($325,499$ posts).

As shown in Section 4.8, the ToxicSpans dataset and toxic span detectors can also help study and evaluate explicit toxicity removal when rephrasing toxic posts to be civil. In this case, toxic spans can be used to get a better understanding of how toxic-to-civil models operate by showing the toxic spans and their context, along with their rephrases.

For more details on the shared task we organized at SemEval-2021, please see the competition description in Appendix A.

4.10 Intended use and misuse potential

The toxic span detection systems we considered are trained (the sequence-labeling ones) and tested (all systems) on posts with binary ground-truth character offset labels (toxic or not), reflecting the majority opinion of the annotators (Section 4.4). This runs the risk of ignoring the opinions of minorities, who may also be minorities among crowd-annotators. To address this issue, we also released the toxic spans of all the annotators and the pseudonymous rater identities, not just the spans that reflect the majority opinion, to allow different label binarisation strategies and further studies.

Toxic span detection systems are intended to assist the decision-making of moderators, not to replace moderators. When they operate correctly, systems of this kind are expected to ease decision-making (reject/accept a post). Incorrect results could be of two types; toxic spans that were not highlighted and non-toxic spans that were highlighted. Mistakes of both types, especially the first one, may mislead a moderator working under time pressure.

As with other content filtering systems (e.g., spam filters, phishing detectors), toxic span detectors may trigger an adversarial reaction of malicious users, who may study which types of toxic expressions evade the detectors (esp. publicly available ones) and may gradually start using more implicit toxic language (e.g., irony, false claims), which may be more difficult to detect. However, this is a danger that concerns any toxicity detection system, including systems that classify user content at the post level (without detecting toxic spans).

4.11 Subsequent related works

Following our work on English toxic spans, Ravikiran et al. [415] published a similar approach adapted to Code-Mixed Tamil-English comments. We hope our project initiated a larger trend in multilingual toxic span detection as non-English-only platforms may benefit from automated online moderation at the span level as well.

Chapter 4 conclusion

We studied toxicity detection, which aims to identify the spans of a user post that make it toxic. Our work is the first of this kind in general toxicity detection. We constructed and released a dataset for the new task, along with baselines and models. Fine-tuning the SPAN-BERT sequence labelling model of Joshi et al. [219], yielded the best results. A post-level BILSTM toxicity classifier that was combined with an attention-based attribution method, not trained on annotations at the span level, performed well for the task. By leveraging the dataset of posts annotated as toxic or non-toxic (without spans), we showed that this method can reach the performance of a BILSTM sequence labeling approach that was trained on the more costly toxic spans annotations. This result is particularly interesting for future work aiming to perform toxic spans detection by using only datasets with whole-post toxicity annotations. In a final experiment, we examined toxic-to-civil transfer, showing how toxic spans can help shed more light on this task, too, by helping assess how well systems and humans address explicit toxicity. In future work, we plan to study toxic span detection in multiple languages and in context-dependent toxic posts.

Chapter 5

Semantic Encoding of Review Sentences for Memory-Based Recommenders

This chapter presents research conducted with Thomas Bonald, Lucas Dixon, and Raghuram Vadapalli. It led to an article submitted to the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP 2022) and not reviewed yet.

Chapter 5 abstract

This chapter explores a novel use of review text to represent user-preferences for rating prediction. The approach leveraged textual semantic similarity models to represent a user's preferences as a graph of textual snippets, where the edges were defined by semantic similarity. This textual, memory-based approach to rating prediction offers the promise of improved explanations for recommendations. The method was evaluated quantitatively, highlighting that leveraging text in this way can out-perform both memory-based and model-based collaborative filtering baselines.

5.1 Introduction

With the democratization of online shopping and content delivery platforms, internet users are accustomed to writing textual reviews of products and services. While recommender systems initially relied only on quantitative preference scores, newer datasets with textual reviews, combined with advances in Natural Language Understanding, suggest that both the performance of recommenders and their explainability might benefit from leveraging textual representations.

While *item recommendation* systems aim to predict the next interaction corresponding to the users' preferences, this work focused on *rating prediction* systems that try to predict the rating a user will give an item. For example, when a user is trying to evaluate a given item, e.g., they heard about a movie and are looking it up before choosing whether to watch it or not. In this setting, search engines and film streaming sites like Netflix show information to help the user make a decision, such as the average rating, public reviews, and sometimes personalized ratings and explanations for why the user may enjoy it (so-called endorsements). This work focuses on providing high-quality personalized rating prediction leveraging review text in a way that supports explanations based on snippets of review text from other users.

Statistical and ML models are often used to compute personalized predictions based on the past interactions (e.g., ratings and reviews) r_{ui} of a set of users $u \in U$ on a set of items $i \in I$. The archetypal baseline for recommender systems is Collaborative Filtering (CF) [421]. CF-based approaches infer preference scores $\hat{r}_{ui}$ from the similarity between users who expressed common tastes. The computation of these commonalities varies from the data the system has access to, though the majority of traditional models rely on past ground-truth ratings. Even if they show acceptable quantitative results, rating-based recommender systems often suffer from a lack of clear and precise explanation for why the recommendation was made.

Srifi et al. [481] recently reviewed attempts to integrate review text in recommender systems. They examined existing methods, classified by Chen et al. [72], that extract information from reviews in order to enhance recommender systems. While many of these studies inferred high-level features such as sentiments or topics, only a few directly built semantic-aware recommender systems from the unstructured text found in reviews.

Memory-based and model-based systems are the main two families of methods seen in CF [3, 71]. Memory-based CF predicts user ratings by directly aggregating ratings from other users with similar preferences. Model-based CF uses interaction data to train a model—typically a high dimensional vector—that can be used to generate ratings for future items. Even when memory-based CF shows worse absolute quantitative evaluation on standard metrics, it benefits from the simplicity of implementation and ease of creating explanations.

To the best of our knowledge, the work from Terzi et al. [501] (cf. Section 5.2) is the only effort to leverage text in memory-based CF by using word-based similarity measures between reviews to compute user similarities. However, recent progress in NLU enables far richer and more effective semantic representations of text [515, 108, 405]. In this chapter, we used modern large Pretrained Language Models [552] to embed sentences of reviews into a so-called semantic vector space (Section 5.5). We then employed a user k-NN model based on user similarity (Section 5.6). The key insight is that user similarity can be defined in terms of a graph based on semantic similarity from text in user reviews. Specifically, we showed how to build sentence-based k-Nearast Neighbors (k-NN) graphs (Section 5.7) to compute user similarity. We denote our model Text-KNN[1] (Figure 5.2).

We evaluated our model in the context of two large datasets containing both numerical and text reviews; the Amazon Review dataset [175] and the Yelp dataset[222]. Offline metrics are often used to evaluate recommender

[1] Our code is available at `https://bit.ly/3N1L6Qa`

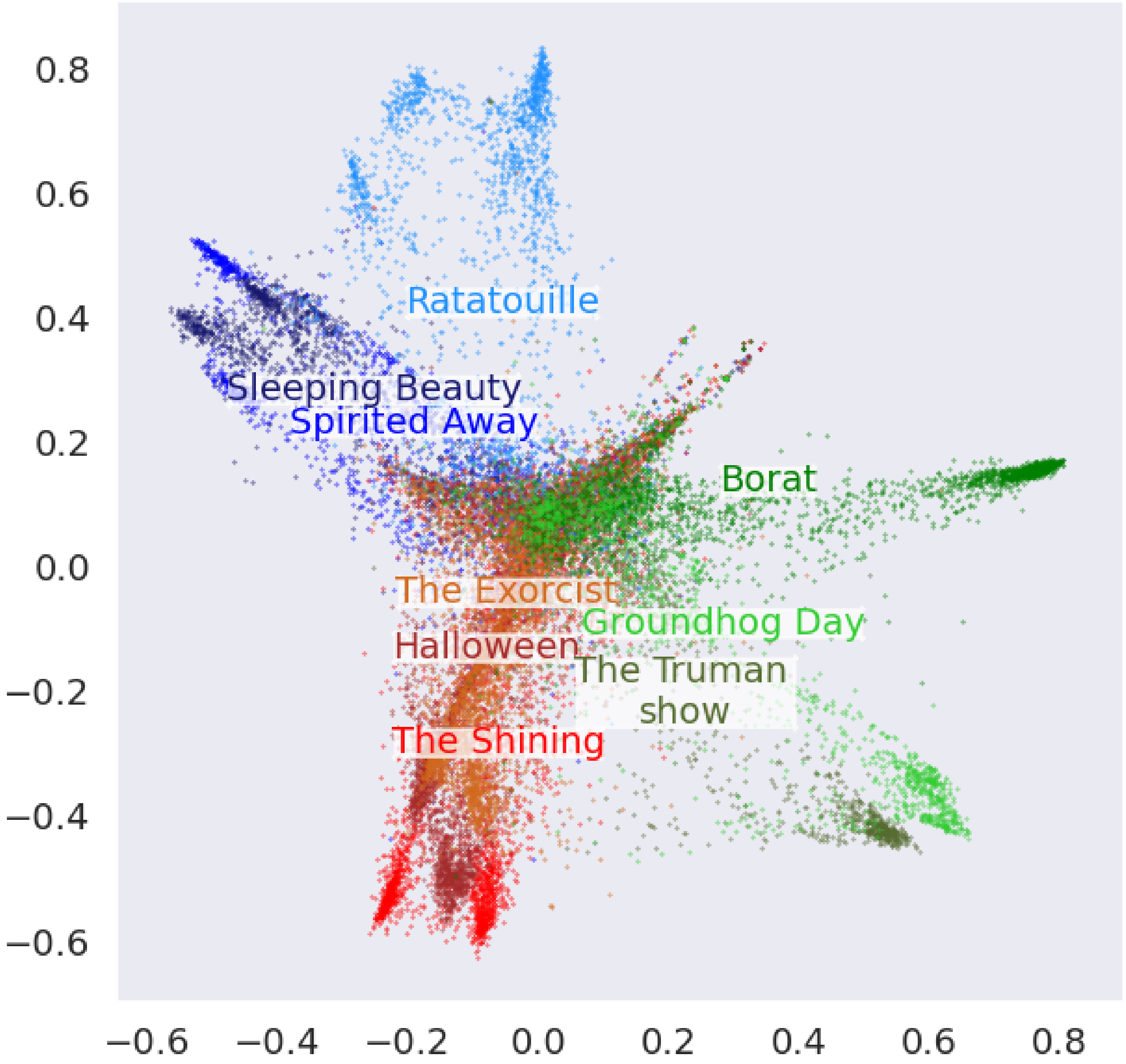


Figure 5.1: 2D t-SNE projection of sentence embeddings. Red, green, and blue colors represent sentences from horror, comedy, and animation movies, respectively. Movie titles are displayed at the barycenter of their review sentences (with adjustment for readability purposes). Figure best viewed in color.

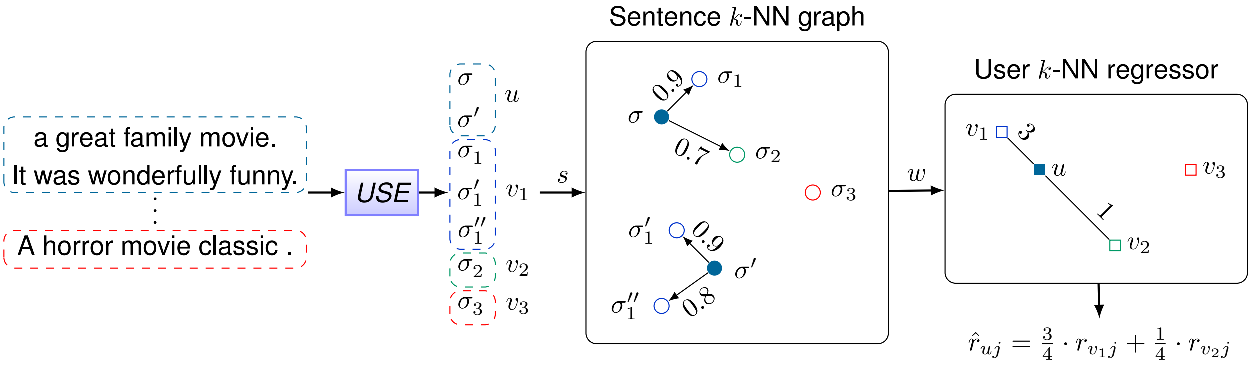


Figure 5.2: Text-KNN pipeline. $\sigma_n, \sigma_n', \sigma_n'', \ldots$ represent the embeddings of sentences written by user v_n. k-NN graphs are built from the sentence similarity s, and are used for computing the weights w of the user k-NN regressor. Figure best viewed in color.

systems. Although top-N ranking metrics such as discounted cumulative gain [210] and hit rate evaluate item recommendation, rating prediction has traditionally been evaluated with error-based predictive accuracy metrics like the Root-Mean-Square Error (RMSE). However, it has been observed that RMSE is not effective at distinguishing the quality of algorithms—high performance can even be achieved by naive baselines [91]. Moreover, RMSE unfairly represents users with a smaller spread of scores (as error values will be smaller). In Sections 5.8, we explored arguments in favor of rating-prediction evaluation based on rankings, leading us to also report a metric inspired by the Fraction of Concordant Pairs [243].

5.2 Related work

User k-NN is the main approach studied in memory-based CF. The cornerstone of k-NN is the definition of a similarity measure between two users [363]. Several similarities can be computed from the vectors of the ratings of co-reviewed items, among which mean squared difference (MSD, 463, 383), cosine similarity [301], Pearson correlation coefficient [180] and Jaccard coefficient [246].

Model-based CF rely on intrinsic models trained with various machine learning techniques [485, 546, 573] to directly predict user preferences. In particular, state-of-the-art latent factor models [244] such as probabilistic matrix factorization [437], Singular Value Decomposition (SVD) [138] and SVD++ [242] achieve the most precise rating predictions.

Research has explored different ways of extending the rating-only-based systems with additional data ources to represent users and items. Non-text-review information include social information [484], tags [167, 17] and descriptions [7]. However, text found in reviews is the most promising type of information to augment rating-based datasets and models [580, 72, 182]. Many works are based on that approach. Some strategies consist in extracting relevant information from the text review before integrating it into a recommender system. This information can narrow-down to specific content, such as topics [335, 20, 74] or sentiments [393, 577, 100].

Further strategies rely directly on the words found in reviews without trying to focus on specific informative elements (like topics, overall sentiment, or aspect-based opinion). Considering integrating the information at the word level provides potentially richer compositionality power. The literature contains examples of approaches building latent representations of sentences with neural networks before integrating these into matrix factorization CF [235, 587, 69, 305].

Even if a few works combine text information with review information in memory-based CF [234, 283, 352], Terzi et al. [501] were the first to design user-based k-NN similarities directly from the words found in text reviews and therefore their work is the most relevant to us. They introduced six word-based similarity measures ranging from simple word overlap, to depth of two words [384] in external lexical graphs[2], and information content [418, 213, 297]. Even though this is the closest approach to ours, we differed in that sentence similarity relies on continuous (sentence embedding) rather than discrete (co-occurrence of terms) semantic representations.

Evaluating recommender systems is a critical step in their development and remains an open research topic. Many offline (automatic) evaluation metrics have been introduced as proxies to approximate online (human) evaluation. It has not been shown yet that a metric is universally better than others, though RMSE seems to be the dominant approach for its simplicity of implementation and understanding. Cremonesi et al. [91] justified why different metrics perform differently on different tasks (rating prediction versus item recommendation) and Herlocker et al. [180] summarized existing metrics targeting specific criteria of recommender systems. Chandar et al. [66] argued for taking into account user behavior models when designing and selecting metrics. The Fraction of Concordant Pairs was first introduced by Koren and Sill [243] as an offline metric relevant for evaluating if predicted ratings were ranked the same way as the ground truth. It was re-used by Wu et al. [542], Beniwal et al. [30].

Besides recommender systems, the field of information retrieval (IR) [58] is relevant to our study. There has been a large trend of work in IR, including snippet extraction [291] and endorsement [286].

5.3 The more explicit the feedback, the richer the information

Our preferences are only partially reflected by numerical scores. The lowest-level information can be found in implicit feedback, where data only indicates whether a user has interacted with an item (e.g., movie watched or not, product bought or not). Numerical explicit feedback provides richer information: 5-points scale ratings enable systems to ignore the non-rated items, focusing on a linear scale of preferences. Explicit feedback is not limited to quantitative assessment, and the Web facilitated immediate exchanges of opinion on various topics in online fora

[2] WordNet was employed [345].

ITEM j LIKED BY SOME USER u	OTHER ITEM REVIEWED BY u, ALONG WITH A SENTENCE u WROTE
Sleeping beauty	**Ratatouille** – “ Ratatouille ” is a thoroughly entertaining movie that is perfect family fare for the summer .
The Shining	**Haloween** – This is a must movie for all true horror fans .
Groundhog day	**The Truman Show** – “ The Truman Show ” will make you laugh , and keep you on the edge of your seat , wondering if Truman will ever get out.

Table 5.1: Examples of sentences (right column) written by a user u on some item (in bold) they reviewed and useful to infer that u likes another item j (left column). An item is “liked” when the rating is above $4/5$.

first. Then, social networks[3], e-commerce[4] and streaming service[5] platforms democratized it and nowadays users and customers get the opportunity to review services and products they had consumed. Two users u and v may have rated equally the same item $r_{ui} = r_{vi}$ but for totally different reasons. For example, one can appreciate the *Matrix* franchise for its spectacular action scenes, groundbreaking special effects, dystopian sci-fi plot, cyberpunk characters, cult dialogues, actors, the philosophical/spiritual themes addressed, the references to films, literature, myths and religion, etc. A system trained on rating-only data, precise as it is, will never be able to capture those reasons; hence it will make biased predictions.

Contrary to tags or movie plot descriptions, text reviews are totally free-form and may contain many levels of preference information (cf. examples shown in Table 5.1). Text reviews include sentences useful to represent users' tastes or movie features. Some may even contain direct explicit recommendations from a human, which is gold information from the point of view of the task we address. This motivated integrating such information to design or explain recommender systems and assess to what extent they can benefit from it.

5.4 Datasets

We experimented with the 2014 Amazon Product Review dataset [175], made of reviews from users on items with 5-points scale ratings and text reviews. It has been used in related work [525, 524, 589] for its important size. Specifically, we focused on the movie subset because common knowledge makes it easier to interpret movie reviews than reviews of other items. Following He et al. [177], López et al. [316] we restricted our study on the k-core subset, where $k = 20$. Additionally, we trained and evaluated our method on the k-core subset of the Yelp review dataset [222], made of reviews of businesses (like restaurants). Table 5.2 shows statistics for the datasets.

The datasets were split into training, validation, and test subsets. Various splitting strategies exist [339], showing benefits and limitations depending on the nature of the feedback (explicit or implicit), the task (rating prediction or item recommendation), and the model used. Following previous works [501, 316], we preserved time ordering [460] by employing the *Leave One Last Item* strategy. The test and validation sets were respectively made of all users'

[3] https://www.facebook.com/, https://twitter.com/

[4] https://www.amazon.com/, https://www.ebay.com/

[5] https://www.netflix.com/

Dataset	Amazon 20-core	Yelp 20-core
Users	$3{,}728$	$38{,}595$
Items	$3{,}911$	$27{,}823$
Train reviews	$205{,}158$	$1{,}929{,}332$
Validation reviews	$3{,}728$	$38{,}595$
Test reviews	$3{,}728$	$38{,}595$
Average sentence # per review	14.1	9.8
Average token # per sentence	46.2	30.5

Table 5.2: Statistics for the 2014 Amazon Movie Review dataset and the Yelp Review dataset. Following Yang et al. [552], sentences are tokenized with SentencePiece [251].

last and penultimate interactions; the training set consisted of the remaining interactions. In the next section, we describe how a Universal Sentence Encoder can be used to build representations of sentences found in reviews.

5.5 Semantic matches

5.5.1 Semantic representations of reviews

The Universal Sentence Encoder (USE) is a sentence embedding model pretrained on a variety of NLP tasks with diverse degrees of supervision, namely Retrieval Question-Answering, Translation Ranking and Natural Language Inference [52]. It can be thought of as a paraphrase similarity model, and Yang et al. [552] showed such embeddings were useful for a variety of downstream NLP tasks. We encoded review sentences in a single high-dimensional space using the USE model. To illustrate the intuition for the relevancy of USE on review sentences, we applied it to three popular and distinct categories of items. We selected three items per category, according to the genre tag found on the Internet Movie Database[6]. The sentences found in the train reviews of the nine items were embedded in the USE model's 512 dimensional semantic space and we show in Figure 5.1 their 2D t-SNE [512] projection. We note that individual items have clustered sentences. On top of that, we observe that the projections of categories are Y-shaped, indicating some clustering at the category level as well, though sentences at the origin seem to share a high similarity across categories. We note many of the latter sentences, with low semantic variability, are general sentences, not specific to any item, such as "I liked it".

5.5.2 Aggregation of matches per-item category

We computed the k-NN sentence graph G of all the sentences found in the 20-core Amazon movie review trainset, embedded with USE. It is directed since each edge leaves sentence σ_1 and enters their k nearest neighbors σ_2. Following the approach of Yang et al. [552] for transfer learning, we used the cosine distance (defined as 1 minus

[6] https://www.imdb.com/

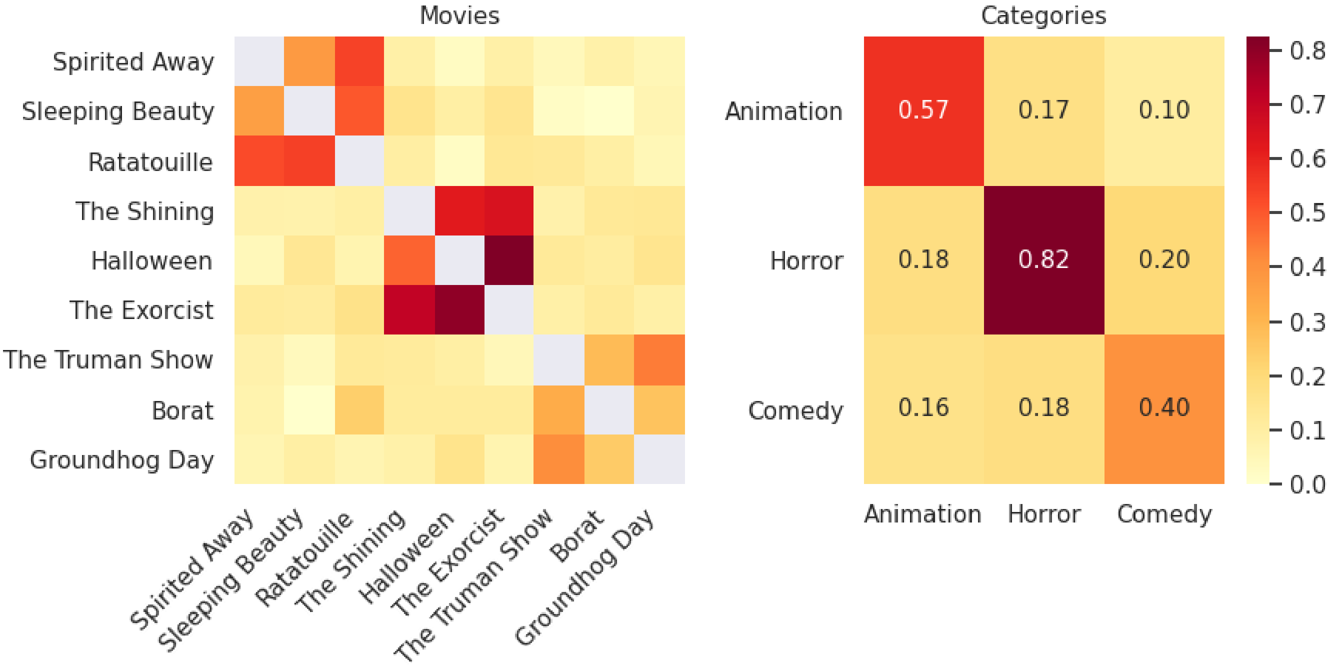


Figure 5.3: Heatmaps of k-NN sentence embedding matches where $k = 10$. Rows and columns respectively represent head and tail vertices. Same-item matches are discarded. Left: matches aggregated per movie. Right: matches aggregated per category.

the cosine similarity)[7]. Doing so echoes the approach we introduced in Chapter 3 when we evaluated sentence similarity with SIM (cf. Section 3.9.2 for more details). We call a "semantic match" a pair of connected sentences in the graph. Figure 5.3 shows the heatmaps of matches in the sub-graph of the nine mentioned items, aggregated by items (left) and categories (right). Since the items have different numbers of train sentences, the match count matrices C are normalized with the square root of its row and column weights [190], i.e., the heatmaps correspond to the matrices computed in Equation (5.1).

$$\text{diag}\left(C\mathbf{1}\right)^{-\frac{1}{2}} \cdot C \cdot \text{diag}\left(C^{\top}\mathbf{1}\right)^{-\frac{1}{2}} \tag{5.1}$$

It appears that the relative number of matches is higher for pairs of sentences belonging to the same category of items.

Visualization and quantitative analysis of matches indicate that one can represent item reviews with USE and build some recommender system from it.

[7] USE's embedding are normalized on the unit sphere.

5.6 Text-based k-nearest neighbors recommenders

Text-KNN estimates the test rating as the following weighted sum:

$$\hat{r}_{ui} = \frac{\sum\limits_{v \in N_i^{k'}(u)} w\left(u, v\right) \cdot r_{vi}}{\sum\limits_{v \in N_i^{k'}(u)} w\left(u, v\right)} \tag{5.2}$$

where $w\left(u, v\right)$ is the weight of v for predicting u's rating. Among all the possibilities to compute $w\left(u, v\right)$, we present below three alternatives based on counting the semantic matches. $N_i^{k'}$ is the set of k' nearest neighbor users of user u who have rated item i.

A simple method, called "baseline" in our experimental setups and introduced by Koren [242], estimates ratings b_{ui} as the overall average μ rating summed to user and item biases (respectively b_u and b_i). We experimented with the baseline-aware user k-NN (Text-BKNN) variant:

$$\hat{r}_{ui} = b_{ui} + \frac{\sum\limits_{v \in N_i^{k'}(u)} w\left(u, v\right) \cdot \left(r_{vi} - b_{vi}\right)}{\sum\limits_{v \in N_i^{k'}(u)} w\left(u, v\right)} \tag{5.3}$$

We proposed a user similarity measure based on the representations of sentences seen in Section 5.5.1.

Notations Let $G = \left(V, E\right)$ be the k-NN directed graph of sentences. Each directed edge $\left(\sigma_1, \sigma_2\right) \in E$ has an associated semantic similarity weight $s\left(\sigma_1, \sigma_2\right) \in \left[0, 1\right]$, detailed in Section 5.7. By extension, $s\left(\sigma_1, \sigma_2\right) = 0$ if $\left(\sigma_1, \sigma_2\right) \notin E$. The three approaches experimented with computing $w\left(u, v\right)$, denoted "One-to-One", "Many-to-One" and "Many-to-Many", rely on the fact that the set of sentences is partitioned by the set of users. For that reason, let S_u be the set of sentences written by u.

$N\left(\sigma_1\right) = \{v \in U : \exists \sigma_2 \in S_\nu, s\left(\sigma_1, \sigma_2\right) > 0\}$ is the set of users v whose sentences appear at least once in the neighborhood of sentence σ_1.

$N\left(u\right) = \bigcup_{\sigma_1 \in S_u} N\left(\sigma_1\right)$ is the set of users v whose sentences appear at least once in the neighborhood of sentences written by user u. Indicators are denoted by I.

One-to-One matching First, we considered u and v to have expressed similar preferences if v wrote at least one sentence in the semantic neighborhood of at least one sentence written by u, i.e.

$$w\left(u, v\right) = I\{v \in N\left(u\right)\} \tag{5.4}$$

Many-to-One matching Alternatively, we computed the occurrence of v's sentences in the neighborhood of u's sentences, i.e.

$$w\left(u, v\right) = \sum_{\sigma_1 \in S_u} I\{v \in N\left(\sigma_1\right)\} \tag{5.5}$$

Many-to-Many matching Our third approach consisted in counting the number of sentence matches when considering all pairs of sentences written by u and v. In this case,

$$w\left(u, v\right) = \max\left[\sum_{\left(\sigma_1, \sigma_2\right) \in \left(S_u \times S_v\right)} s\left(\sigma_1, \sigma_2\right), 0\right] \tag{5.6}$$

As we shall see in the next section, $s\left(\sigma_1, \sigma_2\right)$ is not necessary non-negative, though Equations (5.2) and (5.3) require non-negative weights.

5.7 Sentence graph

The weight $s\left(\sigma_1, \sigma_2\right)$ introduced in section 5.6 corresponds to the semantic similarity of sentence σ_2 regarding σ_1. Again, we considered various ways of computing it: “binary”, “continous” and “polarized”.

Binary count The basic sentence weight consisted in counting 1 if and only if the edge from σ_1 to σ_2 appears in the graph.

Continuous similarity scores To include finer information about sentence similarity, we experimented with the edge weight to be the cosine similarity scaled between 0 and 1. We defined s by:

$$s\left(\sigma_1, \sigma_2\right) = \frac{1 + \text{cosine similarity}\left(\sigma_1, \sigma_2\right)}{2}$$

Polarization Supplementary information at the sentence level may be relevant where semantic matching fails. For instance, the sentences ``I love DiCaprio'' and ``I hate DiCaprio'' have a cosine similarity of 0.94. We proposed to integrate information from a sentiment attribution mechanism $\alpha(\sigma) \in [-1, 1]$ with a polarization function p defined in equation (5.7). Then, the sentence weight becomes $p \cdot s$.

$$p\left(\sigma_1, \sigma_2\right) = \begin{cases} 1, & \text{if } |\alpha(\sigma_1) - \alpha(\sigma_2)| \leq 1 \\ -1, & \text{otherwise} \end{cases} \tag{5.7}$$

We experimented with a rating-aware sentiment attribution defined in equation (5.8).

$$\alpha(\sigma) = \begin{cases} 1, & \text{if } r\left(\sigma\right) \geq 4 \\ -1, & \text{if } r\left(\sigma\right) \leq 2 \\ 0, & \text{otherwise} \end{cases} \tag{5.8}$$

$r(\sigma)$ being the rating of the review σ belongs to.

Item graphs Moreover, we experimented with the methods discussed above, considering graphs made of sentences of a single item, rather than a single "global" graph of sentences from all items. The final user weight $w\left(u, v\right)$ was the sum of the per-item user weights, though other aggregations may be considered.

Normalization Some users write more sentences than others, some items receive more sentences than others, and users may write a different number of sentences on various items. For all these reasons, bias may appear when the user weights are computed from counting semantic matches between sentences (or co-occurrence like in the ablation study in Section 5.10). To mitigate this potential source of bias, we integrated different ad hoc normalizations in the computation of the user weights.

On the one hand, we tried to normalize $w\left(u, v\right)$ by the number of sentences $|S_v|$ written by v. In addition to this option, we considered normalizing the weights by the number of sentences $|S_i|$ written on i or the number of sentences $|S_{ui}|$ written by u on i when per-item graphs were considered. On the other hand, bias induced by an imbalance of sentence set sizes could also be mitigated by normalizing the match counting methods introduced in section 5.6. Concerning the One-to-One matching (Equation (5.4)), we tried to normalize by $N\left(u\right)$. When considering Many-to-One matching, each term in Equation (5.5) was divided by 1 or $N\left(\sigma_1\right)$. For the Many-to-Many matching, the terms in the Equation's (5.6) sum are divided either by 1, $\delta^- = \sum_{\sigma \in V} |s\left(\sigma, \sigma_2\right)|$ or $\delta^+ = \sum_{\sigma \in V} |s\left(\sigma_1, \sigma\right)|$. The latter two options respectively correspond to extensions of in and out-degrees adapted to our weighted graph. Normalizing by degrees is unsupervised mitigation of semantic similarity between "common" sentences, which are irrelevant for the sake of representing user preferences (cf. Section 5.5.1).

5.8 Evaluation measures

Offline evaluation of rating prediction is popularly measured through RMSE. However, Cremonesi et al. [91] showed the limits of pure error-based metrics.

As an example to understand RMSE's limitations, consider the toy dataset in Table 5.3. Neighborhood-based CF clearly indicates that u_2 behaves like u_0 and the ground truth tells that u_2 prefers i_1 over i_0.

Item User	i_0	i_1	i_2	i_3
u_0	2	4	1	5
u_1	5	1	5	1
u_2	**2.5**	**3.5**	1	5
$u_3 \dots u_n$	5	5	5	5

Table 5.3: Toy dataset of ratings. Bold values are in the test set, and the remaining values are in the train set.

The baseline asymptotically predicts $\lim_{n\to\infty} \hat{r}_{u_2 i_0}(n) = 3^+$ and $\lim_{n\to\infty} \hat{r}_{u_2 i_1}(n) = 3^-$. Therefore, it yields $\lim_{n\to\infty}$ RMSE$(n) = 0.5$, relatively good compared to the uniform random ($\mathbb{E}$ [RMSE] $= \frac{17}{16}$). However, the baseline is non-personalized when it comes to compare rankings, and always predicts $\hat{r}_{u_2 i_0}(n) > \hat{r}_{u_2 i_1}(n)$. This example illustrates RMSE's limitation in reflecting user preferences modeled by systems. This motivates a ranking-based metric measuring how well the ordering of items is preserved.

We argue that the Fraction of Concordant Pairs (FCP), measuring the proportion of well-ranked item pairs, is a suitable metric for three reasons. First, it directly and unambiguously measures preferences expressed by users, by definition. Second, it is grounded in statistics since concordant and discordant pairs have already proved relevant to comparing two measured quantities [231]. Third, it generalizes to non-binary ordered sets the ROC-AUC binary metric, known to test whether positive examples are ranked higher than negative examples by classifiers.

With TFCP, we handle equalities the same way as Hug [206], i.e. $((r_{ui}, \hat{r}_{ui}), (r_{uj}, \hat{r}_{uj}))$ is:

- Concordant (CP) iif $r_{ui} \neq r_{uj}$ and sgn$(r_{ui} - r_{uj}) =$ sgn$(\hat{r}_{ui} - \hat{r}_{uj})$,
- Discordant (DP) iif $r_{ui} \neq r_{uj}$ and sgn$(r_{ui} - r_{uj}) \neq$ sgn$(\hat{r}_{ui} - \hat{r}_{uj})$,
- Ignored if $r_{ui} = r_{uj}$

In the toy example, the baseline has a FCP of 0, signaling its inability to model u_2's preferences.

Koren and Sill [243] randomly split the dataset into train and test data, enabling the computation of FCP on pairs of test items rated by the same user. Yet, modern time-based *Leave One Last Item* splitting strategies provide one single test item per user. We adapted FCP to consider, for each user, all the pairs made of the test item and a train item. We called it *Time-based FCP* (TFCP). Thereby, the evaluation is equivalent to assessing the frequency of the model to correctly rank the next rating compared to all past ratings for each user. Denoting respectively n_c and n_d the number of concordant and discordant pairs, we reported the per-user macro-averaged metric: TFCP $= \sum\limits_{u \in U} \frac{n_c(u)}{n_c(u)+n_d(u)}$.

5.9 Results

Experimental setup We ran a grid search (corresponding to 276 search trials) over the set of parameters and options described in the previous section. For both Text-KNN and Text-BKNN, we selected the models yielding the best

Option Model	User similarity	Sentence weight	Polarization	Normalization	Type of graph
Text-KNN-R Text-BKNN-R	Many-to-Many One-to-One	Continuous Binary	Yes No	Out-degree 1	Global graph
Text-KNN-F Text-BKNN-F	Many-to-Many	Continuous	Yes	In-degree and $\|S_v\|$	Item-graphs

Table 5.4: Result of the hyperparameters (columns) tuning when models (rows) are optimized on each validation metric (Amazon dataset). For details about the normalization methods explored, see Section 5.7.

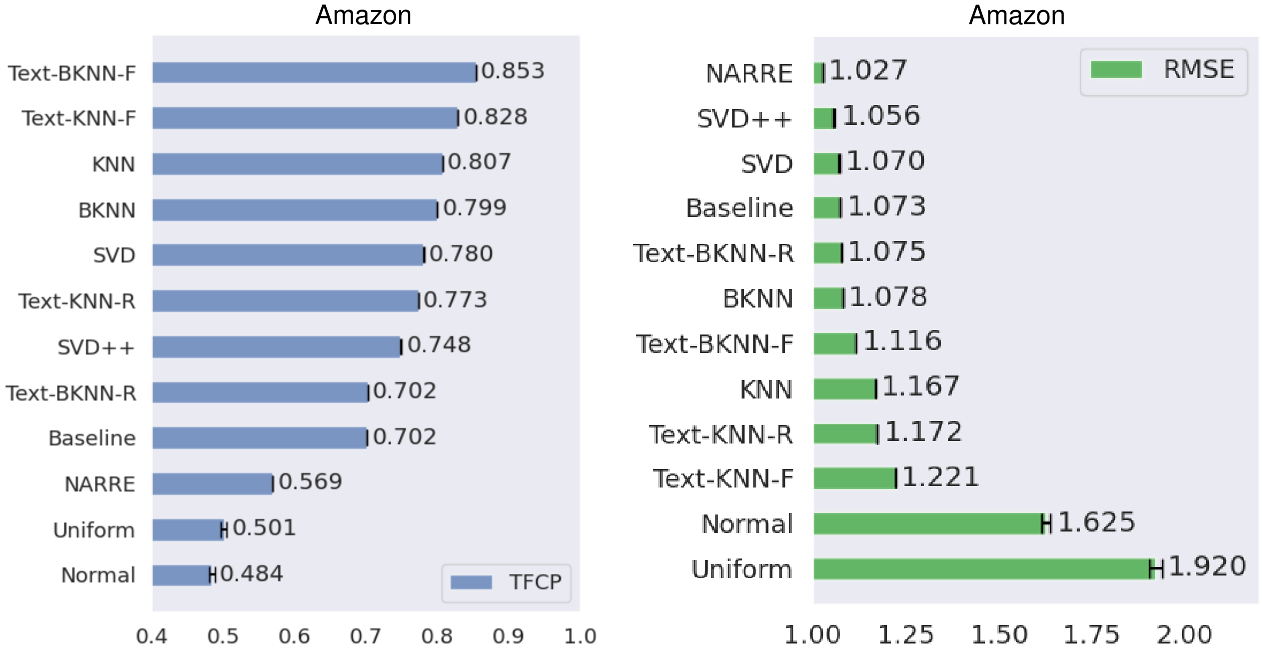


Figure 5.4: Bar charts of the models' test scores (average and standard deviation) on the **Amazon** dataset.

validation RMSE (-R), and TFCP (-F) on the Amazon 20-core dataset. The set of parameters resulting from the tuning is found in Table 5.4 and discussed in Section 5.10. We compared our approach to three baselines (two random systems: Uniform, Normal, and a popularity-based baseline, defined in Section 5.6), two popular memory-based methods (the rating-based KNN and BKNN, with MSD as similarity), and three state-of-the-art[8] model-based systems (SVD, SVD++ and NARRE [69]). NARRE is a Deep Learning model relying on a neural attention mechanism to make recommendations and decide which reviews are most relevant concurrently. Unless specified otherwise, their hyperparameters are the default parameters from Hug [206]. Specifically, all k-NN regressors had a number of *user* neighbors $k' = 40$ and text k-NN graphs used a number of *sentence* neighbors $k = 10$. For random baselines, SVD and SVD++, we repeated 10 times training and evaluation with different random seeds. In order to assess how the best set of hyperparameters found on the Amazon dataset generalizes to other datasets, we trained and evaluated Text-KNN-F and Text-BKNN-F on the Yelp dataset, i.e., without any hyperparameter tuning specific to that dataset. Figure 5.4 and Figure 5.5 compare the performances of different models over both metrics.

[8] Relative to error-based metrics and rating prediction tasks.

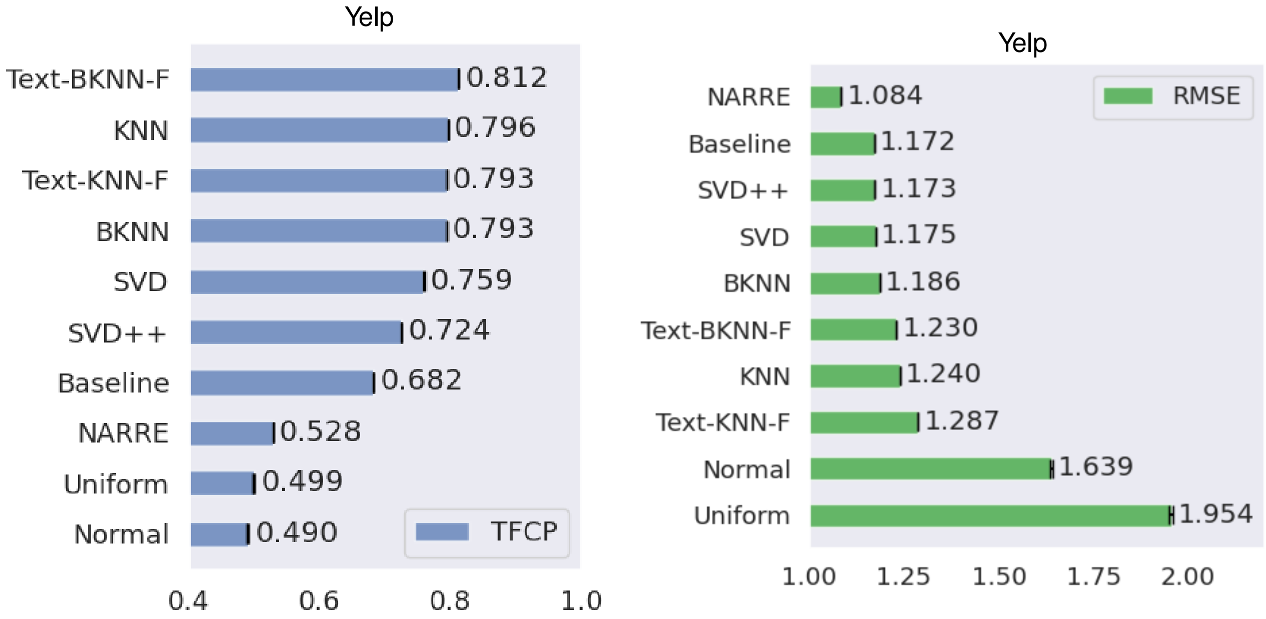


Figure 5.5: Bar charts of the models' test scores (average and standard deviation) on the **Yelp** dataset.

5.10 Discussion

Quantitative evaluation indicates that our approach is comparable to previous text-agnostic memory-based systems for both RMSE and TFCP. A general trend is that model-based systems give better RMSE but worse TFCP. Surprisingly, RMSE suggests the baseline's performance is better than all memory-based system, although being a naive approach. This echoes our analysis in Section 5.8 and validates the motivation behind the ranking-based evaluation. TFCP does indeed rank the non-"rankingwise personalized" baseline in penultimate place among non-random systems. Similarly, NARRE outperformed all other systems when evaluated with RMSE while being only slightly better than random predictions regarding TFCP. For further analysis of the correlation between RMSE and TFCP, Figure 5.6 shows Spearman's ρ and Kendall's τ correlation coefficients of the rankings produced by the metrics. RMSE and TFCP produce decorrelated rankings.

After hyperparameter tuning on the Amazon dataset, our systems are ranked first on TFCP, while showing competitive RMSE scores. In particular, the metric ranks text-based systems as the best among memory-based. Even when trained and evaluated on the Yelp dataset, our models rank first and third if we do not tune the hyperparameters specifically for that dataset.

Furthermore, results show that one can tune our approach's hyperparameters to optimize each metric. Doing so enables text-based models to yield better evaluation scores than their text-agnostic versions (e.g., once optimized for a metric, text-BKNN has a better score than BKNN), except for KNN on the RMSE metric. There is no winner-take-all hyperparameter set, achieving best results on both RMSE and TFCP. We observed that three out of four of the tuned text-based models involved the Many-to-Many matching and polarization. The latter signals the

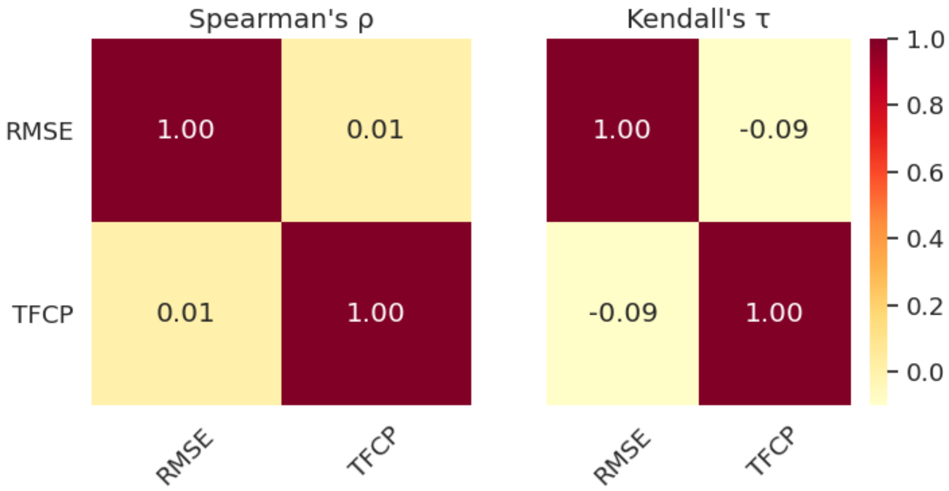


Figure 5.6: Correlation coefficients of the rankings produced by the metrics when considering the systems in Figure 5.4.

relevancy of sentiment awareness for sentence comparison. Incidentally, the TFCP-optimized text-based models with rating-agnostic computation of user similarity (i.e. without polarization) had TFCP $= 0.849$ when trained and evaluated on the Amazon dataset. When optimizing TFCP, both Text-KNN and Text-BKNN made the same set of hyperparameters stand out (cf. Table 5.4's second row). According to this metric, the best way of computing user similarity with semantic matches is to use the fine-grained Many-to-Many match count, with continuous similarity and polarization information, when per-item graphs are considered. On the contrary, the best Text-based model for RMSE (Text-BKNN-R) was obtained with the coarse-grained One-to-One matching, using minimal information from the binary count. It is worth noting that in the latter case, the global graph is preferred. Plus, we also considered filtering out matches with sentence similarity below a certain threshold, but interestingly, early experiments showed no quantitative benefit.

Ablation study We observed that our best text-based models involved per-item graphs instead of the global graph. To assess the benefit of semantic similarity, we studied the performance of three naive implementations counting the co-occurrence of sentences written by a pair of users u and v on an item i. The weights $w\left(u,v\right)$ was computed by summing user weights $w_i\left(u,v\right)$ over all items i. The implementations we tried were:

- $w_i\left(u,v\right) = I\{|S_{ui}| > 0$ and $|S_{vi}| > 0\}$, i.e. the similarity between users u and v relative to item i is 1 if both users reviewed i, and 0 otherwise.

- $w_i\left(u,v\right) = |S_{ui}| \cdot I\{|S_{vi}| > 0\}$, i.e. the similarity between users u and v relative to item i is proportional to the number of sentences written by u on i if v reviewed i, and 0 otherwise.

- $w_i\left(u,v\right) = |S_{ui}| \cdot |S_{vi}|$, i.e. the similarity between users u and v relative to item i is proportional to the number of sentences written by u and v on i.

After selecting the best option on the Amazon validation set, we found the best test scores to be RMSE $=$ 1.197 and TFCP $= 0.787$. Even though the evaluation scores indicate existing signals from this similarity-agnostic approach, we see significant improvement and interest of integrating semantic similarity and finer relationships between sentences.

Explainability Besides quantitative performances, our text-based approach has benefits regarding the explainability of automatic recommendation. Table 5.5, Table 5.6, Table 5.7 and Table 5.8 show pairs of sentences from the Amazon train reviews written by some user u and u's nearest user neighbor v. Both users u and v liked u's test item j, i.e. $r_{uj} \geq 4$ and $r_{vj} \geq 4$. We manually selected sentence pairs counted as semantic matches by the system and relevant for explaining why the system predicts that u's opinion on j ($\hat{r}_{uj}$) should resemble v's opinion on j (r_{vj}), through a high user similarity $w(u, v)$. For readers unaware of the items, we added relevant attributes and commonalities relative to items. Table 5.5 and Table 5.6 correspond to the case of per-item graphs while table 5.7 and table 5.8 gather matches between sentences from different items in the global graph. The latter tables differ from each other in whether tail sentences (written by v) review u's test item j or not, which is equivalent for the recommender system but may matter in human interpretation.

5.10.1 Qualitative analysis of random semantic matches

Table 5.9 provides random matches found in 5 random per-item graphs. The first two examples have lower semantic similarity than the last three examples. Both sentences of the first example are descriptive, and the association focuses on the ``killing'' aspect. The second match is even unclearer; the only commonality is a mention of some female character/actress. The third example is made of a long and detailed tail sentence. The explicit reference to the Madea cinematic universe and the emphasized funniness are expressed in both sentences. The fourth pair of sentences refers to both the item's name they review as well as the ``family'' audience. We remark that mentionning an item name or title in review sentences may mislead the whole system, as it can trigger irrelevant semantic matches, especially if names are long. The fifth example includes a direct recommendation for lovers of suspense in both sentences.

5 random matches found in the global graph are shown in Table 5.10. The first match shows a pair of sentences reviewing the same item in which the item's name does not appear. Even if the director's name is common to both sentences, references to the Battle of Mount Austen are phrased with different vocabulary. Indeed, the head sentence refers to ``empathy and apathy'' during ``the conflict at Guadalcanal'' while the tail sentence describes ``lush beauty of the South Pacific'' contrasting with ``the destruction of war''. The second and third examples are short sentences with almost identical head and tail sentences. These matches convey limited

u'S TEST ITEM j	TRAIN ITEM i CO-REVIEWED BY u AND v	COMMONALITY SHARED BY BOTH i AND j	MATCHING SENTENCES IN THE TRAIN SET
Justice League: The Flashpoint Paradox	Green Lantern: First Flight	Animated superhero film	You simply must check out this awesome superhero epic . Personally , it's the finest superhero animated effort ever brought to screen .
Fort Apache	The Searchers	Western film directed by John Ford and starring John Wayne	That aside , and at the risk of repeating myself it's vintage John Ford and John Wayne with some magnificent scenery . This is one of John Ford's and one of John Wayne's best movies .
Peter Pan	Monsters, Inc.	Animated film	This is a movie that I ' d suggest any family to pop in and have a family movie night . a great family movie .
Killing Them Softly	Django Unchained	Violence & Gore	This movie is Bloody , gory , violent , emotional , serious and hilarious . All in all . . . . a really good movie , but a bit bloody .
1776	Downfall	Historical drama film	It is almost a must see for historians and those with an interest in history . Should be required viewing for high school history students .
Black Swan	Coraline	Dark drama film with frightening & intense scenes	It can be immensely dark and scary at times - even for adults ! The story and the situations became spooky . . . (button eyes) , and may be a little intense for younger children .
Shallow Hal	Bridget Jones's Diary	Romantic comedy	It was wonderfully funny and romantic . This is a totally winning romantic comedy !
Clash by Night	Niagara	'50s film noir starring Marilyn Monroe	Great acting by all parties : Monroe , Cotten , Peters et . Monroe aside , this movie is definitely a Joseph Cotten vehicle and may be his finest acting job from a long career .
Rocky III	Rocky II	Installment in Rocky franchise	Sylvester Stallone delivered a movie with great screenplay , great characters , a great plot and all together a wonderful movie everyone will remember . Stallone is good in all his roles , the playing of Rocky as a working class bum made good is bang on .

Table 5.5: Examples of semantic matches when our system considers **per-item graphs**. The last column shows the head sentence first (written by u) and the tail sentence then (written by v). Additional examples from the same setup are shown in Table 5.6.

u's test item j	Train item i co-reviewed by u and v	Commonality shared by both i and j	Matching sentences in the train set
Fahrenheit 9/11	Bowling for Columbine	Documentary film directed by political commentator and left-wing activist Michael Moore	Charlton Heston , like any old republican relic , is a bitter twisted old man . Anyone who actually thinks that Michael Moore was being unfair to Charlton Heston in his interview just can ' t accept the fact that Heston leads a extremist group that can ' t possibly defend it's policy goals .
Evil Dead (2013, soft reboot)	Evil Dead II (1987)	Installment in Evil dead franchise	evil dead 2 is just like the first one with the thrills and chills we have come to love and expect . Evil Dead 2 is ok , but not as good as the first .
Beauty and the Beast 2	Robin Hood	Animated film produced by Disney	All the characters are likable & funny . This is a fun movie with a lot of likable characters and fun songs .
The Alamo	Rio Bravo	Western film starring John Wayne	The plot is pretty good with John Wayne leading the action . John Wayne is great as usual .
Imitation of Life	Peyton Place	'50s american drama film starring Lana Turner	Lana Turner's performance does sometimes border on camp , but would we have her any other way ? Lana Turner gives her best performance , while Hope Lange steals the movie as a haunting and wonderfullytouching Selena .
Westworld	Futureworld	Installment in Westworld franchise	You have to watch Westworld first . . . . then Futureworld , the sequal . " Westworld " was a great movie that was highly acclaimed , but there has been much criticism over its sequel " Futureworld " , and I can ' t see why - it's a great movie !

Table 5.6: Examples of semantic matches when our system considers **per-item graphs**. The last column shows the head sentence first (written by u) and the tail sentence then (written by v). Additional examples from the same setup are shown in Table 5.5.

u's TEST ITEM j	ATTRIBUTE OF j	MATCHING SENTENCES IN THE TRAIN SET
Hancock	Action film	**To Live** – Usually I ' m bouncing off the walls and watching a crazy action flick . **Gladiator** – Sometimes I just fast forward straight to the epic battle scenes .
Non-Stop	Action thriller film	**The Debt** – Plot is very slow developing , so slow and uninteresting , did not even finish watching . **Whiteout** – Plot was so S-L-O-W and dull .
Dune	Action-adventure science-ficton film	**Oblivion** – Good action and story line . **Serenity** – great action and story line .
Evil Dead	Horror film	**Suspiria** – A legend in the horror movie genre . **Prince of Darkness** – A horror movie classic .
Rocky II	Written by and starring Sylvester Stallone	**Rambo III** – His range may be limited , as we saw in his ' comedy ' films , but when he sticks to his forte , playing great heroes , Stallone is the greatest ever . **Rocky** – But Stallone gives just about the best performance of his career here .
The Living Daylights	Entry in the James Bond (a.k.a. *007*) series	**Never Say Never Again** – " Never Again " ultimately retains a very watchable magic featuring the original Agent 007 one last time . **Goldfinger** – After the first two 007 films , this third Bond adventure cemented forever the style and fun of the series .
My Neighbor Totoro	Animated film	**Popeye** – Hey parents , want a good , clean , wholesome movie for your kids ? **Kiki's Delivery Service** – An excellent movie for kids that parents don ' t have to worry about .

Table 5.7: Examples of semantic matches when our system considers the **global graph**. The last column shows the head sentence first (written by u) and the tail sentence then (written by v), along with the respective item's review they belong to. Here, tail sentences do not review j but express attributes also present in j and indicated in the second column.

HEAD SENTENCE σ_1 WRITTEN BY u ON SOME ITEM i	TAIL SENTENCE σ_2 WRITTEN BY v ON u'S TEST ITEM j AND MATCHING σ_1
Cemetery Man – This highly entertaining little zombie movie from Italy has all the elements that make it a wonderfully dark horror - comedy in the same vein asEvil Dead 2 : Dead by DawnandAn American Werewolf in London .	**The Horde** – one of the best horror zombie movies of all the times , this movie is equal than 28 days later , in these days European horror movies are the best of the best . . good for Friday at night
Charlie's Angels – I like drew barrymore , she is the best angel out of the three .	**Fever Pitch** – Drew Barrymore , as always , is phenomenal .
Ice Age – While not perfect , it is full of laughs and beautiful computer animation .	**Finding Nemo** – Highlights : Spectacular computer animation ; hilarious , well - developed characters ; original plot .
Monsters, Inc. – Great for parents and kids (or people without kids) .	**Toy Story 3** – Great for kids and adults alike .
Island in the Sky – The Duke[a] does a great job in his role as Dooley - the plane's captain .	**The Alamo** – The Duke turns out one of his best performances , as well as putting together this film .
I Am Legend – At the end of the movie you fail to realise it was just one man , Will Smith in most of the scences and yet the movie is neither boring , or lacking in elements that make for a great thriller .	**Hancock** – Will Smith adds a lot of flare to the movie , when it could ' ve been bland and cheesy .
War of the Worlds – The special effects were great and the acting was believable .	**Munich** – The acting , story line and special effects were great .
Red River – I think this is my favorite early John Wayne film (it's not exactly early , but it was one of his earlier big hits) .	**The Searchers** – This is John Wayne's favorite " John Wayne movie , " and his acting is superb .
Bubba Ho-Tep – Fans of Bruce Campbell will love this movie , but I don ' t know how fans of the King will take it .	**My Name Is Bruce** – First of all if your not a bruce campbell fan or dont enjoy his movies then you probably wont like this one as it is trademark campbell . . . .

[a] A nickname for the American actor John Wayne

Table 5.8: Examples of semantic matches when our system considers the **global graph**. Sentences are preceded by the item they review. v represents u's nearest user neighbor. Here, tail sentences review j, though tail sentences can just as well review other items and be counted as semantic matches. as in Table 5.7.

ITEM	SEMANTIC SIMILARITY	MATCHING SENTENCES IN THE TRAIN SET
Spartan	0.69	/ / Spoiler alert / / Our hero , of course , is a loner - - which is a sure tip - off that anyone he sort of becomes close to is going to get killed . Kilmer's character survives and even thrives in this morass because he is an unsentimental machine , the " Spartan " ideal of an all - male world where it's kill or be killed , and there's plenty of both - even a suicide .
Shopgirl	0.66	I have a feeling there is probably more of her left on the cutting room floor . It's no surprise that she's on medication for depression .
Madea's Witness Protection	0.87	This movie is a very funny movie and Madea is trying to be a private Detective and she finally gets her man This Madea was the first time in the series to be released in the summer and did pretty well as the second highest grossing Madea film ever behind Madea Goes to Jail and this one actually crossed over with other audiences and other demographics besides his usual african - american audience and this one was the funniest film of the series and Tyler Perry's best film yet .
Radio	0.82	' Radio ' is a wonderful film that the whole family can enjoy and learn from together ! Radio is good , touching and very sad at times but it has a lot to learn about standing on its own as a family classic .
The Gift	0.75	Highly recommended for suspense fans . I would reccomend that anyone who likes mystery or suspense thrillers should go see this .

Table 5.9: Random examples of semantic matches when our system considers **per-item graphs**. The last column shows the head sentence first and the tail sentence then.

information when taken out of context. The last two matching pairs are triggered by common vocabulary or clauses but can't be considered matches in users' preferences.

5.10.2 Domain-specific sentence embeddings

In addition to the pretrained USE model, we experimented with domain-specific embeddings. First, we fine-tuned a pretrained T5 [405] LM on an item prediction classification task using the 2018 version of the Amazon Product Review dataset. We then applied our method to the representations learned by the fine-tuned T5 model in the hope that these representations would be finer than the more general USE representations. Results of baselines and our best models (cf. Figure 5.7) show that there is no clear advantage to using domain-specific sentence representations over general embeddings.

SEMANTIC SIMILARITY	MATCHING SENTENCES IN THE TRAIN SET
0.81	**The Thin Red Line** – / / Director Terrence Malick focuses on the conflict at Guadalcanal from ground up and showing empathy and apathy along the way . **The Thin Red Line** – Mr . Malick captures the lush beauty of the South Pacific and uses it to perfectly contrast with the destruction of war .
1.00	**Seabiscuit** – This is one of my all time favorite movies . **The Jazz Singer** – This is one of my all - time favorite movies .
0.86	**Tropic Thunder** – it stole the show . **My Week with Marilyn** –steals the show .
0.75	**Heist** – It's a known fact that he uses a metronome in order to keep his dialogue to have a certain rhythm to it . **Black Dynamite** – He draws attention to it by repeatedly glaring at the mic throughout the scene , but doesn ' t miss a beat of the dialogue .
0.88	**Grown Ups** – Now I will say from the get go this film is not for everybody and I have noticed that some people just don ' t get this film . **Barbarella** – This film is definately not for everyone & I ' d honestly recommend that most people rent it before they but it .

Table 5.10: Random examples of semantic matches when our system considers the **global graph**. The second column shows the head sentence first and the tail sentence then, along with the respective item they review.

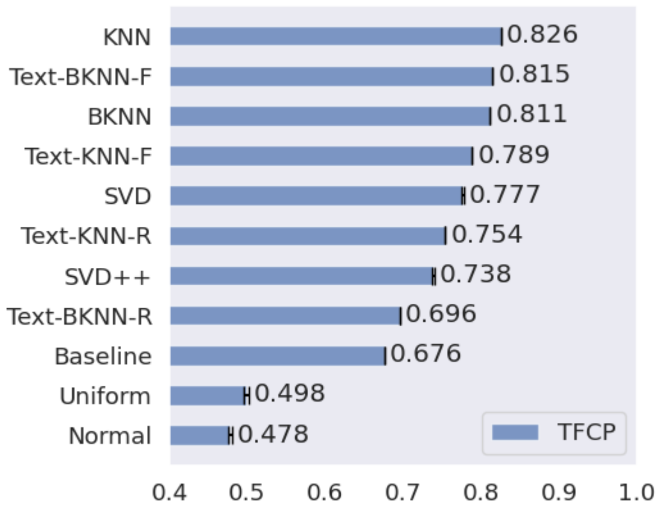


Figure 5.7: Bar chart of the models' test TFCP scores (average and standard deviation) on the 2018 version of the Amazon dataset, when we considered **domain-specific embeddings** rather than USE's embeddings.

5.11 Limitations

Regarding scalability and time complexity, the first limiting phase is the embedding of sentences. Embedding sentences with an on-the-shelf pretrained USE took a few hours using Apache Beam[9] on $1,000$ machines. Besides this step, the bottleneck characteristic of our method is the computing of sentence k-NN graphs. When per-item graphs are considered, we iterated Scikit-learn's 10-NN graph algorithm[10] on an average of ~ 700 train sentences per item. For reasonable amounts of items, this can be achieved quickly, though this becomes more demanding for large item sets. It took 3 minutes to compute all sentence graphs on the Amazon dataset and 23 minutes on the Yelp dataset. The global graph, though, is much heavier to compute. We used the ScaNN similarity search method [162] to compute the 10-NN global graph at scale on ~ 2.9 million sentences for the Amazon dataset and ~ 20 million sentences for the Yelp dataset. The computations were enabled by several hours of computing time on a 1TB RAM server. By comparison, the NARRE Deep Learning model was trained during 10 epochs with respective average epoch runtime of 17 minutes and 168 minutes for the Amazon and Yelp datasets, using an NVIDIA A100 GPU.

A second limitation to this work that we reserved for future work is a missing stronger theoretical background on FCP's relevance compared to RMSE, especially for contexts where the rating distribution is skewed (as is the case for the Amazon dataset).

Chapter 5 conclusion

This study explored, for the first time to our knowledge, the direct integration of semantic similarities from review text in recommender systems, through the representation power of modern pretrained NLP models. Our experiments quantitatively showed the benefits of this approach relative to memory-based models. Further, preference representation from text is a step toward better explainability of machine learning systems. Lastly, we argued in favor of ranking-based metrics to evaluate rating-prediction systems.

Future work includes evaluating explanations from our approach as well as building better representations of users and items, for example, by using more recent advances in language modeling or by joint tuning of a LM with the recommendation task. Unsupervised techniques could also help filter out irrelevant or redundant text. For instance, rationale extraction systems such as WT5 [355] could be employed as a preprocessing step to select sentences more likely to represent meaningful preferences. Another promising advance at the intersection of text and recommenders that could be built on is entity prediction, following the work of Zemlyanskiy et al. [566]. More generally, our work highlights an exciting and emerging intersection between the fields of Natural Language Understanding and recommender systems.

[9] https://beam.apache.org/

[10] https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.kneighbors_graph.html

Chapter 6

General conclusion and perspectives

Industrial revolutions transform societies by relying on emerging technologies. Scientific breakthroughs coupled with innovative engineering bring humans to design and implement novel machines, as well as usages impacting people's lives. After the digital revolution democratized computers and made Internet users globally connected, the Fourth Industrial Revolution, driven by Artificial Intelligence, seeks to blur the boundaries between the physical and digital worlds. In this game, Natural Language Processing plays a key role. Indeed, machines will be incredibly powerful and/or useful when they become able to hold complex reasoning automatically, and perhaps someday even *think*. Even though "the ability to speak does not make you intelligent."[1], language has long been thought of as a crucial part of the reasoning process. In *The Theaetetus*, Plato defined "thinking" with the Ancient Greek word λόγος[2] ("reason", "speech", "computation").

> "So then, I, for my part, refer to thinking as speaking, and opinion as speech."

In Chapter 2, we saw that computer scientists, together with mathematicians, linguists, and neuroscientists, have proposed algorithms to bridge the gap between machines and humans through automatic natural language understanding and generation. Recent progress in Machine Learning, enabled by weakly supervised Deep Learning architectures, software and hardware advances as well as increasingly larger datasets, have made NLP ripe for hitherto unseen applications. This thesis explored a subset of the possibilities offered by pretrained NLP models, focusing on assisting humans in their online interactions.

Chapter 3 introduced an original method (CAE-T5) to address toxic-to-civil transfer with pretrained Language Models, in the hope that it will nudge healthier conversations online. Specifically, we tackled the problem in a self-supervised manner, where the dataset we had access to was made of comments annotated in toxicity only. Given these unpaired examples, we were able to fine-tune a pretrained T5 model with an end-to-end denoising objective function requiring no previous sentence alignment. Even though human and automatic evaluations show that our

[1] Quote from Star Wars: Episode I – The Phantom Menace
[2] lógos

approach significantly outperforms baselines, issues regarding hallucination remain to be solved before the system can be implemented in real use cases. At the time of our project, no parallel dataset existed, but posterior works released small paired datasets, which we then used to train a strongly supervised sequence-to-sequence model (SED-T5). Chapter 4 revealed that this latter approach shows less hallucination but at the cost of lower fluency and detoxification accuracy. Future work should investigate ensembling CAE-T5 and SED-T5, or even training a single incompletely supervised model leveraging both ideas (cf. Section 2.4.2).

Besides, a new task was presented in Chapter 4. We released the first dataset annotated in toxicity at the span level in the hope that it will help future developments in this direction. We described the setup along with an evaluation metric suited to the task. We also experimented with a few baselines that were released too. In particular, we showed that weakly supervised models trained on potentially larger datasets annotated at the post level only could almost reach the performance of strongly supervised models. Furthermore, the ToxicSpans dataset proved useful for assessing the mitigation of explicit toxicity in toxic-to-civil transfer. The takeaways were twofold: first, models often successfully detect toxic spans and try to rephrase them, and then humans did rephrase almost all cases of explicit toxicity in the toxic posts they were given. The outcome of the online competition we organized using ToxicSpans in a new shared task (cf. Appendix A) has been satisfying. The strong participation both indicated the community's high interest and helped us identify avenues worth exploring. First, low performance from both human crowdworkers and Machine Learning systems on highly-context-dependent posts could be targeted with automatic context sensitivity estimation, as described in Xenos et al. [544]. Then, as systems struggle to predict empty toxic span when the post is either non-toxic, implicitly toxic, or toxic in its entirety, span detection models could benefit from a pre-filtering process detecting if a post falls in one of these categories.

Finally, Chapter 5 describes an unprecedented way of predicting items' ratings from users according to their reviews of past items. We demonstrated that our method may not only outperforms rating-only-based recommender systems but also provide explanations for the predictions. Additionally, we investigated a concrete application leveraging the metric space of semantic embeddings produced by a pretrained sentence encoder. Our approach based on k-Nearast Neighbors illustrates a substantial procedure using graph techniques and deep semantic representations. In the future, we wish to find efficient unsupervised strategies to select sentences written by users best representing their tastes. Our node ranking experiments were inconclusive but we think that a clever combination of PageRank and Language Models may lead to effective solutions.

All in all, this thesis was a journey in the hot field of Natural Language Processing. This field and our contributions provide evidence that Artificial Intelligence can assist people in their interactions online by improving social exchanges, moderating conversations, and offering explainable recommendations based on what Internet users express. There remain ethical concerns that should be addressed in a multidisciplinary approach, such as potential bias in the definition of toxicity, risks of automated moderation being used for political censorship, or energy-intensive computing to run our models.

The main challenges encountered throughout this thesis outlined the following general perspectives regarding evaluation, datasets, and modeling. First, evaluation metrics, especially when no or few paired examples are available, remain an open topic and raise central issues to the development of weakly supervised Machine Learning algorithms. Then, while we focused here on the English language, we are excited by the increasing number of multilingual resources (regarding data and pretrained models) currently available and wish to see our works transferred to more languages. Lastly, we would be happy to analyze the few-shot performances of large pretrained Language Models on the three tasks studied here: style transfer, ToxicSpans prediction, and rating prediction.

Appendix A

SemEval-2021 Task 5: Toxic Spans Detection

A.1 Task description

The Toxic Spans Detection task of SemEval-2021 required participants to predict the spans of toxic posts that were responsible for the toxic label of the posts, when detecting such spans is possible[1]. Systems had to extract a list of toxic spans, or an empty list, per post. The task could be addressed as supervised sequence labeling, using training data with gold toxic spans provided by us. It could also be treated as rationale extraction, using classifiers trained on potentially larger external datasets of posts manually annotated as toxic or not, without toxic span annotations. Participants submitted their predicted spans for a held-out test set, and were scored using character-based F_1 (cf. Section 4.5).

The evaluation period started on January 10, 2021, and finished on January 31, 2021. In the first week, 10 submissions were allowed per day per team. In the second week, this number was reduced to 5, and further to 1 during the final week. We chose to allow an extended evaluation period combined with multiple team submissions to promote the competition. However, we also dicided on a decreasing submission limit to make it harder for participants to overfit the test set. As shown in Figure A.1, the number of submissions dropped over time due to this constraint, but the interest was continuous, and there were submissions until the last day. Despite the decreasing total number of submissions per day, the top daily score increased, reaching its maximum on the last day (see Figure A.2).

[1] Although we defined the task at the word level, gold labels were provided at the character level, counting from zero.

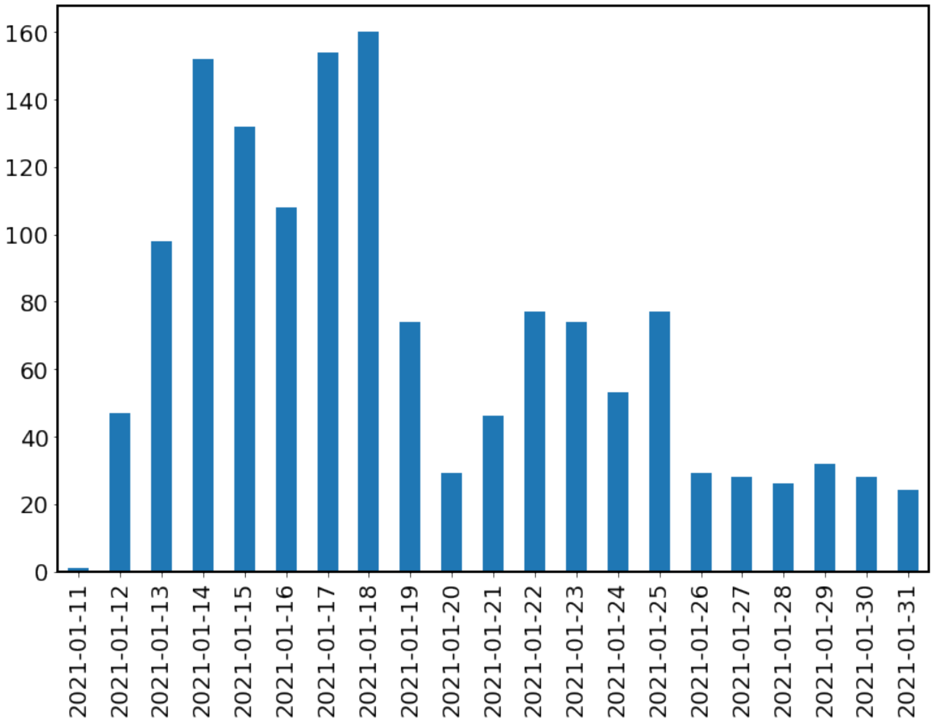


Figure A.1: Number of submissions per evaluation day.

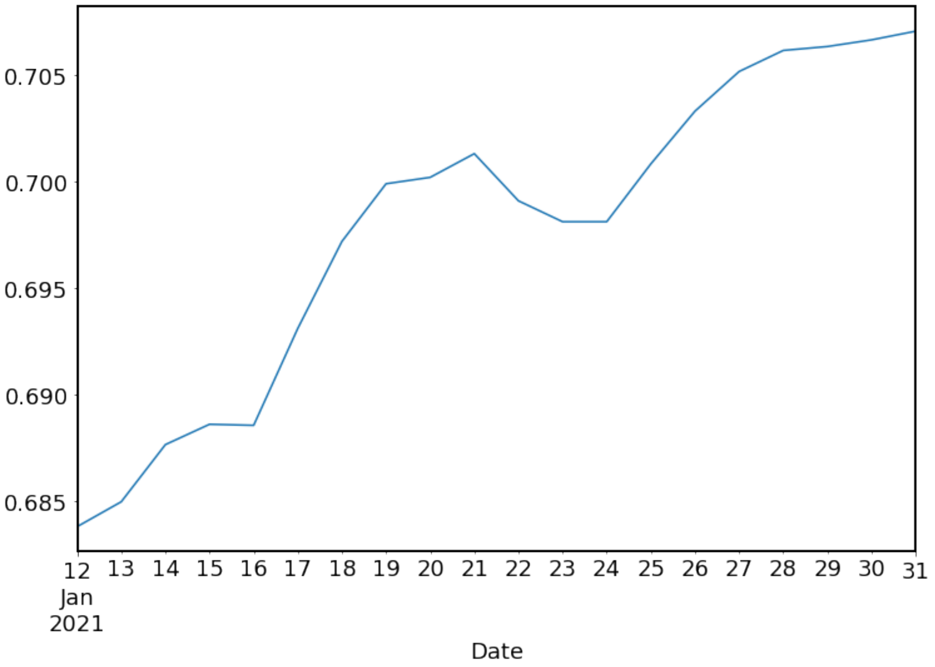


Figure A.2: The evaluation score (character F_1) of the best submission per day during the evaluation period.

A.2 Participation overview

We received 479 individual participation requests, 92 team formations, and $1,449$ submissions. 91 teams submitted valid predictions ($1,385$ valid submissions in total) and were scored; out of these, only 36 submitted system descriptions.

A.2.1 The HITSZ-HLT submission

The best performing team (HITSZ-HLT) formulated the problem as a combination of token labeling and span extraction [595].

For their token labeling approach, the team used two systems based on BERT [107]. Both systems had a Conditional Random Field (CRF) layer [496] on top, but one of the two also had an LSTM layer [193] between BERT and the CRF layer. In both approaches, word-level BIO tags were used, i.e., words were labeled as B (beginning word of a toxic span), I (inside word of a toxic span), or O (outside of any toxic span).

For their span extraction approach, the team also used BERT. Roughly speaking, in this case, BERT produces probabilities indicating how likely it is for each token to be the beginning or end of a toxic span. Then a heuristic search algorithm, originally developed for target extraction in sentiment analysis by Hu et al. [201], selects the best combinations of candidate begin and end tokens, aiming to output the most likely set of toxic spans per post.

The character predictions of the three systems described above were combined with majority voting per character. That is, if any two systems considered a character to be part of a toxic span, then the ensemble classified the character as toxic; otherwise, the ensemble classified it as non-toxic.

A.2.2 The S-NLP submission

The team with the second best performing system (S-NLP) consists of individual participants who grouped and submitted an ensemble of their systems [361]. The ensemble combines two approaches, both of which are based on a RoBERTa model [311]. The latter is first fine-tuned to classify posts as toxic or non-toxic, using three Kaggle toxicity datasets[2]. For toxic span detection, RoBERTa's subword representations from three different layers (1, 6, 12) are summed to produce the corresponding word embeddings. A binary classifier on top of RoBERTa, operating on the word embeddings, predicts whether a word belongs to a toxic span or not.

For the first component of the ensemble, the word embeddings obtained from RoBERTa's subword representations are concatenated with FLAIR [5] and FastText [38] embeddings.[3] The resulting embeddings are passed on to a two-layer stacked BiLSTM with a CRF layer on top to generate a BIO tag per word.

[2] `https://github.com/unitaryai/detoxify`

[3] In the latter case, in-vocabulary word embeddings were imported to Word2vec for efficiency, and out of vocabulary words were handled with BPEs [457].

The second component of the ensemble used the RoBERTa model as a teacher to produce silver toxic spans for $30,000$ unlabelled toxic posts [48]. RoBERTa was then retrained as a student on the augmented dataset ($30,000$ posts with silver labels and the training posts we provided) to predict toxic offsets.

The ensemble returns the intersection of the toxic spans identified by the two components.

A.2.3 Additional interesting approaches

We now discuss some of the most interesting alternative approaches tried by the participants, even if they did not lead to high scores.

Rationales Some participants experimented with training toxicity classifiers on external datasets containing posts labeled as toxic or non-toxic; and then employing model-specific or model-agnostic rationale extraction mechanisms to produce toxic spans as explanations of the decisions of the classifier. The model-specific rationale mechanism of Rusert [434] used the attention scores of an LSTM toxicity classifier to detect the toxic spans. Pluciński and Klimczak [392] used the same approach, but also employed an orthogonalisation technique [349]. The model-agnostic rationale mechanism of Rusert [434] combined an LSTM classifier with a token-masking approach that we call Input Erasure (IE), due to its similarities to the method of Li et al. [287]. The model-agnostic approach of Pluciński and Klimczak [392] combined SHAP [318] with a fine-tuned BERT model. Ding and Jurgens [110] and Benlahbib et al. [31] also experimented with model-agnostic approaches, but they combined LIME [420] with a Logistic Regression (LR) or with a linear Support Vector Machine (SVM) toxicity classifier. All the above-mentioned approaches used a threshold to turn the explanation scores (e.g., attention or LIME scores) of the words into binary decisions (toxic/non-toxic words).

Lexicon-based No team relied on a purely lexicon-based approach, but few experimented with lexicon-based baselines [595, 369] or used such components in ensembles [413]. Three kinds of lexicon-based methods were used. First, the lexicon was handcrafted by domain experts [474], and it was simply employed as a list of toxic words for lookup operations [369]. Second, the lexicon was compiled using the set of tokens labeled as toxic in our span-annotated training set, and it was used as a lookup table [57], possibly also storing the frequency of each lexicon token in the training set [595]. The former two were also combined [413]. Third, the least supervised lexicons were built with statistical analysis on the occurrences of tokens in a training set solely annotated at the comment level (toxic/non-toxic post) [434]. An added value of these approaches is that easy-to-use resources (toxicity lexicons) are built and shared publicly, such as the one suggested by Pluciński and Klimczak [392].[4]

Custom losses Zhen Wang and Liu [586] experimented with a new custom loss, which weighted false toxicity predictions based on their location in the text. If a false prediction was located near a ground truth toxic span, then it would contribute less to the overall loss for that post compared to one located further away. The loss function

[4] https://github.com/Orthrus-Lexicon/Toxic

used by Kuyumcu et al. [257] to train their system is the Tversky Similarity Index [508], a generalization of the Sørensen–Dice coefficient and the Jaccard index, which the authors adjusted to weigh up false negatives.

Data augmentation The vast majority of the participating teams employed additional training data annotated at the post level. That is, either to build lexicons [434], to leverage unsupervised rationale extraction methods [434, 392, 110, 31], or to filter posts [322] that were not labeled as toxic by a toxicity classifier. Suman and Jain [489] astutely produced silver data from external sources to augment the initial golden annotated dataset, training their model iteratively in a semi-supervised manner.

A.3 Results

Table A.1 shows the scores and ranks of all participating teams that described their approach, i.e., 36 out of 91 teams that participated. HITSZ-HLT (Section A.2.1) was ranked first, followed by S-NLP (Section A.2.2) that scored 0.06% lower. The rest of the teams followed with scores lower than 70%.

The score of the median system is 67.58%, which is not far below the top scored team (-3.22 percent units), while it is far above the last two ($+17.52$ percent units). The standard deviation of system scores above the median is much lower (0.94) than that of the systems below the median (4.12). Most teams that were excluded from the table (because they did not describe their methods) scored lower than the median. However, there were also top-scoring teams among those that were excluded, such as a team with a RoBERTa-based token-level ensemble that was ranked 4^{th}.[5]

A.4 Analysis and discussion

Overall we were happy to see the degree of involvement in this shared task and the resulting diversity of approaches to this problem. We include some of our observations regarding the evaluation administration and what we have learned from the results.

A.4.1 Participation

We reached out to teams that decided not to submit a description paper, and the vast majority were students who were time-limited. The fact that students participated in the task is promising, and we plan to consider more ways to introduce SemEval tasks in classrooms. On the other hand, 60% of the participants chose not to describe their approach, which is problematic and should be addressed. A team could take advantage of such an option to create

[5] We asked for details from participants that did not submit description paper, but not all of them replied.

RANK	TEAM	F_1 SCORE (%)
1	HITSZ-HLT	70.83
2	S-NLP	70.77
3	hitmi&t	69.85
5	YNU-HPCC	69.63
7	Cisco	69.22
8	MedAI	69.03
9	IITKDetox	68.95
13	GHOST	68.59
14	HLE-UPC	68.54
15	UTNLP	68.44
16	YoungSheldon	68.42
17	Lone Pine	68.38
18	sk	68.32
20	WLV-RIT	68.01
21	CSECUDSG	67.95
22	LISAC FSDM USMBA	67.84
23	UoT-UWF-PartAI	67.70
25	uob	67.61
MEDIAN	The median score	67.58
26	UAntwerp	67.55
27	MIPT-NSU-UTMN	67.55
28	NLRG	67.53
30	HamiltonDinggg	67.15
33	lz1904	67.00
34	UIT-E10dot3	66.99
36	UniParma	66.72
37	hub	66.40
38	GoldenWindPlymouth	66.37
41	AStarTwice	66.16
44	sefamerve_arge	66.01
46	UPB	65.73
49	Entity	65.61
57	BennettNLP (Fuchsia)	64.53
58	TeamGriek	64.31
63	UIT-ISE-NLP	62.23
75	NLP_UIowa	50.09
90	macech	7.33

Table A.1: Official rank and F_1 score (%) of the 36 participating teams that submitted system description papers. (There were 91 teams with submissions in total.) The median is shown in blue.

duplicate submissions and bypass any submission limits. More importantly, potentially interesting approaches are not discussed and properly compared to others.

It is also worth mentioning that the extended timeline allowed participants to join forces. For instance, a number of participants decided to combine their systems and form the 2^{nd} ranked S-NLP. Their ensemble scored higher than all their standalone systems, though their best standalone system would still be ranked 2^{nd}. In any case, we welcome the collaboration between participants, which may provide further insights regarding effective combinations of architectures.

A.4.2 General remarks on the approaches

Except for lexicon-based baselines, we observed that the vast majority of systems adopted the recent paradigm in NLP: fine-tuning large off-the-shelf transformers [515] pretrained on massive corpora. Non-transformer-based approaches, mostly LSTMs with pretrained word embeddings, were also used. The nature of the task, similar to the well-studied Named Entity Recognition task, led many competitors to use a CRF layer on top of the model (e.g., transformers or LSTMs) of their choice.

A.4.3 Performance

The winning team (HITSZ-HLT) combined BERT with two approaches for their ensemble: a token labeling approach (two versions, with/without an LSTM between BERT and the CRF) and a span extraction approach (Section A.2.1). The comparison of the two showed that span extraction is slightly better on posts with a single span, but token labeling is clearly better on multi-span posts [595]. The complementary nature of the two approaches is probably what makes even a simple majority voting ensemble better than its competitors.

The system ranked second (S-NLP) also employed an ensemble, using a RoBERTa model initially fine-tuned to classify posts as toxic or non-toxic as the starting point [361]. The ensemble combined (i) the resulting RoBERTa model, now fine-tuned to predict toxic spans, with additional FLAIR and FastText embeddings, and (ii) a RoBERTa model retrained as a student to predict toxic spans (Section A.2.2). Although the two standalone models achieved higher scores than the standalone models of the top-ranked team (HITSZ-HLT), the ensemble did not yield significant improvements. This may be due to the student's decisions not being that complementary to the teacher's, as the team notes [361].

Teams that experimented with rationale extraction mechanisms (Section A.2.3) did not find this approach advantageous compared to supervised sequence labeling in terms of F_1 scores. However, the reported results of the rationale-based systems show that this approach is promising, especially because it does not require any data annotated at the span level. Hence, we explored this direction in Section 4.7.2. Table A.2 shows the F_1 scores of all the rationale-based systems that were reported by participants. The binary toxic post classifiers that were

TBC	RE	$\mathbf{F_1}$ SCORE (%)	Report
LSTM	IE	38.29	Rusert [434]
LSTM	ATT	49.70	Pluciński and Klimczak [392]
LSTM	ATT	50.07	Rusert [434]
LR	LIME	58.88	Benlahbib et al. [31]
SVM	LIME	59.21	Benlahbib et al. [31]
BERT	SHAP	**59.87**	Pluciński and Klimczak [392]

Table A.2: F_1 on the evaluation set for systems employing rationale extraction (RE) mechanisms combined with post-level toxicity binary classifiers (TBC). Rationales are obtained via Input Erasure (IE), Attention (ATT), LIME, or SHAP. The binary classifier is an LSTM, Logistic Regression (LR), SVM, or BERT.

Lexicon Name	$\mathbf{F_1}$ SCORE (%)	Report
WIEGAND 1 †	33.07	Zhu et al. [595]
WORD-MATCH	40.86	Ranasinghe et al. [413]
FREQ-RATIO †	41.55	Rusert [434]
LOOKUP ‡	41.61	Burtenshaw and Kestemont [57]
WIEGAND 2 †	50.98	Zhu et al. [595]
ORTHRUS	61.07	Palomino et al. [369]
HITSZ-HLT ‡	**64.98**	Zhu et al. [595]
+WORDNET	64.09	Zhu et al. [595]
+GLOVE	64.19	Zhu et al. [595]

Table A.3: F_1 on the evaluation set for lexicon-based systems. Systems that are followed by † and ‡ use exclusively external and internal resources, respectively.

used were LSTM, Logistic Regression (LR), Support Vector Machines (SVM), and BERT. The attention scores of an LSTM were used with [392] and without an orthogonality method [434], with the latter being slightly better; these are model-specific rational extraction methods (Section A.2.3). Model-agnostic approaches (Input Erasure, LIME, SHAP) were better than the model-specific ones. The best rationale-based method employed a BERT model, fine-tuned for toxic post classification and SHAP.

Lexicon-based approaches were only used as baselines or components in ensembles, as already noted. In principle, all lexicon-based systems are extremely efficient and interpretable. Table A.3 shows they can also achieve surprisingly high scores.

A.4.4 Error analysis

A common theme across many competitor reports was the serious challenge posed by comments with no toxic spans. It is not readily evident why this is a common occurrence in the task, and certainly, the way that annotation consensus is used to combine annotations can be a contributing factor. However, many systems seemed determined to tag *some* spans and many authors noted that performance on posts with no tagged span was extremely poor compared to performance on posts with tagged spans.

Many systems were also reluctant to tag function words like ``of'' and ``and'', which can be included in multi-word

Type	Description
INCONSISTENCIES	Not all the occurrences of the same toxic span are annotated in the same post.
FALSE NEGATIVES	Toxic words missed.
FALSE POSITIVES	Non-toxic words labeled.

Table A.4: The types and descriptions of the annotation mistakes that were detected by some of the participants.

spans (e.g., ‘‘piece of crap’), leading to a decline in performance as measured by the chosen F_1 measure. The overwhelming presence of single-word gold spans in the training set favors short spans. But the majority of the short spans comprise common cuss, or clearly abusive words, which can be directly classified as toxic [146]; by contrast, the infrequent longer spans are rather context-dependent and more challenging to detect. This probably also contributed to the performance of the best system (HITSZ-HLT), since one of the two components of that ensemble handled better long spans, as already discussed in Section A.4.3.

Annotation mistakes reported are summarized in Table A.4.

Participants that were notable for their effort in error analysis include Bansal et al. [18], Hoang and Nguyen [191], Ding and Jurgens [110], and Ghosh and Kumar [146], where an additional effort was made to examine their model's ability to correctly tag words in toxic and non-toxic contexts. Interestingly Sans and Farràs [441] also noted in their analysis that racial and ethnic terms are labeled in biased ways that reflect patterns not only in the training toxic spans but also in external data used to pretrain underlying transformer models.

Bibliography

[1] A. Abid, M. F. Balin, and J. Zou. Concrete autoencoders for differentiable feature selection and reconstruction. *arXiv preprint arXiv:1901.09346*, 2019. 33

[2] D. Adiwardana et al. Towards a human-like open-domain chatbot. *CoRR*, abs/2001.09977, 2020. URL https://arxiv.org/abs/2001.09977. 29

[3] C. C. Aggarwal et al. *Recommender systems*, volume 1. Springer, 2016. 106

[4] A. V. Aho and J. D. Ullman. *The Theory of Parsing, Translation, and Compiling*. Prentice-Hall, Inc., USA, 1972. ISBN 0139145567. 19

[5] A. Akbik et al. Flair: An easy-to-use framework for state-of-the-art nlp. In *NAACL Demonstrations*, pages 54–59, 2019. 135

[6] G. Alain and Y. Bengio. What regularized auto-encoders learn from the data-generating distribution. *The Journal of Machine Learning Research*, 15(1):3563–3593, 2014. 33

[7] G. Alshammari et al. A switching multi-level method for the long tail recommendation problem. *Journal of Intelligent and Fuzzy Systems*, 37:7189–7198, 2019. ISSN 1064-1246. doi: 10.3233/JIFS-179331. URL https://uwe-repository.worktribe.com/output/847155. Comments and Suggestions : The final publication is available at IOS Press through http://dx.doi.org/10.3233/JIFS-179331. 108

[8] F. Alva-Manchego et al. ASSET: A dataset for tuning and evaluation of sentence simplification models with multiple rewriting transformations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4668–4679, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.424. URL https://aclanthology.org/2020.acl-main.424. 81

[9] R. Ando and T. Zhang. A high-performance semi-supervised learning method for text chunking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 1–9, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. doi: 10.3115/1219840.1219841. URL https://www.aclweb.org/anthology/P05-1001. 32

[10] A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2006. URL https://proceedings.neurips.cc/paper/2006/file/0afa92fc0f8a9cf051bf2961b06ac56b-Paper.pdf. 32

[11] M. Artetxe, G. Labaka, E. Agirre, and K. Cho. Unsupervised neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018*, 2018. 61, 80

[12] M. Aßenmacher and C. Heumann. On the comparability of pre-trained language models. *arXiv preprint arXiv:2001.00781*, 2020. 46

[13] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 54

[14] A. Baevski et al. Cloze-driven pretraining of self-attention networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5360–5369, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1539. URL https://aclanthology.org/D19-1539. 39

[15] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations*, 2014. 58, 62

[16] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*, Jan. 2015. Conference date: 07-05-2015 Through 09-05-2015. 53

[17] K. Balog, F. Radlinski, and S. Arakelyan. Transparent, scrutable and explainable user models for personalized recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 265–274, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450361729. doi: 10.1145/3331184.3331211. URL https://doi.org/10.1145/3331184.3331211. 108

[18] A. Bansal, A. Kaushik, and A. Modi. IITK@Detox at SemEval-2021 Task 5: Semi-supervised learning and dice loss for toxic spans detection. In *SemEval*, 2021. 141

[19] H. Bao et al. Unilmv2: Pseudo-masked language models for unified language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020. 23, 32, 33, 34, 42

[20] Y. Bao, H. Fang, and J. Zhang. Topicmf: Simultaneously exploiting ratings and reviews for recommendation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 28(1), Jun. 2014. URL https://ojs.aaai.org/index.php/AAAI/article/view/8715. 109

[21] R. Bar-Haim et al. The second pascal recognising textual entailment challenge. In *Proceedings of the second PASCAL challenges workshop on recognising textual entailment*, volume 6, pages 6–4. Venice, 2006. 22

[22] C. Baziotis, I. Androutsopoulos, I. Konstas, and A. Potamianos. SEQˆ3: Differentiable sequence-to-sequence-to-sequence autoencoder for unsupervised abstractive sentence compression. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 673–681, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1071. URL https://www.aclweb.org/anthology/N19-1071. 32

[23] C. Baziotis, I. Androutsopoulos, I. Konstas, and A. Potamianos. Seq3: Differentiable sequence-to-sequence-to-sequence autoencoder for unsupervised abstractive sentence compression. In *Proceedings of NAACL-HLT*, pages 673–681, 2019. 61

[24] R. Bellman. The theory of dynamic programming. *Bulletin of the American Mathematical Society*, 60(6): 503–515, 1954. 50

[25] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994. 52

[26] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3(null):1137–1155, mar 2003. ISSN 1532-4435. 23, 29, 35, 51

[27] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, 19, 2006. 50

[28] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013. URL http://arxiv.org/abs/1206.5538. cite arxiv:1206.5538. 28, 50

[29] Y. Bengio, L. Yao, G. Alain, and P. Vincent. Generalized denoising auto-encoders as generative models. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'13, page 899–907, Red Hook, NY, USA, 2013. Curran Associates Inc. 33

[30] R. Beniwal, R. Khairwal, R. Mahajan, and S. P. Narayan Singh. A comparison of similarity measures for neighbourhood based collaborative filtering recommender systems. In *2021 Asian Conference on Innovation in Technology (ASIANCON)*, pages 1–6, 2021. doi: 10.1109/ASIANCON51346.2021.9544901. 109

[31] A. Benlahbib, H. Alami, and A. Alami. LISAC FSDM USMBA at SemEval 2021 Task 5: Tackling toxic spans detection challenge with supervised spanBERT-based model and unsupervised LIME-based model. In *SemEval*, 2021. 136, 137, 140

[32] L. Bentivogli, P. Clark, I. Dagan, and D. Giampiccolo. The fifth pascal recognizing textual entailment challenge. In *TAC*, 2009. 22

[33] L. Bertinetto et al. Learning feed-forward one-shot learners. In D. Lee et al., editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/file/839ab46820b524afda05122893c2fe8e-Paper.pdf. 44

[34] M. Bhandari et al. Re-evaluating evaluation in text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9347–9359, Online, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.751. URL https://aclanthology.org/2020.emnlp-main.751. 81

[35] C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006. 28

[36] S. Black et al. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow, Mar. 2021. URL https://doi.org/10.5281/zenodo.5297715. 38

[37] A. Bogdanoff. Saying goodbye to civil comments, 12 2017. URL http://medium.com/@aja_15265/saying-goodbye-to-civil-comments-41859d3a2b1d. Accessed: 2021-04-15. 86

[38] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *TACL*, 5:135–146, 2017. ISSN 2307-387X. 135

[39] O. Bojar et al. Findings of the 2013 Workshop on Statistical Machine Translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 1–44, Sofia, Bulgaria, Aug. 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W13-2201. 22

[40] O. Bojar et al. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/W14-3302. URL https://www.aclweb.org/anthology/W14-3302. 22

[41] O. Bojar et al. Findings of the 2015 workshop on statistical machine translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46, Lisbon, Portugal, Sept. 2015. Association for Computational Linguistics. doi: 10.18653/v1/W15-3001. URL https://www.aclweb.org/anthology/W15-3001. 22

[42] O. Bojar et al. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 131–198, Berlin, Germany, Aug. 2016. Association for Computational Linguistics. doi: 10.18653/v1/W16-2301. URL https://www.aclweb.org/anthology/W16-2301. 22

[43] O. Bojar et al. Findings of the 2017 conference on machine translation (WMT17). In *Proceedings of the Second Conference on Machine Translation*, pages 169–214, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4717. URL https://www.aclweb.org/anthology/W17-4717. 22

[44] O. Bojar et al. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 272–303, Belgium, Brussels, Oct. 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6401. URL https://www.aclweb.org/anthology/W18-6401. 22

[45] L. Boltzmann and F. Hasenöhrl. Studien über das gleichgewicht der lebendigen kraft zwischen bewegten materiellen punkten, 2012. 47

[46] R. Bommasani et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021. 46

[47] E. V. Bonilla, K. Chai, and C. Williams. Multi-task gaussian process prediction. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007. URL https://proceedings.neurips.cc/paper/2007/file/66368270ffd51418ec58bd793f2d9b1b-Paper.pdf. 32

[48] D. Borkan et al. Nuanced metrics for measuring unintended bias with real data for text classification. In *WWW*, pages 491–500, San Francisco, USA, 2019. 84, 86, 95, 101, 136

[49] D. Borkan et al. Nuanced metrics for measuring unintended bias with real data for text classification. *CoRR*, abs/1903.04561, 2019. URL http://arxiv.org/abs/1903.04561. 58, 66

[50] D. Borkan, J. Sorensen, and L. Vasserman. Exploring the role of human raters in creating nlp datasets, 11 2019. URL http://medium.com/jigsaw/creating-labeled-datasets-and-exploring-the-role-of-human-raters-56367b6db298. Accessed: 2021-04-15. 86

[51] H. Bourlard and Y. Kamp. Auto-association by multilayer perceptrons and singular value decomposition. *Biological cybernetics*, 59(4):291–294, 1988. 33

[52] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, Sept. 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1075. URL https://www.aclweb.org/anthology/D15-1075. 111

[53] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*, 2015. 22

[54] S. Bozinovski. Reminder of the first paper on transfer learning in neural networks, 1976. *Informatica*, 44(3), 2020. 32

[55] P. F. Brown et al. An estimate of an upper bound for the entropy of english. *Comput. Linguist.*, 18(1):31–40, mar 1992. ISSN 0891-2017. 20

[56] T. Brown et al. Language models are few-shot learners. In H. Larochelle et al., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf`. 23, 29, 31, 32, 34, 38, 43, 44, 45, 46

[57] B. Burtenshaw and M. Kestemont. UAntwerp at SemEval-2021 Task 5: Spans are spans, stacking a binary word level approach to toxic span detection. In *SemEval*, 2021. 136, 140

[58] S. Buttcher, C. L. Clarke, and G. V. Cormack. *Information retrieval: Implementing and evaluating search engines*. Mit Press, 2016. 109

[59] N. Carmeli et al. Constructing explainable opinion graphs from reviews. In *Proceedings of the Web Conference 2021*, WWW '21, page 3419–3431, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383127. doi: 10.1145/3442381.3450081. URL `https://doi.org/10.1145/3442381.3450081`. 26

[60] R. Caruana. Multitask learning. *Machine Learning*, 28:41–75, 2004. 45

[61] A. Cauchy et al. Méthode générale pour la résolution des systemes d'équations simultanées. *Comp. Rend. Sci. Paris*, 25(1847):536–538, 1847. 50

[62] D. Cer et al. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*, 2017. 22

[63] D. Cer et al. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*, 2018. 23, 51, 67, 80

[64] D. Cer et al. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium, Nov. 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-2029. URL `https://www.aclweb.org/anthology/D18-2029`. 30

[65] C. F. Chabris and D. J. Simons. *The invisible gorilla: And other ways our intuitions deceive us*. Harmony, 2010. 53

[66] P. Chandar, F. Diaz, and B. S. Thomas. Beyond accuracy: Grounding evaluation metrics for human-machine learning systems. *Advances in Neural Information Processing Systems*, 2020. 109

[67] M.-W. Chang, L.-A. Ratinov, D. Roth, and V. Srikumar. Importance of semantic representation: Dataless classification. In *Aaai*, volume 2, pages 830–835, 2008. 44

[68] E. Charniak. Statistical techniques for natural language parsing. *AI Magazine*, 18(4):33–44, 1997. URL `http://dblp.uni-trier.de/db/journals/aim/aim18.html#Charniak97`. 19

[69] C. Chen, M. Zhang, Y. Liu, and S. Ma. Neural attentional rating regression with review-level explanations. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, page 1583–1592, Republic and Canton of Geneva, CHE, 2018. International World Wide Web Conferences Steering Committee. ISBN 9781450356398. doi: 10.1145/3178876.3186070. URL `https://doi.org/10.1145/3178876.3186070`. 109, 117

[70] J. Chen et al. Visualgpt: Data-efficient image captioning by balancing visual input and linguistic knowledge from pretraining. *CoRR*, abs/2102.10407, 2021. URL `https://arxiv.org/abs/2102.10407`. 20

[71] L. Chen and F. Wang. Preference-based clustering reviews for augmenting e-commerce recommendation. *Know.-Based Syst.*, 50(C):44–59, sep 2013. ISSN 0950-7051. 106

[72] L. Chen, G. Chen, and F. Wang. Recommender systems based on user reviews: The state of the art. *User Modeling and User-Adapted Interaction*, 25(2):99–154, jun 2015. ISSN 0924-1868. doi: 10.1007/s11257-015-9155-5. URL `https://doi.org/10.1007/s11257-015-9155-5`. 106, 108

[73] M. Chen et al. Generative pretraining from pixels. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1691–1703. PMLR, 13–18 Jul 2020. URL `https://proceedings.mlr.press/v119/chen20s.html`. 46

[74] J. Y. Chin, K. Zhao, S. Joty, and G. Cong. Anr: Aspect-based neural recommender. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, page 147–156, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450360142. doi: 10.1145/3269206.3271810. URL `https://doi.org/10.1145/3269206.3271810`. 109

[75] K. Cho et al. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, Oct. 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1179. URL `https://www.aclweb.org/anthology/D14-1179`. 52

[76] E. Choi et al. QuAC: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184, Brussels, Belgium, Oct.-Nov. 2018. Associa-

tion for Computational Linguistics. doi: 10.18653/v1/D18-1241. URL https://www.aclweb.org/anthology/D18-1241. 22

[77] N. Chomsky. *Syntactic Structures*. Mouton and Co., The Hague, 1957. 19

[78] N. Chomsky. Three factors in language design. *LINGUISTIC INQUIRY*, 36(1):1–22, 2005. 39

[79] A. Chowdhery et al. Palm: Scaling language modeling with pathways, 2022. URL https://arxiv.org/abs/2204.02311. 23, 29, 31, 34, 39, 44, 45

[80] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014. URL https://arxiv.org/abs/1412.3555. 23, 52

[81] C. Clark et al. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1300. URL https://www.aclweb.org/anthology/N19-1300. 22

[82] K. Clark, U. Khandelwal, O. Levy, and C. D. Manning. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy, Aug. 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4828. URL https://www.aclweb.org/anthology/W19-4828. 50

[83] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning. Pre-training transformers as energy-based cloze models. In *EMNLP*, 2020. URL https://www.aclweb.org/anthology/2020.emnlp-main.20.pdf. 23, 31, 33, 34

[84] C. G. Coll, E. L. Bearer, and R. M. Lerner. *Nature and nurture: The complex interplay of genetic and environmental influences on human behavior and development*. Psychology press, 2014. 43

[85] R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 160–167, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605582054. doi: 10.1145/1390156.1390177. URL https://doi.org/10.1145/1390156.1390177. 30

[86] R. Collobert et al. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12(null): 2493–2537, nov 2011. ISSN 1532-4435. 30

[87] R. Collobert et al. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(ARTICLE):2493–2537, 2011. 50

[88] A. Conneau and G. Lample. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, pages 7057–7067, 2019. 20, 33, 61, 62, 63, 67

[89] A. Conneau et al. Word translation without parallel data. In *International Conference on Learning Representations (ICLR)*, 2018. 61

[90] T. M. Cover. *Elements of information theory*. John Wiley & Sons, 1999. 28

[91] P. Cremonesi, Y. Koren, and R. Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, page 39–46, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781605589060. doi: 10.1145/1864708.1864721. URL https://doi.org/10.1145/1864708.1864721. 108, 109, 115

[92] G. Da San Martino et al. Fine-grained analysis of propaganda in news article. In *EMNLP-IJCNLP*, pages 5640–5650, 2019. 90

[93] R. Dabre, C. Chu, and A. Kunchukuttan. A survey of multilingual neural machine translation. *ACM Comput. Surv.*, 53(5), sep 2020. ISSN 0360-0300. doi: 10.1145/3406095. URL https://doi.org/10.1145/3406095. 20

[94] I. Dagan, O. Glickman, and B. Magnini. The pascal recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, pages 177–190. Springer, 2005. 22

[95] A. M. Dai and Q. V. Le. Semi-supervised sequence learning. In C. Cortes et al., editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper/2015/file/7137debd45ae4d0ab9aa953017286b20-Paper.pdf. 34

[96] N. Dai, J. Liang, X. Qiu, and X. Huang. Style transformer: Unpaired text style transfer without disentangled latent representation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5997–6007, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1601. URL https://www.aclweb.org/anthology/P19-1601. 62, 63, 66, 68, 72, 73

[97] W. Dai, G.-R. Xue, Q. Yang, and Y. Yu. Co-clustering based classification for out-of-domain documents. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, page 210–219, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595936097. doi: 10.1145/1281192.1281218. URL https://doi.org/10.1145/1281192.1281218. 32

[98] Z. Dai et al. Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, 2019. 61

[99] S. Dathathri et al. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=H1edEyBKDS`. 62

[100] A. Dau, N. Salim, R. Idris, and A. Osman. Weighted aspect-based opinion mining using deep learning for recommender system. *Expert Systems with Applications*, 140:112871, 08 2019. doi: 10.1016/j.eswa.2019.112871. 109

[101] T. Davidson, D. Warmsley, M. Macy, and I. Weber. Automated hate speech detection and the problem of offensive language. In *Eleventh international aaai conference on web and social media*, pages 512–515, 2017. 58

[102] M.-C. De Marneff, M. Simons, and J. Tonhauser. The commitmentbank: Investigating projection in naturally occurring discourse. *proceedings of Sinn und Bedeutung 23*, 2019. 22

[103] F. de Saussure and W. Baskin. *Course in General Linguistics: Translated by Wade Baskin. Edited by Perry Meisel and Haun Saussy*. Columbia University Press, 2011. URL `http://www.jstor.org/stable/10.7312/saus15726`. 39

[104] D. Dementieva et al. Russe-2022: Findings of the first russian detoxification shared task based on parallel corpora. 81

[105] D. Dementieva et al. Crowdsourcing of parallel corpora: the case of style transfer for detoxification. In *Proceedings of the 2nd Crowd Science Workshop: Trust, Ethics, and Excellence in Crowdsourced Data Management at Scale co-located with 47th International Conference on Very Large Data Bases (VLDB 2021 (https://vldb.org/2021/))*, pages 35–49, Copenhagen, Denmark, 2021. CEUR Workshop Proceedings. URL `http://ceur-ws.org/Vol-2932/paper2.pdf`. 81, 84, 97

[106] J.-L. Dessalles. *Des intelligences très artificielles*. Odile Jacob, 2019. 26

[107] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186, Minneapolis, Minnesota, 2019. 23, 91, 135

[108] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL `https://www.aclweb.org/anthology/N19-1423`. 23, 29, 31, 32, 33, 34, 40, 44, 45, 46, 63, 66, 106

[109] J. DeYoung et al. ERASER: A benchmark to evaluate rationalized NLP models. In *ACL*, pages 4443–4458, Online, 2020. doi: 10.18653/v1/2020.acl-main.408. URL https://aclanthology.org/2020.acl-main.408. 84, 91

[110] H. Ding and D. Jurgens. HamiltonDinggg at SemEval-2021 Task 5: Investigating toxic span detection using RoBERTa pre-training. In *SemEval*, 2021. 136, 137, 141

[111] W. B. Dolan and C. Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, 2005. 22

[112] L. Dong et al. Unified language model pre-training for natural language understanding and generation. *Advances in Neural Information Processing Systems*, 32, 2019. 23, 32, 33, 34, 38, 42

[113] A. G. D'Sa, I. Illina, and D. Fohr. Towards non-toxic landscapes: Automatic toxic comment detection using DNN. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 21–25, Marseille, France, 2020. European Language Resources Association (ELRA). ISBN 979-10-95546-56-6. URL https://aclanthology.org/2020.trac-1.4. 91

[114] Z. Du et al. All NLP tasks are generation tasks: A general pretraining framework. *CoRR*, abs/2103.10360, 2021. URL https://arxiv.org/abs/2103.10360. 20, 23, 32, 34, 42

[115] N. Durrani, B. Haddow, P. Koehn, and K. Heafield. Edinburgh's phrase-based machine translation systems for WMT-14. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 97–104, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/W14-3309. URL https://www.aclweb.org/anthology/W14-3309. 22

[116] S. Edunov, M. Ott, M. Auli, and D. Grangier. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, 2018. 61, 63

[117] W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed. Automatic text summarization: A comprehensive survey. *Expert Syst. Appl.*, 165:113679, 2021. 20

[118] J. L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990. ISSN 0364-0213. doi: https://doi.org/10.1016/0364-0213(90)90002-E. URL https://www.sciencedirect.com/science/article/pii/036402139090002E. 52

[119] C. Emmery, E. Manjavacas Arevalo, and G. Chrupała. Style obfuscation by invariance. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 984–996, Santa Fe, New Mexico, USA, Aug. 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/C18-1084. 61

[120] D. Erhan et al. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11(19):625–660, 2010. URL http://jmlr.org/papers/v11/erhan10a.html. 34

[121] T. Evgeniou and M. Pontil. Regularized multi–task learning. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, page 109–117, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 1581138881. doi: 10.1145/1014052.1014067. URL https://doi.org/10.1145/1014052.1014067. 32

[122] A. R. Fabbri et al. SummEval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409, 2021. doi: 10.1162/tacl_a_00373. URL https://aclanthology.org/2021.tacl-1.24. 81

[123] M. Fadaee, A. Bisazza, and C. Monz. Data augmentation for low-resource neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, 2017. doi: 10.18653/v1/p17-2090. URL https://doi.org/10.18653%2Fv1%2Fp17-2090. 26

[124] Z. Fan, S. Zhou, and B. Xu. Unsupervised pre-traing for sequence to sequence speech recognition. *CoRR*, abs/1910.12418, 2019. URL http://arxiv.org/abs/1910.12418. 19

[125] W. Fedus, B. Zoph, and N. Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity, 2021. URL https://arxiv.org/abs/2101.03961. 23, 32, 34, 45

[126] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):594–611, 2006. doi: 10.1109/TPAMI.2006.79. 43

[127] J. Ferrando and M. R. Costa-jussà. Attention weights in transformer NMT fail aligning words between sequences but largely explain model predictions. In *Findings of EMNLP*, pages 434–443, Punta Cana, Dominican Republic, 2021. URL https://aclanthology.org/2021.findings-emnlp.39. 91

[128] J. Ficler and Y. Goldberg. Controlling linguistic style aspects in neural language generation. In *Proceedings of the Workshop on Stylistic Variation*, pages 94–104, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4912. URL https://www.aclweb.org/anthology/W17-4912. 62

[129] K. Finley. Want to save the comments from trolls? do it yourself, 3 2016. URL http://www.wired.com/2016/03/want-save-comments-trolls/. Accessed: 2021-04-15. 86

[130] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/finn17a.html. 44

[131] R. Fisher and K. Pearson. On an absolute criterion for fitting frequency curves, 1911. URL https://books.google.fr/books?id=dXXzjgEACAAJ. 28

[132] M. Fitzgerald, A. Boddy, and S. D. Baum. 2020 survey of artificial general intelligence projects for ethics, risk, and policy, 2020. 44

[133] E. Forgy. Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *Biometrics*, 21:768–780, 1965. 27

[134] P. Fortuna and S. Nunes. A survey on automatic detection of hate speech in text. *ACM Computing Surveys (CSUR)*, 51(4):1–30, 2018. 58

[135] K. P. F.R.S. Liii. on lines and planes of closest fit to systems of points in space. *Philosophical Magazine Series 1*, 2:559–572, 1901. 27

[136] Z. Fu et al. Style transfer in text: Exploration and evaluation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 61, 67, 68, 73, 86

[137] K. Fukushima and S. Miyake. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*, pages 267–285. Springer, 1982. 51

[138] S. Funk. Netflix update: Try this at home. https://sifter.org/simon/journal/20061211.html, 2006. URL https://sifter.org/simon/journal/20061211.html. 108

[139] B. Gambäck and U. K. Sikdar. Using convolutional neural networks to classify hate-speech. In *Proceedings of the first workshop on abusive language online*, pages 85–90, 2017. 58

[140] Z. Gan et al. Semantic compositional networks for visual captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5630–5639, 2017. 61

[141] J. Gao, W. Fan, J. Jiang, and J. Han. Knowledge transfer via multiple model local structure mapping. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’08, page 283–291, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605581934. doi: 10.1145/1401890.1401928. URL https://doi.org/10.1145/1401890.1401928. 32

[142] T. Gao, A. Fisch, and D. Chen. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online, Aug. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.295. URL https://aclanthology.org/2021.acl-long.295. 45

[143] L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015. 59

[144] A. Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. " O'Reilly Media, Inc.", 2019. 33

[145] F. A. Gers, J. Schmidhuber, and F. Cummins. Learning to Forget: Continual Prediction with LSTM. *Neural Computation*, 12(10):2451–2471, 10 2000. ISSN 0899-7667. doi: 10.1162/089976600300015015. URL https://doi.org/10.1162/089976600300015015. 52

[146] S. Ghosh and S. Kumar. Cisco at SemEval-2021 Task 5: What's toxic?: Leveraging transformers for multiple toxic span extraction from online comments. In *SemEval*, 2021. 141

[147] D. Giampiccolo, B. Magnini, I. Dagan, and B. Dolan. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9, Prague, June 2007. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W07-1401. 22

[148] X. Glorot, A. Bordes, and Y. Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, page 513–520, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195. 33

[149] Y. Goldberg. *Neural Network Methods for Natural Language Processing*, volume 37 of *Synthesis Lectures on Human Language Technologies*. Morgan & Claypool, San Rafael, CA, 2017. ISBN 978-1-62705-298-6. doi: 10.2200/S00762ED1V01Y201703HLT037. 46

[150] D. Goldhaber. *The nature-nurture debates: Bridging the gap*. Cambridge University Press, 2012. 43

[151] H. Gong et al. Reinforcement learning based text style transfer without parallel training corpus. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3168–3180, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1320. URL https://www.aclweb.org/anthology/N19-1320. 24

[152] H. Gong et al. Reinforcement learning based text style transfer without parallel training corpus. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3168–3180, 2019. 61, 67

[153] I. Goodfellow et al. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 59

[154] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. URL http://www.deeplearningbook.org. 33, 34, 40, 51

[155] J. T. Goodman. A bit of progress in language modeling. *Computer Speech & Language*, 15(4):403–434, 2001. 35

[156] T. Goucha and A. D. Friederici. The language skeleton after dissecting meaning: A functional segregation within broca's area. *Neuroimage*, 114:294–302, 2015. 51

[157] E. Grave et al. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018. 23, 30

[158] A. Graves. *Supervised Sequence Labelling with Recurrent Neural Networks*. Studies in computational intelligence. Springer, Berlin, 2012. doi: 10.1007/978-3-642-24797-2. URL https://cds.cern.ch/record/1503877. 52

[159] A. Graves. Generating sequences with recurrent neural networks. *CoRR*, abs/1308.0850, 2013. URL http://arxiv.org/abs/1308.0850. 52, 53

[160] A. Graves, G. Wayne, and I. Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014. 52

[161] Y. Gu, X. Han, Z. Liu, and M. Huang. Ppt: Pre-trained prompt tuning for few-shot learning, 2021. URL https://arxiv.org/abs/2109.04332. 45

[162] R. Guo et al. Accelerating large-scale inference with anisotropic vector quantization. In *International Conference on Machine Learning*, 2020. URL https://arxiv.org/abs/1908.10396. 80, 127

[163] A. Gupta, G. Boleda, M. Baroni, and S. Padó. Distributional vectors encode referential attributes. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 12–21, Lisbon, Portugal, Sept. 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1002. URL https://www.aclweb.org/anthology/D15-1002. 30

[164] W. Haas. J.r. firth: Papers in linguistics, 1934–1951. xii, 233 pp., 11 plates. london, etc.: Oxford university press, 1957. 35s. *Bulletin of the School of Oriental and African Studies*, 21(3):668–671, 1958. doi: 10.1017/S0041977X00060559. 30

[165] K. Hambardzumyan, H. Khachatrian, and J. May. WARP: Word-level Adversarial ReProgramming. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4921–4933, Online, Aug. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.381. URL https://aclanthology.org/2021.acl-long.381. 45

[166] B.-O. Han. On language peculiarities: when language evolves that much that speakers find it strange. *Studia Universitatis Petru Maior. Philologia*, (18):138, 2015. 18

[167] H. Han, M. Huang, Y. Zhang, and U. A. Bhatti. An extended-tag-induced matrix factorization technique for recommender systems. *Information*, 9(6), 2018. ISSN 2078-2489. doi: 10.3390/info9060143. URL https://www.mdpi.com/2078-2489/9/6/143. 108

[168] J. Han and C. Moraga. The influence of the sigmoid function parameters on the speed of backpropagation learning. In *Proceedings of the International Workshop on Artificial Neural Networks: From Natural to Artificial Neural Computation*, IWANN '96, page 195–201, Berlin, Heidelberg, 1995. Springer-Verlag. ISBN 3540594973. 47

[169] X. Han and Y. Tsvetkov. Fortifying toxic speech detectors against veiled toxicity. In *EMNLP*, pages 7732–7739, Online, 2020. 86

[170] X. Han et al. Pre-trained models: Past, present and future. *AI Open*, 2:225–250, 2021. ISSN 2666-6510. doi: https://doi.org/10.1016/j.aiopen.2021.08.002. URL https://www.sciencedirect.com/science/article/pii/S2666651021000231. 20, 21

[171] P. Hayes-Roth et al. Speech understanding systems: Summary of results of the five-year research effort, 1976. 36

[172] J. He, X. Wang, G. Neubig, and T. Berg-Kirkpatrick. A probabilistic formulation of unsupervised text style transfer. *arXiv preprint arXiv:2002.03912*, 2020. 66

[173] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 54

[174] P. He, J. Gao, and W. Chen. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing, 2021. 23, 31, 33, 34, 40, 46

[175] R. He and J. McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, page 507–517, Republic and Canton of Geneva, CHE, 2016. International World Wide Web Conferences Steering Committee. ISBN 9781450341431. doi: 10.1145/2872427.2883037. URL https://doi.org/10.1145/2872427.2883037. 26, 106, 110

[176] R. He and J. McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, pages 507–517, 2016. 61

[177] X. He et al. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182, 2017. 110

[178] D. Hendrycks and K. Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 47

[179] S. Herculano-Houzel. The human brain in numbers: a linearly scaled-up primate brain. *Frontiers in Human Neuroscience*, 3, 2009. ISSN 1662-5161. doi: 10.3389/neuro.09.031.2009. URL https://www.frontiersin.org/article/10.3389/neuro.09.031.2009. 46

[180] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, jan 2004. ISSN 1046-8188. doi: 10.1145/963770.963772. URL https://doi.org/10.1145/963770.963772. 108, 109

[181] K. M. Hermann et al. Teaching machines to read and comprehend. In C. Cortes et al., editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper/2015/file/afdec7005cc9f14302cd0474fd0f3c96-Paper.pdf. 22

[182] M. Hernández-Rubio, I. Cantador, and A. Bellogín. A comparative analysis of recommender systems based on item aspect opinions extracted from user reviews. *User Modeling and User-Adapted Interaction*, 29, 04 2019. doi: 10.1007/s11257-018-9214-9. 108

[183] F. Hill, A. Bordes, S. Chopra, and J. Weston. The goldilocks principle: Reading children's books with explicit memory representations, 2015. URL https://arxiv.org/abs/1511.02301. 22

[184] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313 (5786):504–507, 2006. doi: 10.1126/science.1127647. URL https://www.science.org/doi/abs/10.1126/science.1127647. 34

[185] G. E. Hinton and R. Zemel. Autoencoders, minimum description length and helmholtz free energy. In J. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems*, volume 6. Morgan-Kaufmann, 1993. URL https://proceedings.neurips.cc/paper/1993/file/9e3cfc48eccf81a0d57663e129aef3cb-Paper.pdf. 33

[186] G. E. Hinton, J. L. McClelland, and D. E. Rumelhart. *Distributed Representations*, page 77–109. MIT Press, Cambridge, MA, USA, 1986. ISBN 026268053X. 29

[187] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006. 50

[188] G. E. Hinton, A. Krizhevsky, and S. D. Wang. Transforming auto-encoders. In *International conference on artificial neural networks*, pages 44–51. Springer, 2011. 33

[189] G. E. Hinton et al. Learning distributed representations of concepts. In *Proceedings of the eighth annual conference of the cognitive science society*, volume 1, page 12. Amherst, MA, 1986. 50

[190] H. O. Hirschfeld. A connection between correlation and contingency. *Mathematical Proceedings of the Cambridge Philosophical Society*, 31(4):520–524, 1935. doi: 10.1017/S0305004100013517. 112

[191] P. G. Hoang and L. T. Nguyen. UIT-E10dot3 at SemEval 2021 Task 5: Toxic spans detection with roberta and spacy's library base systems. In *SemEval*, 2021. 141

[192] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, nov 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL `https://doi.org/10.1162/neco.1997.9.8.1735`. 52

[193] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. 135

[194] M. Hodosh, P. Young, and J. Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47:853–899, 2013. 81

[195] J. Hoffmann et al. Training compute-optimal large language models, 2022. URL `https://arxiv.org/abs/2203.15556`. 23, 31, 34, 39, 45

[196] I. Holloway. Simultaneity as a unique property of visual-spatial language: the simultaneous structure of two-handed classifier predicates in bimodal asl/english narrative ebooks for deaf children*, 2017. 39

[197] M. Honnibal and I. Montani. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 2017. 91

[198] H. Hosseini, S. Kannan, B. Zhang, and R. Poovendran. Deceiving google's perspective api built for detecting toxic comments. In *arXiv preprint*, 2017. 86

[199] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933. 27

[200] J. Howard and S. Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1031. URL `https://www.aclweb.org/anthology/P18-1031`. 23, 34

[201] M. Hu et al. Open-domain targeted sentiment analysis via span-based extraction and classification. In *ACL*, pages 537–546, 2019. 135

[202] Z. Hu et al. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1587–1596. JMLR. org, 2017. 61

[203] M. Huang, X. Zhu, and J. Gao. Challenges in building intelligent open-domain dialog systems. *ACM Trans. Inf. Syst.*, 38(3), apr 2020. ISSN 1046-8188. doi: 10.1145/3383123. URL https://doi.org/10.1145/3383123. 20

[204] Z. Huang, W. Xu, and K. Yu. Bidirectional lstm-crf models for sequence tagging. *ArXiv*, abs/1508.01991, 2015. 19

[205] D. H. Hubel and T. N. Wiesel. Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, 195(1):215–243, 1968. 51

[206] N. Hug. Surprise, a Python library for recommender systems. http://surpriselib.com, 2017. 116, 117

[207] S. Iyer, N. Dandekar, and K. Csernai. First quora dataset release: Question pairs, 2017. URL https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs. 22

[208] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Reading text in the wild with convolutional neural networks. *International Journal of Computer Vision*, 116:1–20, 2015. 51

[209] S. Jain, S. Wiegreffe, Y. Pinter, and B. C. Wallace. Learning to faithfully rationalize by construction. In *ACL*, pages 4459–4473, Online, 2020. doi: 10.18653/v1/2020.acl-main.409. URL https://aclanthology.org/2020.acl-main.409. 84, 91

[210] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, oct 2002. ISSN 1046-8188. doi: 10.1145/582415.582418. URL https://doi.org/10.1145/582415.582418. 108

[211] G. Jawahar, B. Sagot, and D. Seddah. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1356. URL https://www.aclweb.org/anthology/P19-1356. 50

[212] F. Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, Cambridge, MA, USA, 1998. ISBN 0262100665. 35

[213] J. J. Jiang and D. W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the 10th Research on Computational Linguistics International Conference*, pages 19–33, Taipei, Taiwan, Aug. 1997. The Association for Computational Linguistics and Chinese Language Processing (ACLCLP). URL https://www.aclweb.org/anthology/O97-1002. 109

[214] Z. Jiang, F. F. Xu, J. Araki, and G. Neubig. How Can We Know What Language Models Know? *Transactions of the Association for Computational Linguistics*, 8:423–438, 07 2020. ISSN 2307-387X. doi: 10.1162/tacl_a_00324. URL https://doi.org/10.1162/tacl_a_00324. 45

[215] Z. Jin et al. Imat: Unsupervised text attribute transfer via iterative matching and translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3088–3100, 2019. 68

[216] V. John, L. Mou, H. Bahuleyan, and O. Vechtomova. Disentangled representation learning for non-parallel text style transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 424–434, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1041. URL https://www.aclweb.org/anthology/P19-1041. 61, 67, 68

[217] J. Johnson, M. Douze, and H. Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547, 2021. doi: 10.1109/TBDATA.2019.2921572. 80

[218] I. T. Jolliffe. *Principal component analysis for special types of data*. Springer, 2002. 27

[219] M. Joshi et al. SpanBERT: Improving pre-training by representing and predicting spans. *TACL*, 8:64–77, 2020. 91, 92, 103

[220] R. Jozefowicz et al. Exploring the limits of language modeling, 2016. URL https://arxiv.org/pdf/1602.02410.pdf. 29, 35

[221] D. Jurafsky and J. H. Martin. Speech and language processing (3rd (draft) ed.), 2019. 38

[222] Kaggle. Yelp dataset, Mar. 2021. URL https://www.kaggle.com/yelp-dataset/yelp-dataset. 26, 106, 110

[223] L. Kaiser, O. Nachum, A. Roy, and S. Bengio. Learning to remember rare events. *CoRR*, abs/1703.03129, 2017. URL http://arxiv.org/abs/1703.03129. 44

[224] N. Kalchbrenner, E. Grefenstette, and P. Blunsom. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665, Baltimore, Maryland, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/P14-1062. URL https://www.aclweb.org/anthology/P14-1062. 51

[225] K. S. Kalyan, A. Rajasekharan, and S. Sangeetha. Ammus: A survey of transformer-based pretrained models in natural language processing. *arXiv preprint arXiv:2108.05542*, 2021. 21

[226] E. R. Kandel et al. *Principles of neural science*, volume 4. McGraw-hill New York, 2000. 47

[227] H. Kane et al. NUBIA: NeUral based interchangeability assessor for text generation. In *Proceedings of the 1st Workshop on Evaluating NLG Evaluation*, pages 28–37, Online (Dublin, Ireland), Dec. 2020. Association for Computational Linguistics. URL `https://aclanthology.org/2020.evalnlgeval-1.4`. 20, 82

[228] J. Kaplan et al. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 45

[229] M. Karan and J. Šnajder. Preemptive toxic language detection in Wikipedia comments using thread-level context. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 129–134, Florence, Italy, 2019. 86

[230] E. F. Keller. The mirage of a space between nature and nurture. In *The Mirage of a Space between Nature and Nurture*. Duke University Press, 2010. 43

[231] M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1-2):81–93, 06 1938. ISSN 0006-3444. doi: 10.1093/biomet/30.1-2.81. URL `https://doi.org/10.1093/biomet/30.1-2.81`. 116

[232] N. S. Keskar et al. CTRL - A Conditional Transformer Language Model for Controllable Generation. *arXiv preprint arXiv:1909.05858*, 2019. 20, 43, 62, 63

[233] D. Khashabi et al. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 252–262, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1023. URL `https://www.aclweb.org/anthology/N18-1023`. 22

[234] C. W. ki Leung, S. C. fai Chan, and K. F.-L. Chung. Integrating collaborative filtering and sentiment analysis: A rating inference approach. 2006. 109

[235] D. Kim et al. Convolutional matrix factorization for document context-aware recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, page 233–240, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450340359. doi: 10.1145/2959100.2959165. URL `https://doi.org/10.1145/2959100.2959165`. 109

[236] Y. Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, Oct. 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1181. URL `https://www.aclweb.org/anthology/D14-1181`. 51

[237] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 33

[238] D. P. Kingma and M. Welling. An introduction to variational autoencoders. *arXiv preprint arXiv:1906.02691*, 2019. 33

[239] R. Kiros et al. Skip-thought vectors. In C. Cortes et al., editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper/2015/file/f442d33fa06832082290ad8544a8da27-Paper.pdf. 29

[240] G. Kobayashi, T. Kuribayashi, S. Yokoi, and K. Inui. Attention is not only a weight: Analyzing transformers with vector norms. In *EMNLP*, pages 7057–7075, Online, 2020. doi: 10.18653/v1/2020.emnlp-main.574. URL https://aclanthology.org/2020.emnlp-main.574. 91

[241] P. Koehn. *Statistical Machine Translation*. Cambridge University Press, 2009. doi: 10.1017/CBO9780511815829. 20

[242] Y. Koren. Factor in the neighbors: Scalable and accurate collaborative filtering. *ACM Trans. Knowl. Discov. Data*, 4(1), jan 2010. ISSN 1556-4681. doi: 10.1145/1644873.1644874. URL https://doi.org/10.1145/1644873.1644874. 108, 113

[243] Y. Koren and J. Sill. Collaborative filtering on ordinal user feedback. In *IJCAI*, 2013. 108, 109, 116

[244] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42 (8):30–37, 2009. doi: 10.1109/MC.2009.263. 108

[245] S. B. Kotsiantis. Supervised machine learning: A review of classification techniques. In *Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in EHealth, HCI, Information Retrieval and Pervasive Technologies*, page 3–24, NLD, 2007. IOS Press. ISBN 9781586037802. 24

[246] G. Koutrika, B. Bercovitz, and H. Garcia-Molina. Flexrecs: Expressing and combining flexible recommendations. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data*, SIGMOD '09, page 745–758, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605585512. doi: 10.1145/1559845.1559923. URL https://doi.org/10.1145/1559845.1559923. 108

[247] M. A. Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, 37(2):233–243, 1991. 33

[248] S. Kramer. *History Begins at Sumer: Thirty-Nine Firsts in Recorded History*. University of Pennsylvania Press, Incorporated, 1981. ISBN 9780812212761. URL https://books.google.fr/books?id=RkUFwAEACAAJ. 13

[249] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing*

Systems, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf. 14

[250] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 50

[251] T. Kudo and J. Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium, Nov. 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-2012. URL https://www.aclweb.org/anthology/D18-2012. 29, 71, 97, 111

[252] S. Kullback and R. A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1): 79 – 86, 1951. doi: 10.1214/aoms/1177729694. URL https://doi.org/10.1214/aoms/1177729694. 28

[253] S. Kumar and P. Talukdar. Reordering examples helps during priming-based few-shot learning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4507–4518, Online, Aug. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.395. URL https://aclanthology.org/2021.findings-acl.395. 44

[254] S. Kumar and Y. Tsvetkov. Von mises-fisher loss for training sequence to sequence models with continuous outputs. *arXiv preprint arXiv:1812.04616*, 2018. 61

[255] S. Kumar and Y. Tsvetkov. Von mises-fisher loss for training sequence to sequence models with continuous outputs. In *Proc. of ICLR*, 2019. URL https://arxiv.org/pdf/1812.04616.pdf. 82

[256] R. Kurzweil. How to create a mind: The secret of human thought revealed, viking, 2012. 49

[257] B. Kuyumcu, S. Delil, and C. aksakallı. Sefamerve_arge at SemEval-2021 Task 5: Toxic span detection using segmentation based 1-d convolutional neural network model. In *SemEval*, 2021. 137

[258] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1558607781. 19

[259] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by betweenclass attribute transfer. In *In CVPR*, 2009. 44

[260] G. Lample et al. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1030. URL https://www.aclweb.org/anthology/N16-1030. 19

[261] G. Lample, A. Conneau, L. Denoyer, and M. Ranzato. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations (ICLR)*, 2018. 61

[262] G. Lample et al. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018. 61

[263] G. Lample et al. Multiple-attribute text rewriting. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1g2NhC5KQ. 61, 62, 63, 66, 86

[264] Z. Lan et al. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=H1eA7AEtvS. 23, 31, 33, 34, 40

[265] H. Larochelle, D. Erhan, and Y. Bengio. Zero-data learning of new tasks. In *AAAI*, 2008. 44

[266] L. Laugier, J. Pavlopoulos, J. Sorensen, and L. Dixon. Civil rephrases of toxic texts with self-supervised transformers. In *EACL*, pages 1442–1461, Online, 2021. URL https://aclanthology.org/2021.eacl-main.124. 84, 86, 96, 98

[267] L. Laugier, J. Pavlopoulos, J. Sorensen, and L. Dixon. Civil rephrases of toxic texts with self-supervised transformers. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1442–1461, Online, Apr. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.124. URL https://aclanthology.org/2021.eacl-main.124. 15, 57

[268] A. Lavie and A. Agarwal. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W07-0734. 20, 82

[269] N. D. Lawrence and J. C. Platt. Learning to learn with the informative vector machine. In *Proceedings of the Twenty-First International Conference on Machine Learning*, ICML '04, page 65, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 1581138385. doi: 10.1145/1015330.1015382. URL https://doi.org/10.1145/1015330.1015382. 32

[270] H. Le et al. Flaubert: Unsupervised language model pre-training for french, 2019. 62

[271] Q. Le and T. Mikolov. Distributed representations of sentences and documents. In E. P. Xing and T. Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1188–1196, Bejing, China, 22–24 Jun 2014. PMLR. URL https://proceedings.mlr.press/v32/le14.html. 29

[272] T. Le Scao and A. Rush. How many data points is a prompt worth? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2627–2636, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.208. URL https://aclanthology.org/2021.naacl-main.208. 44

[273] Y. LeCun and M. Ishan. Self-supervised learning: The dark matter of intelligence, 2021. URL https://ai.facebook.com/blog/self-supervised-learning-the-dark-matter-of-intelligence/. 27

[274] Y. LeCun, Y. Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995. 51

[275] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791. 51

[276] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436–444, 2015. 14

[277] T. Lei, R. Barzilay, and T. Jaakkola. Rationalizing neural predictions. In *EMNLP*, pages 107–117, Austin, Texas, 2016. doi: 10.18653/v1/D16-1011. URL https://aclanthology.org/D16-1011. 84

[278] G. W. Leibniz. Memoir using the chain rule. *Cited in TMME*, 7:321–332, 2010. 50

[279] D. Lepikhin et al. Gshard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=qrwe7XHTmYb. 23, 31, 34

[280] B. Lester, R. Al-Rfou, and N. Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.243. URL https://aclanthology.org/2021.emnlp-main.243. 23, 45

[281] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710, feb 1966. Doklady Akademii Nauk SSSR, V163 No4 845-848 1965. 80

[282] H. Levesque, E. Davis, and L. Morgenstern. The winograd schema challenge. In *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*, 2012. 22

[283] A. Levi, O. Mokryn, C. Diot, and N. Taft. Finding a needle in a haystack of reviews: Cold start context-based hotel recommender system. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, RecSys '12, page 115–122, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450312707. doi: 10.1145/2365952.2365977. URL https://doi.org/10.1145/2365952.2365977. 109

[284] M. Lewis et al. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL https://aclanthology.org/2020.acl-main.703. 23, 31, 33, 34, 38

[285] G. F. A. L'Hospital. *Analyse des infiniment petits pour l'intelligence des lignes courbes*. chez François Montalant ..., A Paris, seconde edition edition, 1715. 50

[286] C. Li, S. Lin, and M. Shan. Exploiting endorsement information and social influence for item recommendation. In W. Ma et al., editors, *Proceeding of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2011, Beijing, China, July 25-29, 2011*, pages 1131–1132. ACM, 2011. doi: 10.1145/2009916.2010084. URL https://doi.org/10.1145/2009916.2010084. 109

[287] J. Li, W. Monroe, and D. Jurafsky. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*, 2016. 68, 84, 136

[288] J. Li, W. Monroe, and D. Jurafsky. Understanding neural networks through representation erasure. *CoRR*, abs/1612.08220, 2016. URL http://arxiv.org/abs/1612.08220. 26

[289] J. Li, R. Jia, H. He, and P. Liang. Delete, retrieve, generate: a simple approach to sentiment and style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1865–1874, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1169. URL https://www.aclweb.org/anthology/N18-1169. 61, 66, 67, 68, 73

[290] J. Li, T. Tang, W. X. Zhao, and J.-R. Wen. Pretrained language model for text generation: A survey. In Z.-H. Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4492–4499. International Joint Conferences on Artificial Intelligence Organization, 8 2021. doi: 10.24963/ijcai.2021/612. URL https://doi.org/10.24963/ijcai.2021/612. Survey Track. 21, 38

[291] Q. Li and Y. P. Chen. Personalized text snippet extraction using statistical language models. *Pattern Recogn.*, 43(1):378–386, jan 2010. ISSN 0031-3203. doi: 10.1016/j.patcog.2009.06.003. URL https://doi.org/10.1016/j.patcog.2009.06.003. 109

[292] X. L. Li and P. Liang. Prefix-tuning: Optimizing continuous prompts for generation, 2021. 45

[293] J. Liao et al. Generating human readable transcript for automatic speech recognition with pre-trained language model. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2021, Toronto, ON, Canada, June 6-11, 2021*, pages 7578–7582. IEEE, 2021. doi: 10.1109/ICASSP39728.2021.9414626. URL https://doi.org/10.1109/ICASSP39728.2021.9414626. 19

[294] O. Lieber, O. Sharir, B. Lenz, and Y. Shoham. Jurassic-1: Technical details and evaluation. Technical report, AI21 Labs, Aug. 2021. 23, 31, 34, 39, 45

[295] C.-Y. Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W04-1013. 20, 82

[296] C.-Y. Lin and F. J. Och. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04, page 605–es, USA, 2004. Association for Computational Linguistics. doi: 10.3115/1218955.1219032. URL https://doi.org/10.3115/1218955.1219032. 20

[297] D. Lin. An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, page 296–304, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. ISBN 1558605568. 109

[298] J. Lin, E. Keogh, S. Lonardi, and B. Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, DMKD '03, page 2–11, New York, NY, USA, 2003. Association for Computing Machinery. ISBN 9781450374224. doi: 10.1145/882082.882086. URL https://doi.org/10.1145/882082.882086. 31

[299] T. Lin, Y. Wang, X. Liu, and X. Qiu. A survey of transformers, 2021. URL https://arxiv.org/abs/2106.04554. 54

[300] T. Lin, Y. Wang, X. Liu, and X. Qiu. A survey of transformers. *arXiv preprint arXiv:2106.04554*, 2021. 21

[301] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Comput.*, 7:76–80, 2003. 108

[302] C.-Y. Liou, J.-C. Huang, and W.-C. Yang. Modeling word perception using the elman network. *Neurocomputing*, 71(16-18):3150–3157, 2008. 33

[303] C.-Y. Liou, W.-C. Cheng, J.-W. Liou, and D.-R. Liou. Autoencoder for words. *Neurocomputing*, 139:84–96, 2014. 33

[304] D. Liu et al. Revision in continuous space: Unsupervised text style transfer without adversarial learning. *arXiv preprint arXiv:1905.12304*, 2019. 68

[305] H. Liu et al. Hybrid neural recommendation with joint deep representation learning of ratings and reviews. *Neurocomputing*, 374:77–85, 2020. 109

[306] P. Liu, X. Qiu, and X. Huang. Recurrent neural network for text classification with multi-task learning. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, page 2873–2879. AAAI Press, 2016. ISBN 9781577357704. 34

[307] P. Liu et al. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*, 2021. 21, 33, 44

[308] P. J. Liu, Y.-A. Chung, and J. Ren. Summae: Zero-shot abstractive text summarization using length-agnostic auto-encoders, 2019. 61, 62, 63, 64, 67

[309] X. Liu et al. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 2021. doi: 10.1109/TKDE.2021.3090866. 26

[310] X. Liu et al. GPT understands, too. *CoRR*, abs/2103.10385, 2021. URL https://arxiv.org/abs/2103.10385. 45

[311] Y. Liu et al. Roberta: A robustly optimized bert pretraining approach, 2019. URL http://arxiv.org/abs/1907.11692. cite arxiv:1907.11692. 23, 31, 33, 34, 40, 46, 135

[312] Z. Liu et al. Finbert: A pre-trained financial language representation model for financial text mining. In C. Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 4513–4519. International Joint Conferences on Artificial Intelligence Organization, 7 2020. doi: 10.24963/ijcai.2020/622. URL https://doi.org/10.24963/ijcai.2020/622. Special Track on AI in FinTech. 33

[313] S. Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982. 27

[314] V. Logacheva et al. Paradetox: Detoxification with parallel data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6804–6818, 2022. 81

[315] L. Logeswaran, H. Lee, and S. Bengio. Content preserving text generation with attribute controls. In *Advances in Neural Information Processing Systems*, pages 5103–5113, 2018. 66

[316] F. López et al. Augmenting the user-item graph with textual similarity models. *CoRR*, abs/2109.09358, 2021. URL https://arxiv.org/abs/2109.09358. 110

[317] J. Luketina et al. A survey of reinforcement learning informed by natural language. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 6309–6317. International Joint Conferences on Artificial Intelligence Organization, 7 2019. doi: 10.24963/ijcai.2019/880. URL https://doi.org/10.24963/ijcai.2019/880. 24

[318] S. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*, 2017. 136

[319] F. Luo et al. A dual reinforcement learning framework for unsupervised text style transfer. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence, IJCAI 2019*, 2019. 24

[320] F. Luo et al. A dual reinforcement learning framework for unsupervised text style transfer. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 5116–5122. AAAI Press, 2019. 61, 68, 73

[321] H. Luo et al. Univilm: A unified video and language pre-training model for multimodal understanding and generation. *CoRR*, abs/2002.06353, 2020. URL https://arxiv.org/abs/2002.06353. 20

[322] S. T. Luu and N. Nguyen. UIT-ISE-NLP at SemEval-2021 Task 5: Toxic span detection with BiLSTM - CRF and toxic BERT comment classification. In *SemEval*, 2021. 137

[323] Q. Ma, J. Wei, O. Bojar, and Y. Graham. Results of the WMT19 metrics shared task: Segment-level and strong MT systems pose big challenges. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 62–90, Florence, Italy, Aug. 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5302. URL https://www.aclweb.org/anthology/W19-5302. 81

[324] X. Ma and E. Hovy. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany, Aug. 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1101. URL https://www.aclweb.org/anthology/P16-1101. 19

[325] X. Ma et al. Flowseq: Non-autoregressive conditional sequence generation with generative flow. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4273–4283, 2019. 82

[326] A. L. Maas et al. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P11-1015. 61

[327] J. MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967. 27

[328] K. Marino, M. Rastegari, A. Farhadi, and R. Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 3195–3204. Computer Vision Foundation / IEEE, 2019. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Marino_OK-VQA_A_Visual_Question_Answering_Benchmark_Requiring_External_Knowledge_CVPR_2019_paper.html. 81

[329] A. A. Markov. Essai d'une recherche statistique sur le texte du roman "Eugene Onegin" illustrant la liaison des epreuve en chain ('Example of a statistical investigation of the text of "Eugene Onegin" illustrating the dependence between samples in chain'). *Izvistia Imperatorskoi Akademii Nauk (Bulletin de l'Académie Impériale des Sciences de St.-Pétersbourg)*, 7:153–162, 1913. English translation by Morris Halle, 1956. 35

[330] L. Martin. *Automatic sentence simplification using controllable and unsupervised methods*. PhD thesis, Sorbonne Université, 2021. 80

[331] L. Martin et al. Multilingual unsupervised sentence simplification. *arXiv preprint arXiv:2005.00352*, 2020. 80

[332] L. Martin et al. Muss: multilingual unsupervised sentence simplification by mining paraphrases. *arXiv preprint arXiv:2005.00352*, 2020. 81

[333] G. D. S. Martino et al. A survey on computational propaganda detection. In *IJCAI*, pages 4826–4832, 2020. 84

[334] B. Mathew et al. Hatexplain: A benchmark dataset for explainable hate speech detection. In *AAAI*, pages 14867–14875, 2021. URL https://arxiv.org/abs/2012.10289. 84, 86, 91

[335] J. McAuley and J. Leskovec. Hidden factors and hidden topics: Understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, page 165–172, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450324090. doi: 10.1145/2507157.2507163. URL https://doi.org/10.1145/2507157.2507163. 109

[336] B. McCann, J. Bradbury, C. Xiong, and R. Socher. Learned in translation: Contextualized word vectors. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6297–6308, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964. 23, 34, 40

[337] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133, 1943. doi: 10.1007/bf02478259. 47

[338] O. Melamud, J. Goldberger, and I. Dagan. context2vec: Learning generic context embedding with bidirectional LSTM. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 51–61, Berlin, Germany, Aug. 2016. Association for Computational Linguistics. doi: 10.18653/v1/K16-1006. URL https://www.aclweb.org/anthology/K16-1006. 29, 40

[339] Z. Meng, R. McCreadie, C. Macdonald, and I. Ounis. Exploring data splitting strategies for the evaluation of recommendation models. In *Fourteenth ACM Conference on Recommender Systems*, RecSys '20, page 681–686, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450375832. doi: 10.1145/3383313.3418479. URL https://doi.org/10.1145/3383313.3418479. 110

[340] Z. Miao, Y. Li, X. Wang, and W.-C. Tan. *Snippext: Semi-Supervised Opinion Mining with Augmented Data*, page 617–628. Association for Computing Machinery, New York, NY, USA, 2020. ISBN 9781450370233. URL https://doi.org/10.1145/3366423.3380144. 26

[341] T. Mikolov et al. Recurrent Neural Network Based Language Model. In *Proceedings of the 11th Annual Conference of the International Speech Communication Association*, INTERSPEECH 2010, pages 1045–1048. ISCA, 2010. URL http://www.isca-speech.org/archive/interspeech_2010/i10_1045.html. 52

[342] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In Y. Bengio and Y. LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013. URL http://arxiv.org/abs/1301.3781. 23, 30

[343] T. Mikolov et al. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 3111–3119, Red Hook, NY, USA, 2013. Curran Associates Inc. 23, 30, 34, 46

[344] E. Miller, N. Matsakis, and P. Viola. Learning from one example through shared densities on transforms. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*, volume 1, pages 464–471 vol.1, 2000. doi: 10.1109/CVPR.2000.855856. 43

[345] G. A. Miller. *WordNet: An electronic lexical database*. MIT press, 1998. 109

[346] G. A. Miller and J. A. Selfridge. Verbal context and the recall of meaningful material. *The American journal of psychology*, 63(2):176–185, 1950. 35

[347] S. Min et al. Neurips 2020 efficientqa competition: Systems, analyses and lessons learned. In H. J. Escalante and K. Hofmann, editors, *Proceedings of the NeurIPS 2020 Competition and Demonstration Track*, volume 133 of *Proceedings of Machine Learning Research*, pages 86–111. PMLR, 06–12 Dec 2021. URL https://proceedings.mlr.press/v133/min21a.html. 81

[348] A. Miyake and P. Shah. Models of working memory: Mechanisms of active maintenance and executive control, 1999. 53

[349] A. K. Mohankumar et al. Towards transparent and explainable attention models. In *ACL*, pages 4206–4216, 2020. 136

[350] N. Mostafazadeh et al. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1098. URL https://www.aclweb.org/anthology/N16-1098. 22

[351] T. Müller, R. Cotterell, A. Fraser, and H. Schütze. Joint lemmatization and morphological tagging with lemming. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2268–2274, Lisbon, Portugal, Sept. 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1272. URL https://www.aclweb.org/anthology/D15-1272. 19

[352] C.-C. Musat, Y. Liang, and B. Faltings. Recommendation using textual opinions. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, page 2684–2690. AAAI Press, 2013. ISBN 9781577356332. 109

[353] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 807–814, Madison, WI, USA, 2010. Omnipress. ISBN 9781605589077. 47

[354] R. Nallapati et al. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany, Aug. 2016. Association for Computational Linguistics. doi: 10.18653/v1/K16-1028. URL https://aclanthology.org/K16-1028. 22

[355] S. Narang et al. Wt5?! training text-to-text models to explain their predictions. *CoRR*, abs/2004.14546, 2020. URL https://arxiv.org/abs/2004.14546. 26, 127

[356] S. Narang et al. Do transformer modifications transfer across implementations and applications? In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5758–5773, Online and Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.465. URL https://aclanthology.org/2021.emnlp-main.465. 54

[357] S. Narayan, S. B. Cohen, and M. Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical*

Methods in Natural Language Processing, pages 1797–1807, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1206. URL https://www.aclweb.org/anthology/D18-1206. 22

[358] D. Narayanan et al. Efficient large-scale language model training on GPU clusters. *CoRR*, abs/2104.04473, 2021. URL https://arxiv.org/abs/2104.04473. 23, 31, 34, 38, 45

[359] U. Naseem, I. Razzak, S. K. Khan, and M. Prasad. A comprehensive survey on word representation models: From classical to state-of-the-art word representation language models. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 20(5), jun 2021. ISSN 2375-4699. doi: 10.1145/3434237. URL https://doi.org/10.1145/3434237. 21

[360] D. Nguyen, D. Nguyen, D. Pham, and S. Pham. A robust transformation-based learning approach using ripple down rules for part-of-speech tagging. *AI Communications*, 29(3):409–422, Apr. 2016. ISSN 0921-7126. doi: 10.3233/AIC-150698. 19

[361] V. A. Nguyen, T. Nguyen, H. D. Quang, and Q. H. Pham. S-NLP at semeval-2021 task 5: Toxic spans detection. In *SemEval*, 2021. 135, 139

[362] M. A. Nielsen. *Neural networks and deep learning*, volume 25. Determination press San Francisco, CA, USA, 2015. 46

[363] X. Ning, C. Desrosiers, and G. Karypis. *A Comprehensive Survey of Neighborhood-Based Recommendation Methods*, pages 37–76. Springer US, Boston, MA, 2015. ISBN 978-1-4899-7637-6. doi: 10.1007/978-1-4899-7637-6_2. URL https://doi.org/10.1007/978-1-4899-7637-6_2. 108

[364] C. Nogueira dos Santos, I. Melnyk, and I. Padhi. Fighting offensive language on social media with unsupervised text style transfer. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 189–194, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2031. URL https://www.aclweb.org/anthology/P18-2031. 59, 62, 63, 86

[365] F. J. Och et al. A smorgasbord of features for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 161–168, Boston, Massachusetts, USA, May 2 - May 7 2004. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N04-1021. 19

[366] A. G. Oettinger. The uses of computers in science. *Scientific American*, 215(3):160–175, 1966. 18

[367] X. Ouyang et al. ERNIE-M: Enhanced multilingual representation by aligning cross-lingual semantics with monolingual corpora. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 27–38, Online and Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.3. URL https://aclanthology.org/2021.emnlp-main.3. 38

[368] M. Palatucci, D. Pomerleau, G. E. Hinton, and T. M. Mitchell. Zero-shot learning with semantic output codes. In Y. Bengio et al., editors, *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc., 2009. URL https://proceedings.neurips.cc/paper/2009/file/1543843a4723ed2ab08e18053ae6dc5b-Paper.pdf. 44

[369] M. Palomino, D. Grad, and J. Bedwell. An ensemble approach to identify toxicity in text. In *SemEval*, 2021. 136, 140

[370] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010. doi: 10.1109/TKDE.2009.191. 32

[371] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 79–86. Association for Computational Linguistics, July 2002. doi: 10.3115/1118693.1118704. URL https://www.aclweb.org/anthology/W02-1011. 27

[372] R. Y. Pang and K. Gimpel. Unsupervised evaluation metrics and learning criteria for non-parallel textual transfer. *EMNLP-IJCNLP 2019*, page 138, 2019. 61, 67

[373] D. Paperno et al. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1525–1534, Berlin, Germany, Aug. 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1144. URL https://www.aclweb.org/anthology/P16-1144. 22

[374] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://www.aclweb.org/anthology/P02-1040. 20, 67, 82

[375] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio. How to construct deep recurrent neural networks. In *Proceedings of the Second International Conference on Learning Representations (ICLR 2014)*, 2014. 52

[376] J. Pavlopoulos, P. Malakasiotis, and I. Androutsopoulos. Deep learning for user comment moderation. In *Proceedings of the 1st Workshop on Abusive Language Online*, pages 25–35, Vancouver, Canada, 2017. 91, 93

[377] J. Pavlopoulos, P. Malakasiotis, and I. Androutsopoulos. Deeper attention to abusive user content moderation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1125–1135, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1117. URL https://www.aclweb.org/anthology/D17-1117. 58, 84

[378] J. Pavlopoulos, P. Malakasiotis, and I. Androutsopoulos. Deep learning for user comment moderation. In *Proceedings of the First Workshop on Abusive Language Online*, pages 25–35, Vancouver, BC, Canada, Aug. 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-3004. URL https://www.aclweb.org/anthology/W17-3004. 88

[379] J. Pavlopoulos, N. Thain, L. Dixon, and I. Androutsopoulos. Convai at semeval-2019 task 6: Offensive language identification and categorization with perspective and bert. In *SemEval*, Minneapolis, USA, 2019. 94

[380] J. Pavlopoulos et al. Toxicity detection: Does context really matter? In *ACL*, pages 4296–4305, Online, 2020. 86

[381] J. Pavlopoulos, J. Sorensen, L. Laugier, and I. Androutsopoulos. SemEval-2021 task 5: Toxic spans detection. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 59–69, Online, Aug. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.semeval-1.6. URL https://aclanthology.org/2021.semeval-1.6. 15, 83

[382] J. Pavlopoulos et al. From the detection of toxic spans in online discussions to the analysis of toxic-to-civil transfer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3721–3734, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.259. URL https://aclanthology.org/2022.acl-long.259. 15, 83

[383] M. J. Pazzani. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, 13:393–408, 2004. 108

[384] T. Pedersen, S. Patwardhan, and J. Michelizzi. Wordnet::similarity - measuring the relatedness of concepts. pages 1024–1025, Dec. 2004. Proceedings - Nineteenth National Conference on Artificial Intelligence (AAAI-2004): Sixteenth Innovative Applications of Artificial Intelligence Conference (IAAI-2004) ; Conference date: 25-07-2004 Through 29-07-2004. 109

[385] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543,

Doha, Qatar, Oct. 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162. URL https://www.aclweb.org/anthology/D14-1162. 23, 30, 34

[386] D. N. Perkins, G. Salomon, et al. Transfer of learning. *International encyclopedia of education*, 2:6452–6457, 1992. 26

[387] M. Peters, W. Ammar, C. Bhagavatula, and R. Power. Semi-supervised sequence tagging with bidirectional language models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1756–1765, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1161. URL https://www.aclweb.org/anthology/P17-1161. 40

[388] M. Peters et al. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL https://www.aclweb.org/anthology/N18-1202. 23, 34, 40

[389] F. Petroni et al. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1250. URL https://aclanthology.org/D19-1250. 45

[390] S. Petrov, D. Das, and R. McDonald. A universal part-of-speech tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 2089–2096, Istanbul, Turkey, May 2012. European Languages Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2012/pdf/274_Paper.pdf. 19

[391] M. T. Pilehvar and os'e Camacho-Collados. Wic: 10, 000 example pairs for evaluating context-sensitive representations. *CoRR*, abs/1808.09121, 2018. URL http://arxiv.org/abs/1808.09121. 22

[392] K. Pluciński and H. Klimczak. GHOST at SemEval-2021 Task 5: Is explanation all you need? In *SemEval*, 2021. 136, 137, 140

[393] D. Poirier, F. Fessant, and I. Tellier. Reducing the Cold-Start Problem in Content Recommendation Through Opinion Classification. In *Web Intelligence*, Toronto, Canada, Aug. 2010. URL https://hal.inria.fr/inria-00514533. 109

[394] D. Poole, A. Mackworth, and R. Goebel. *Computational Intelligence: A Logical Approach*. Oxford University Press, Inc., USA, 1997. ISBN 0195102703. 14

[395] S. Prabhumoye, Y. Tsvetkov, R. Salakhutdinov, and A. W. Black. Style transfer through back-translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 866–876, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1080. URL https://www.aclweb.org/anthology/P18-1080. 61, 63

[396] R. Pryzant et al. Automatically neutralizing subjective bias in text. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 480–489, 2020. 61

[397] E. Puskala. Materials for teaching the pronunciation of english in finnish upper secondary schools, 2018. 53

[398] W. Qi et al. ProphetNet: Predicting future n-gram for sequence-to-SequencePre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2401–2410, Online, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.217. URL https://aclanthology.org/2020.findings-emnlp.217. 23, 31, 34, 38

[399] G. Qin and J. Eisner. Learning how to ask: Querying LMs with mixtures of soft prompts. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5203–5212, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.410. URL https://aclanthology.org/2021.naacl-main.410. 45

[400] X. Qiu et al. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897, 2020. 21

[401] A. Radford and K. Narasimhan. Improving language understanding by generative pre-training, 2018. 23, 31, 34, 38, 45, 46

[402] A. Radford et al. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019. 23, 29, 31, 34, 38, 43, 44, 45, 46, 66

[403] A. Radford et al. Learning transferable visual models from natural language supervision. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/radford21a.html. 20

[404] J. W. Rae et al. Scaling language models: Methods, analysis & insights from training gopher, 2021. URL https://arxiv.org/abs/2112.11446. 23, 31, 34, 39, 45

[405] C. Raffel et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html. 106, 125

[406] C. Raffel et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html. 23, 29, 31, 32, 33, 34, 38, 42, 43, 44, 45, 46, 62, 63, 64, 68, 96

[407] R. Raina et al. Self-taught learning: Transfer learning from unlabeled data. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, page 759–766, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595937933. doi: 10.1145/1273496.1273592. URL https://doi.org/10.1145/1273496.1273592. 32

[408] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, Nov. 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://www.aclweb.org/anthology/D16-1264. 22

[409] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016. 22

[410] P. Rajpurkar, R. Jia, and P. Liang. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2124. URL https://www.aclweb.org/anthology/P18-2124. 22, 92

[411] P. Ramachandran, B. Zoph, and Q. V. Le. Searching for activation functions. *ArXiv*, abs/1710.05941, 2018. 47

[412] A. Ramesh et al. Hierarchical text-conditional image generation with clip latents, 2022. URL https://arxiv.org/abs/2204.06125. 20, 46

[413] T. Ranasinghe, D. Sarkar, M. Zampieri, and A. Ororbia. WLV-RIT at SemEval-2021 Task 5: A neural transformer framework for detecting toxic spans. In *SemEval*, 2021. 136, 140

[414] S. Rao and J. Tetreault. Dear sir or madam, may I introduce the GYAFC dataset: Corpus, benchmarks and metrics for formality style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 129–140, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1012. URL https://www.aclweb.org/anthology/N18-1012. 61

[415] M. Ravikiran et al. Findings of the shared task on offensive span identification fromCode-mixed Tamil-English comments. In *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian*

Languages, pages 261–270, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.dravidianlangtech-1.40. URL https://aclanthology.org/2022.dravidianlangtech-1.40. 102

[416] S. Reddy, D. Chen, and C. D. Manning. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266, Mar. 2019. doi: 10.1162/tacl_a_00266. URL https://www.aclweb.org/anthology/Q19-1016. 22

[417] Y. Ren et al. A study of non-autoregressive model for sequence generation, 2020. 82

[418] P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'95, page 448–453, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. ISBN 1558603638. 109

[419] M. T. Ribeiro, S. Singh, and C. Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1135–1144, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939778. URL https://doi.org/10.1145/2939672.2939778. 26

[420] M. T. Ribeiro, S. Singh, and C. Guestrin. “Why Should I Trust You?” Explaining the predictions of any classifier. In *SIGKDD*, pages 1135–1144, San Francisco, USA, 2016. 84, 136

[421] F. Ricci, L. Rokach, and B. Shapira. *Introduction to Recommender Systems Handbook*, pages 1–35. Springer US, Boston, MA, 2011. ISBN 978-0-387-85820-3. doi: 10.1007/978-0-387-85820-3_1. URL https://doi.org/10.1007/978-0-387-85820-3_1. 106

[422] A. Ritter, C. Cherry, and B. Dolan. Unsupervised modeling of twitter conversations. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 172–180, Los Angeles, California, June 2010. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N10-1020. 62

[423] A. Rives et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, 2021. doi: 10.1073/pnas.2016239118. URL https://www.pnas.org/doi/abs/10.1073/pnas.2016239118. 46

[424] M. Roemmele, C. A. Bejan, and A. S. Gordon. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI Spring Symposium Series*, 2011. 22

[425] A. Rogers, O. Kovaleva, and A. Rumshisky. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866, 2020. doi: 10.1162/tacl_a_00349. URL https://aclanthology.org/2020.tacl-1.54. 50

[426] F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65 6:386–408, 1958. 47

[427] C. Rosset. Turing-nlg: A 17-billion-parameter language model by microsoft, Feb 2020. URL https://www.microsoft.com/en-us/research/blog/turing-nlg-a-17-billion-parameter-language-model-by-microsoft/. 23, 31, 34, 38, 45

[428] S. Rothe, S. Narayan, and A. Severyn. Leveraging pre-trained checkpoints for sequence generation tasks. *Transactions of the Association for Computational Linguistics*, 8:264–280, 2020. doi: 10.1162/tacl_a_00313. URL https://aclanthology.org/2020.tacl-1.18. 20

[429] D. Rubinstein, E. Levi, R. Schwartz, and A. Rappoport. How well do distributional models capture different types of semantic knowledge? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 726–730, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-2119. URL https://www.aclweb.org/anthology/P15-2119. 30

[430] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985. 50

[431] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. *Learning Internal Representations by Error Propagation*, page 318–362. MIT Press, Cambridge, MA, USA, 1986. ISBN 026268053X. 50

[432] D. E. Rumelhart, J. L. McClelland, and C. PDP Research Group, editors. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*. MIT Press, Cambridge, MA, USA, 1986. ISBN 026268053X. 52

[433] D. E. Rumelhart, J. L. McClelland, and C. PDP Research Group, editors. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 2: Psychological and Biological Models*. MIT Press, Cambridge, MA, USA, 1986. ISBN 0262132184. 52

[434] J. Rusert. NLP_UIOWA at Semeval-2021 Task 5: Transferring toxic sets to tag toxic spans. In *SemEval*, 2021. 136, 137, 140

[435] C. Saharia et al. Photorealistic text-to-image diffusion models with deep language understanding, 2022. URL https://arxiv.org/abs/2205.11487. 20, 46

[436] A. B. Sai, A. K. Mohankumar, and M. M. Khapra. A survey of evaluation metrics used for nlg systems. *ACM Comput. Surv.*, 55(2), jan 2022. ISSN 0360-0300. doi: 10.1145/3485766. URL https://doi.org/10.1145/3485766. 20

[437] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, volume 20, 2008. 108

[438] A. Sancheti, K. Krishna, B. V. Srinivasan, and A. Natarajan. Reinforced rewards framework for text style transfer. In J. M. Jose et al., editors, *Advances in Information Retrieval*, pages 545–560, Cham, 2020. Springer International Publishing. ISBN 978-3-030-45439-5. 24

[439] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2019. URL https://arxiv.org/abs/1910.01108. 23, 31, 34, 40

[440] V. Sanh et al. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*, 2021. 44

[441] R. P. Sans and A. R. Farràs. HLE-UPC at SemEval-2021 Task 5: Multi-Depth DistilBERT for toxic spans detection. In *SemEval*, 2021. 141

[442] M. Sap et al. Social bias frames: Reasoning about social and power implications of language. In *ACL*, pages 5477–5490, Online, 2020. doi: 10.18653/v1/2020.acl-main.486. URL https://aclanthology.org/2020.acl-main.486. 88

[443] N. Saunshi et al. A theoretical analysis of contrastive unsupervised representation learning. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5628–5637. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/saunshi19a.html. 34

[444] T. Schick and H. Schütze. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online, Apr. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.20. URL https://aclanthology.org/2021.eacl-main.20. 23, 45

[445] T. Schick and H. Schütze. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.185. URL https://aclanthology.org/2021.naacl-main.185. 23, 45

[446] T. Schick and H. Schütze. Few-shot text generation with natural language instructions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 390–402, 2021. 45

[447] A. Schmidt and M. Wiegand. A survey on hate speech detection using natural language processing. In *Workshop on Natural Language Processing for Social Media*, pages 1–10, Valencia, Spain, 2017. 84

[448] M. Schuster and K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997. doi: 10.1109/78.650093. 52

[449] H. Schwenk. Continuous space language models. *Computer Speech & Language*, 21(3):492–518, 2007. ISSN 0885-2308. doi: https://doi.org/10.1016/j.csl.2006.09.003. URL https://www.sciencedirect.com/science/article/pii/S0885230806000325. 35

[450] H. Schwenk and M. Douze. Learning joint multilingual sentence representations with neural machine translation. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 157–167, Vancouver, Canada, Aug. 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-2619. URL https://www.aclweb.org/anthology/W17-2619. 30

[451] T. Scialom and F. Hill. Beametrics: A benchmark for language generation evaluation evaluation. *arXiv preprint arXiv:2110.09147*, 2021. 81

[452] J. R. Searle. Minds, brains, and programs. *Behavioral and Brain Sciences*, 3(3):417–424, 1980. doi: 10.1017/S0140525X00005756. 18

[453] A. See, P. J. Liu, and C. D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1099. URL https://www.aclweb.org/anthology/P17-1099. 22

[454] T. Sellam, D. Das, and A. Parikh. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.704. URL https://www.aclweb.org/anthology/2020.acl-main.704. 20, 67, 82

[455] R. Sennrich, B. Haddow, and A. Birch. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*, 2015. 61

[456] R. Sennrich, B. Haddow, and A. Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, Aug. 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162. URL https://www.aclweb.org/anthology/P16-1162. 29

[457] R. Sennrich, B. Haddow, and A. Birch. Neural machine translation of rare words with subword units. In *ACL*, 2016. 135

[458] I. V. Serban et al. A deep reinforcement learning chatbot. *arXiv preprint arXiv:1709.02349*, 2017. 62

[459] B. Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009. URL http://axon.cs.byu.edu/~martinez/classes/778/Papers/settles.activelearning.pdf. 26

[460] G. Shani and A. Gunawardana. *Evaluating Recommendation Systems*, pages 257–297. Springer US, Boston, MA, 2011. ISBN 978-0-387-85820-3. doi: 10.1007/978-0-387-85820-3_8. URL https://doi.org/10.1007/978-0-387-85820-3_8. 110

[461] C. E. Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948. 35

[462] C. E. Shannon. Prediction and entropy of printed english. *The Bell System Technical Journal*, 30(1):50–64, 1951. doi: 10.1002/j.1538-7305.1951.tb01366.x. 35

[463] U. Shardanand and P. Maes. Social information filtering: Algorithms for automating "word of mouth". In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, page 210–217, USA, 1995. ACM Press/Addison-Wesley Publishing Co. ISBN 0201847051. doi: 10.1145/223904.223931. URL https://doi.org/10.1145/223904.223931. 108

[464] R. Sharma, J. Allen, O. Bakhshandeh, and N. Mostafazadeh. Tackling the story ending biases in the story cloze test. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 752–757, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2119. URL https://www.aclweb.org/anthology/P18-2119. 22

[465] N. Shazeer and M. Stern. Adafactor: Adaptive learning rates with sublinear memory cost. *arXiv preprint arXiv:1804.04235*, 2018. 71

[466] T. Shen, T. Lei, R. Barzilay, and T. Jaakkola. Style transfer from non-parallel text by cross-alignment. In *Advances in neural information processing systems*, pages 6830–6841, 2017. 61, 66, 68, 72, 73, 86

[467] H. Shimanaka, T. Kajiwara, and M. Komachi. RUSE: Regressor using sentence embeddings for automatic machine translation evaluation. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 751–758, Belgium, Brussels, Oct. 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6456. URL https://www.aclweb.org/anthology/W18-6456. 20

[468] H. Shimanaka, T. Kajiwara, and M. Komachi. Ruse: Regressor using sentence embeddings for automatic machine translation evaluation. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 751–758, 2018. 67

[469] A. Shimorina, C. Gardent, S. Narayan, and L. Perez-Beltrachini. *WebNLG challenge: Human evaluation results*. PhD thesis, Loria & Inria Grand Est, 2018. 81

[470] T. Shin et al. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, Online, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.346. URL https://aclanthology.org/2020.emnlp-main.346. 23, 45

[471] M. Shoeybi et al. Megatron-lm: Training multi-billion parameter language models using model parallelism. *CoRR*, abs/1909.08053, 2019. URL http://arxiv.org/abs/1909.08053. 23, 31, 34, 38, 45

[472] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 50

[473] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013. 50

[474] T. D. Smedt et al. Profanity & offensive words (POW). *Textgain*, 2020. 136

[475] S. Smith et al. Using deepspeed and megatron to train megatron-turing NLG 530b, A large-scale generative language model. *CoRR*, abs/2201.11990, 2022. URL https://arxiv.org/abs/2201.11990. 23, 31, 34, 39, 45

[476] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng. Zero-shot learning through cross-modal transfer. In C. Burges et al., editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper/2013/file/2d6cc4b2d139a53512fb8cbb3086ae2e-Paper.pdf. 44

[477] R. Socher et al. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, Oct. 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D13-1170. 22, 27

[478] K. Song et al. Mass: Masked sequence to sequence pre-training for language generation. In *International Conference on Machine Learning*, pages 5926–5936, 2019. 23, 31, 33, 34, 38, 42

[479] K. Song et al. Mpnet: Masked and permuted pre-training for language understanding. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546. 23, 42

[480] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014. 50

[481] M. Srifi, A. Oussous, A. Ait Lahcen, and S. Mouline. Recommender systems based on collaborative filtering using review texts—a survey. *Information*, 11(6), 2020. ISSN 2078-2489. doi: 10.3390/info11060317. URL `https://www.mdpi.com/2078-2489/11/6/317`. 106

[482] M. Stephen et al. Pointer sentinel mixture models. In *Proc. the International Conference on Learning Representations (ICLR)*, 2016. 29

[483] B. Stevo and F. Ante. The influence of pattern similarity and transfer learning upon the training of a base perceptron b2. In *Proceedings of Symposium Informatica*, pages 3–121, 1976. 32

[484] J.-H. Su, W.-Y. Chang, and V. S. Tseng. Effective social content-based collaborative filtering for music recommendation. *Intelligent Data Analysis*, 21(S1):S195–S216, 2017. 108

[485] X. Su and T. M. Khoshgoftaar. Collaborative filtering for multi-class data using belief nets algorithms. *2006 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06)*, pages 497–504, 2006. 108

[486] A. Sugiyama and N. Yoshinaga. Data augmentation using back-translation for context-aware neural machine translation. In *Proceedings of the Fourth Workshop on Discourse in Machine Translation (DiscoMT 2019)*, pages 35–44, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-6504. URL `https://aclanthology.org/D19-6504`. 26

[487] Y. Suhara, X. Wang, S. Angelidis, and W.-C. Tan. OpinionDigest: A simple framework for opinion summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5789–5798, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.513. URL `https://aclanthology.org/2020.acl-main.513`. 26

[488] S. Sukhbaatar, a. szlam, J. Weston, and R. Fergus. End-to-end memory networks. In C. Cortes et al., editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL `https://proceedings.neurips.cc/paper/2015/file/8fb21ee7a2207526da55a679f0332de2-Paper.pdf`. 52

[489] T. A. Suman and A. Jain. AStarTwice at SemEval-2021 Task 5: Toxic span detection using RoBERTa-CRF, domain specific pre-training and self-training. In *SemEval*, 2021. 137

[490] C. Sun, F. Baradel, K. Murphy, and C. Schmid. Contrastive bidirectional transformer for temporal representation learning. *CoRR*, abs/1906.05743, 2019. URL `http://arxiv.org/abs/1906.05743`. 20

[491] C. Sun et al. Videobert: A joint model for video and language representation learning. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7463–7472, 2019. 20

[492] Y. Sun et al. Ernie 2.0: A continual pre-training framework for language understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8968–8975, Apr. 2020. doi: 10.1609/aaai.v34i05.6428. URL https://ojs.aaai.org/index.php/AAAI/article/view/6428. 23, 31, 33, 34

[493] Y. Sun et al. ERNIE 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *CoRR*, abs/2107.02137, 2021. URL https://arxiv.org/abs/2107.02137. 23, 31, 33, 34, 40, 46

[494] S. Surya et al. Unsupervised neural text simplification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2058–2068, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1198. URL https://www.aclweb.org/anthology/P19-1198. 80

[495] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA, 2014. MIT Press. 23, 31, 37, 38, 52

[496] C. Sutton and A. McCallum. An Introduction to Conditional Random Fields for relational learning. *Introduction to statistical relational learning*, 2:93–128, 2006. 135

[497] C. Szegedy et al. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015. doi: 10.1109/CVPR.2015.7298594. 50

[498] A. Tamkin et al. Dabs: a domain-agnostic benchmark for self-supervised learning. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/file/8d5e957f297893487bd98fa830fa6413-Paper-round1.pdf. 46

[499] W. L. Taylor. "cloze procedure": A new tool for measuring readability. *Journalism Quarterly*, 30(4):415–433, 1953. doi: 10.1177/107769905303000401. URL https://doi.org/10.1177/107769905303000401. 40

[500] I. Tenney, D. Das, and E. Pavlick. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1452. URL https://www.aclweb.org/anthology/P19-1452. 50

[501] M. Terzi, M. Rowe, M. Ferrario, and J. Whittle. Text-based user-knn: measuring user similarity based on text reviews. In V. Dimitrova et al., editors, *User Modeling, Adaptation, and Personalization*, Lecture Notes in Computer Science, pages 195–206. Springer, 2014. ISBN 9783319087863. doi: 10.1007/978-3-319-08786-3_

17. URL https://link.springer.com/book/10.1007/978-3-319-08786-3. International Conference on User Modelling, Adaptation, and Personalization (was AH and UM) 2014, UMAP 2014 ; Conference date: 07-07-2014 Through 11-07-2014. 106, 109, 110

[502] R. Thoppilan et al. Lamda: Language models for dialog applications, 2022. URL https://arxiv.org/abs/2201.08239. 23, 29, 31, 34, 39, 45

[503] S. Thrun. *Lifelong Learning Algorithms*, page 181–209. Kluwer Academic Publishers, USA, 1998. ISBN 0792380479. 32

[504] S. Thrun and L. Pratt. *Learning to learn*. Springer Science & Business Media, 2012. 32

[505] E. F. Tjong Kim Sang and F. De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147, 2003. URL https://www.aclweb.org/anthology/W03-0419. 19

[506] J. Turian, L.-A. Ratinov, and Y. Bengio. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P10-1040. 30, 34

[507] A. M. Turing. Computing machinery and intelligence. *Mind*, 59(October):433–60, 1950. doi: 10.1093/mind/LIX.236.433. 18

[508] A. Tversky. Features of similarity. *Psychological review*, 84(4):327, 1977. 137

[509] M. Twain and F. H. Cornish. *The adventures of Tom Sawyer*. Macmillan, 2015. 43

[510] B. Upadhyay, A. Sudhakar, and A. Maheswaran. Efficient reinforcement learning for unsupervised controlled text generation, 2022. URL https://arxiv.org/abs/2204.07696. 24

[511] B. Van Aken, J. Risch, R. Krestel, and A. Löser. Challenges for toxic comment classification: An in-depth error analysis. In *Proceedings of the 2nd Workshop on Abusive Language Online*, pages 33–42, Brussels, Belgium, 2018. 84, 86

[512] L. van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86): 2579–2605, 2008. URL http://jmlr.org/papers/v9/vandermaaten08a.html. 111

[513] C. Van Hee et al. Automatic detection of cyberbullying in social media text. *PLOS ONE*, 13(10):1–22, 10 2018. doi: 10.1371/journal.pone.0203794. URL https://doi.org/10.1371/journal.pone.0203794. 58

[514] A. Vaswani et al. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 23, 31, 52, 53, 54, 58, 62, 63

[515] A. Vaswani et al. Attention is all you need. In I. Guyon et al., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf. 106, 139

[516] R. Vedantam, C. L. Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4566–4575, 2015. doi: 10.1109/CVPR.2015.7299087. 81

[517] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 1096–1103, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605582054. doi: 10.1145/1390156.1390294. URL https://doi.org/10.1145/1390156.1390294. 33, 61

[518] P. Vincent et al. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.*, 11:3371–3408, dec 2010. ISSN 1532-4435. 33

[519] R. Voigt et al. RtGender: A corpus for studying differential responses to gender. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, Miyazaki, Japan, May 2018. European Languages Resources Association (ELRA). URL https://www.aclweb.org/anthology/L18-1445. 61

[520] A. Waibel et al. Phoneme recognition using time-delay neural networks. *IEEE transactions on acoustics, speech, and signal processing*, 37(3):328–339, 1989. 51

[521] A. Wang et al. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium, Nov. 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5446. URL https://www.aclweb.org/anthology/W18-5446. 19

[522] A. Wang et al. Superglue: A stickier benchmark for general-purpose language understanding systems. In H. Wallach et al., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/4496bf24afe7fab6f046bf4923da8de6-Paper.pdf. 19

[523] K. Wang, H. Hua, and X. Wan. Controllable unsupervised text attribute transfer via editing entangled latent representation. In *Advances in Neural Information Processing Systems*, pages 11036–11046, 2019. 66, 68

[524] X. Wang et al. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery &amp; Data Mining*, KDD '19, page 950–958, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362016. doi: 10.1145/3292500.3330989. URL https://doi.org/10.1145/3292500.3330989. 110

[525] X. Wang et al. Neural graph collaborative filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 165–174, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450361729. doi: 10.1145/3331184.3331267. URL https://doi.org/10.1145/3331184.3331267. 110

[526] A. Warstadt, A. Singh, and S. R. Bowman. Neural network acceptability judgments. *arXiv preprint arXiv:1805.12471*, 2018. 22

[527] J. Wei and K. Zou. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1670. URL https://aclanthology.org/D19-1670. 26

[528] J. Wei et al. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021. 44

[529] G. Wenzek et al. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL https://aclanthology.org/2020.lrec-1.494. 80

[530] J. Weston, S. Chopra, and A. Bordes. Memory networks. In Y. Bengio and Y. LeCun, editors, *ICLR*, 2015. URL http://dblp.uni-trier.de/db/conf/iclr/iclr2015.html#WestonCB14. 52

[531] B. Widrow and M. E. Hoff. Adaptive switching circuits. *1960 IRE WESCON Convention Record*, pages 96–104, 1960. Reprinted in *Neurocomputing* MIT Press, 1988 . 47

[532] S. Wiegreffe and Y. Pinter. Attention is not not explanation. In *EMNLP-IJCNLP*, pages 11–20, Hong Kong, China, 2019. doi: 10.18653/v1/D19-1002. URL https://www.aclweb.org/anthology/D19-1002. 91

[533] Wikipedia contributors. Black and white — Wikipedia, the free encyclopedia, 2022. URL https://en.wikipedia.org/w/index.php?title=Black_and_white&oldid=1072075133. [Online; accessed 13-May-2022]. 29

[534] Wikipedia contributors. Michael moore — Wikipedia, the free encyclopedia, 2022. URL https://en.wikipedia.org/w/index.php?title=Michael_Moore&oldid=1087445612. [Online; accessed 13-May-2022]. 30

[535] Wiktionary. Monday-morning quarterback — wiktionary, the free dictionary, 2020. URL https://en.wiktionary.org/w/index.php?title=Monday-morning_quarterback&oldid=59218751. [Online; accessed 11-May-2022]. 26

[536] Wiktionary. carpetbagger — wiktionary, the free dictionary, 2022. URL https://en.wiktionary.org/w/index.php?title=carpetbagger&oldid=65486344. [Online; accessed 11-May-2022]. 26

[537] A. Williams, N. Nangia, and S. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1101. URL https://www.aclweb.org/anthology/N18-1101. 22

[538] R. J. Williams and D. Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1(2):270–280, 1989. doi: 10.1162/neco.1989.1.2.270. 37

[539] C. Wu, X. Ren, F. Luo, and X. Sun. A hierarchical reinforced sequence operation method for unsupervised text style transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4873–4883, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1482. URL https://www.aclweb.org/anthology/P19-1482. 24

[540] C. Wu, X. Ren, F. Luo, and X. Sun. A hierarchical reinforced sequence operation method for unsupervised text style transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4873–4883, 2019. 61, 68

[541] X. Wu et al. Mask and infill: Applying masked language model for sentiment transfer. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5271–5277. International Joint Conferences on Artificial Intelligence Organization, 7 2019. doi: 10.24963/ijcai.2019/732. URL https://doi.org/10.24963/ijcai.2019/732. 68

[542] Z. Wu, H. Tian, X. Zhu, and S. Wang. Optimization matrix factorization recommendation algorithm based on rating centrality. In Y. Tan, Y. Shi, and Q. Tang, editors, *Data Mining and Big Data*, pages 114–125, Cham, 2018. Springer International Publishing. ISBN 978-3-319-93803-5. 109

[543] E. Wulczyn, N. Thain, and L. Dixon. Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1391–1399, 2017. 58, 84, 86, 91

[544] A. Xenos et al. Toxicity detection sensitive to conversational context. *First Monday*, 27(5), Sep. 2022. doi: 10.5210/fm.v27i5.12285. URL https://firstmonday.org/ojs/index.php/fm/article/view/12285. 15, 26, 130

[545] Q. Xia et al. Xgpt: Cross-modal generative pre-training for image captioning. In *NLPCC (1)*, pages 786–797, 2021. URL https://doi.org/10.1007/978-3-030-88480-2_63. 20

[546] Z. Xia, Y. Dong, and G. Xing. Support vector machines for collaborative filtering. In *Proceedings of the 44th Annual Southeast Regional Conference*, ACM-SE 44, page 169–174, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595933158. doi: 10.1145/1185448.1185487. URL https://doi.org/10.1145/1185448.1185487. 108

[547] J. Xu et al. Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 979–988, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1090. URL https://www.aclweb.org/anthology/P18-1090. 24

[548] J. Xu et al. Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 979–988, 2018. 61

[549] P. Xu, J. C. K. Cheung, and Y. Cao. On variational learning of controllable representations for text without supervision. *arXiv preprint arXiv:1905.11975*, 2019. 66

[550] Q. Xu, L. Qu, C. Xu, and R. Cui. Privacy-aware text rewriting. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 247–257, Tokyo, Japan, 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-8633. URL https://www.aclweb.org/anthology/W19-8633. 61

[551] W. Xu et al. Paraphrasing for style. In *Proceedings of COLING 2012*, pages 2899–2914, Mumbai, India, Dec. 2012. The COLING 2012 Organizing Committee. URL https://www.aclweb.org/anthology/C12-1177. 61

[552] Y. Yang et al. Multilingual universal sentence encoder for semantic retrieval. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 87–94, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-demos.12. URL https://aclanthology.org/2020.acl-demos.12. 106, 111

[553] Z. Yang et al. Unsupervised text style transfer using language models as discriminators. In S. Bengio et al., editors, *Advances in Neural Information Processing Systems 31*, pages 7287–7298. Curran Associates, Inc., 2018. URL http://papers.nips.cc/paper/7959-unsupervised-text-style-transfer-using-language-models-as-discriminators.pdf. 66

[554] Z. Yang et al. Xlnet: Generalized autoregressive pretraining for language understanding. In H. Wallach et al., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/dc6a7e655d7e5840e66733e9ee67cc69-Paper.pdf. 21

[555] Z. Yang et al. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764, 2019. 23, 31, 33, 34, 42

[556] Z. Yang et al. Tap: Text-aware pre-training for text-vqa and text-caption. In *CVPR*, 2021. 20

[557] L. Yann. *Modeles connexionnistes de lapprentissage*. PhD thesis, These de Doctorat, Universite Paris, 1987. 33

[558] P. Yin, G. Neubig, W.-t. Yih, and S. Riedel. TaBERT: Pretraining for joint understanding of textual and tabular data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8413–8426, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.745. URL https://aclanthology.org/2020.acl-main.745. 46

[559] M. Zampieri et al. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). In *SemEval*, 2019. 84

[560] M. Zampieri et al. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86, 2019. 58

[561] M. Zampieri et al. Semeval-2020 task 12: Multilingual offensive language identification in social media (offenseval 2020). *arXiv preprint arXiv:2006.07235*, 2020. 101

[562] T. P. Zanto and A. Gazzaley. Neural suppression of irrelevant information underlies optimal working memory performance. *Journal of Neuroscience*, 29(10):3059–3066, 2009. 53

[563] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 50

[564] R. Zellers et al. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL https://www.aclweb.org/anthology/P19-1472. 22

[565] R. Zellers et al. Defending against neural fake news. In *Advances in Neural Information Processing Systems 32*, 2019. 23, 31, 34, 38

[566] Y. Zemlyanskiy et al. Docent: Learning self-supervised entity representations from large document collections. In *Proceedings of EACL*, 2021. URL https://www.aclweb.org/anthology/2021.eacl-main.217.pdf. 127

[567] W. Zeng et al. Pangu-α: Large-scale autoregressive pretrained chinese language models with auto-parallel computation. *ArXiv*, abs/2104.12369, 2021. 23, 31, 34, 39, 45

[568] J. Zhang et al. Conversations gone awry: Detecting early signs of conversational failure. *arXiv preprint arXiv:1805.05345*, 2018. 58

[569] J. Zhang, Y. Zhao, M. Saleh, and P. J. Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization, 2019. 23, 31, 33, 34, 42

[570] J. Zhang, Y. Zhao, M. Saleh, and P. Liu. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/zhang20ae.html. 20

[571] L. Zhang, S. Wang, and B. Liu. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1253, 2018. 27

[572] S. Zhang et al. Record: Bridging the gap between human and machine commonsense reading comprehension. *arXiv preprint arXiv:1810.12885*, 2018. 22

[573] S. Zhang, L. Yao, A. Sun, and Y. Tay. Deep learning based recommender system: A survey and new perspectives. *ACM Comput. Surv.*, 52(1), feb 2019. ISSN 0360-0300. doi: 10.1145/3285029. URL https://doi.org/10.1145/3285029. 108

[574] T. Zhang et al. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2019. 67

[575] T. Zhang* et al. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SkeHuCVFDr. 20, 82

[576] Y. Zhang and J. Nivre. Transition-based dependency parsing with rich non-local features. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 188–193, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P11-2033. 19

[577] Y. Zhang et al. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '14, page 83–92, New York, NY, USA, 2014. Association for Computing Machinery.

ISBN 9781450322577. doi: 10.1145/2600428.2609579. URL https://doi.org/10.1145/2600428.2609579. 109

[578] Y. Zhang, J. Xu, P. Yang, and X. Sun. Learning sentiment memories for sentiment modification without parallel data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1103–1108, Brussels, Belgium, 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1138. URL https://www.aclweb.org/anthology/D18-1138. 68

[579] Y. Zhang et al. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-demos.30. URL https://aclanthology.org/2020.acl-demos.30. 20

[580] Z. Zhang, D. Zhang, and J. Lai. urcf:user review enhanced collaborative filtering. *20th Americas Conference on Information Systems, AMCIS 2014*, 01 2014. 108

[581] Z. Zhang et al. Style transfer as unsupervised machine translation. *arXiv preprint arXiv:1808.07894*, 2018. 61, 68

[582] Z. Zhang et al. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1139. URL https://www.aclweb.org/anthology/P19-1139. 23, 31, 33, 34, 46

[583] Z. Zhang et al. Recent advances and challenges in task-oriented dialog systems. *Science China Technological Sciences*, 63(10):2011–2027, 2020. 20

[584] Z. Zhang, K. Rudra, and A. Anand. Explain and predict, and then predict again. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 418–426, 2021. 84

[585] Y. Zhao, L. Chen, Z. Chen, and K. Yu. Semi-supervised text simplification with back-translation and asymmetric denoising autoencoders. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9668–9675, 2020. 80

[586] H. F. Zhen Wang and J. Liu. MedAI at SemEval-2021 Task 5: Start-to-end tagging framework for toxic spans detection. In *SemEval*, 2021. 136

[587] L. Zheng, V. Noroozi, and P. S. Yu. Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM '17, page 425–434, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450346757. doi: 10.1145/3018661.3018665. URL https://doi.org/10.1145/3018661.3018665. 109

[588] Z. Zhong, D. Friedman, and D. Chen. Factual probing is [MASK]: Learning vs. learning to recall. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5017–5033, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.398. URL https://aclanthology.org/2021.naacl-main.398. 45

[589] G. Zhou et al. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery &amp; Data Mining*, KDD '18, page 1059–1068, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355520. doi: 10.1145/3219819.3219823. URL https://doi.org/10.1145/3219819.3219823. 110

[590] L. Zhou, J. Gao, D. Li, and H.-Y. Shum. The Design and Implementation of XiaoIce, an Empathetic Social Chatbot. *Computational Linguistics*, 46(1):53–93, 03 2020. ISSN 0891-2017. doi: 10.1162/coli_a_00368. URL https://doi.org/10.1162/coli_a_00368. 20

[591] L. Zhou et al. Unified vision-language pre-training for image captioning and vqa. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07):13041–13049, Apr. 2020. doi: 10.1609/aaai.v34i07.7005. URL https://ojs.aaai.org/index.php/AAAI/article/view/7005. 20

[592] Z.-H. Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 5, 08 2017. doi: 10.1093/nsr/nwx106. 24, 26

[593] Z.-H. Zhou and M. Li. Semi-supervised learning by disagreement. *Knowl. Inf. Syst.*, 24(3):415–439, 2010. URL http://dblp.uni-trier.de/db/journals/kais/kais24.html#ZhouL10. 26

[594] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 59, 60

[595] Q. Zhu et al. HITSZ-HLT at SemEval-2021 Task 5: Span-based ensemble model with toxic lexicon. In *SemEval*, 2021. 135, 136, 139, 140

[596] Y. Zhu et al. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 19–27, Los Alamitos, CA, USA, dec 2015. IEEE Computer Society. doi: 10.1109/ICCV.2015.11. URL https://doi.ieeecomputersociety.org/10.1109/ICCV.2015.11. 29

DOCTORAL
SCHOOL

Titre: Analyse et Contrôle des Interactions en ligne avec des Réseaux Neuronaux Artificiels de Traitement Automatique du Langage Naturel

Mots clés: Apprentissage automatique, Intelligence artificielle, Traitement automatique du langage naturel, Informatique, Réseaux de neurones artificiels, Systèmes de recommandation

Résumé: Le traitement automatique du langage naturel est motivé par des applications où les ordinateurs doivent acquérir une compréhension sémantique et syntaxique du langage humain. Récemment, le domaine a été impacté par un changement de paradigme. Les architectures d'apprentissage profond couplées à des techniques d'apprentissage auto-supervisé sont devenues le cœur des modèles correspondant à l'état de l'art en compréhension et génération du langage naturel. Parfois considérés comme des "foundation models", ces systèmes ouvrent la voie à de nouveaux cas d'utilisation. Née d'un partenariat académique et industriel entre l'Institut Polytechnique de Paris et Google AI Research, la présente recherche s'est concentrée sur l'étude de la façon dont les modèles neuronaux de traitement du langage naturel pré-entraînés pouvaient être utilisés pour améliorer les interactions en ligne.

Cette thèse a d'abord exploré comment le transfert de style auto-supervisé pouvait être appliqué à la reformulation non-toxique de commentaires offensants dans les conversations en ligne. Dans le contexte de la modération de contenu toxique en ligne, nous avons proposé une méthode de réglage fin d'un modèle texte-à-texte pré-entraîné (T5) avec une fonction-objectif consistant en un auto-encodeur débruiteur cyclique.

Ensuite, les travaux de recherche ont porté sur l'étude de l'annotation humaine et la détection automatique des sous-ensembles de mots toxiques dans les conversations en ligne. Nous avons publié un nouveau jeu de données annoté pour entraîner et évaluer les systèmes automatiques, ce qui a conduit à une tâche partagée lors du 15e International Workshop on Semantic Evaluation.

Enfin, nous avons développé un système de recommandation basé sur des avis en ligne, s'inscrivant dans l'explicabilité des préférences prises en compte par les recommandations prédites. La méthode utilise des modèles basés sur la similarité sémantique textuelle pour représenter les préférences d'un utilisateur sous la forme d'un graphe de fragments de texte, où les arrêtes sont définies par la similarité sémantique.

Title: Analysis and Control of Online Interactions through Neural Natural Language Processing

Keywords: Machine learning, Artificial intelligence, Natural language processing, Computer Science, Artificial neural networks, Recommender systems

Abstract: Natural Language Processing is motivated by applications where computers should gain a semantic and syntactic understanding of human language. Recently, the field has been impacted by a paradigm shift. Deep learning architectures coupled with self-supervised training have become the core of state-of-the-art models used in Natural Language Understanding and Natural Language Generation. Sometimes considered as foundation models, these systems pave the way for novel use cases. Driven by an academic-industrial partnership between the Institut Polytechnique de Paris and Google AI Research, the present research has focused on investigating how pretrained neural Natural Language Processing models could be leveraged to improve online interactions.

This thesis first explored how self-supervised style transfer could be applied to the toxic-to-civil rephrasing of offensive comments found in online conversations. In the context of toxic content moderation online, we proposed to fine-tune a pretrained text-to-text model (T5) with a denoising and cyclic auto-encoder loss.

Then, a subsequent work investigated the human labeling and automatic detection of toxic spans in online conversations. We released a new labeled dataset to train and evaluate systems, which led to a shared task at the 15th International Workshop on Semantic Evaluation.

Finally, we developed a recommender system based on online reviews of items, taking part in the topic of explaining users' tastes considered by the predicted recommendations. The method uses textual semantic similarity models to represent a user's preferences as a graph of textual snippets, where the edges are defined by semantic similarity.

Institut Polytechnique de Paris
91120 Palaiseau, France