Effects of Pretraining Corpora on Scientific Relation Extraction Using BERT and SciBERT

Andrija Poleksić[1], Sanda Martinčić-Ipšić[1,2]

[1]Faculty of Informatics and Digital Technologies (University of Rijeka), Radmile Matejčić 2, Rijeka, 51000, Croatia
[2]Center for Artificial Intelligence and Cybersecurity

Abstract

The amount of data is swiftly increasing and processing of information in unstructured textual data can be of great importance. The natural language processing task that tries to automate this process is information extraction (IE), or rather its subtask, relation extraction. Relation extraction is tasked with identification of relations between entities in each sentence, paragraph or larger unit of text in order to automatically create machine-interpretable data collections of entities, relationships between entities, and attributes describing entities. This paper was motivated by research question of whether SciBERT outperforms BERT, state-of-the-art "*Bidirectional Encoder Representations from Transformers*" model, on the relation extraction task after fine-tuning to the corpora in the domain of science. An overview of datasets suitable for the task of extracting sentence-level relations is elaborated. A new variant of dataset for relation extraction in the domain of science, combo160, is created and used to fine-tune the BERT and SciBERT models. The results show a noticeable increase of 2.63% (on average) in the performance of the SciBERT model over the baseline BERT model, when faced with relations in the scientific domain. It can be inferred from the results that thematically similar (here: scientific) pretrainig corpora can improve the performance of the later fine-tuned models for relation extraction.

Keywords

relation extraction, relation classification, BERT, SciBERT, scientific dataset

1. Introduction

Natural language processing (NLP), as a research area of computer science and artificial intelligence (AI), focuses on the design and analysis of computational algorithms and representations for processing natural human language [1]. NLP has numerous tasks and applications, including machine translation, text summarization, information retrieval, sentiment analysis, text classification, topic modeling, and **information extraction**.

Information extraction (IE) is the subfield of NLP that deals with the problem of extracting information from unstructured texts, as defined in [2]:

> "*Information Extraction refers to the automatic extraction of structured information such as entities, relationships between entities, and attributes describing entities from unstructured sources.*"

SEMANTICS 2023 EU: 19th International Conference on Semantic Systems, September 20-22, 2023, Leipzig, Germany
andrija.2@hotmail.com (A. Poleksić); smarti@uniri.hr (S. Martinčić-Ipšić)

IE is not a trivial problem, it is divided into a number of smaller tasks, which include named entity recognition (NER), coreference resolution, and relation extraction (RE). **Relation extraction (RE)** is the subtask of information extraction that consists of identifying relations between entities in each sentence, paragraph, or larger unit of text. It usually involves extracting the relation between two or more (named) entities [3]. **Named entities (NE)** are traditionally detected by first applying NER and then RE. The result can be defined as a word or phrase representing a specific real-world object. To further clarify, RE model takes unstructured text with/without marked entities as an input and outputs the triples that usually resemble:

(subject (named entity), **relation** (relation type), **object** (named entity)**)**

Relation extraction can be modeled as the classification problem (Section 2 overviews different approaches to relation extraction). Hence, in this work, the BERT model is used as a base language model for the relation classification task due to its proven good results on the majority of NLP tasks [4][5][6][7]. In addition, a performance comparison of two models **SciBERT and BERT**, fine-tuned on the created dataset called **combo160**, is performed. Experiments provide insights into the contributions of using different corpora during pretraining. Primarily, this work is an extension of our previous work in [8], and poses a research question whether SciBERT outperforms BERT model on the relation extraction task after fine-tuning to the corpora in the domain of science.

In the next Section 2 the task of relation extraction and connection to the construction of knowledge graphs is elaborated, followed up by the related work. Section 3 elaborates on the used methods: evaluation metrics (Section 3.2), basics transformer architecture and the architectures of BERT and SciBERT (Section 3.1), and relation extraction datasets (3.3). Section 4 presents the created dataset **combo160** and the experimental setup. The final Sections of the paper 5 and 6, cover the results, discussion and conclusion.

2. Relation Extraction

Relation extraction, compared to other NLP tasks is a relatively novel discipline. According to Hun et al. [9] the evolution of RE is roughly divided into three phases:

- **Pattern extraction models.** The first models rely on sentence analysis tools that identify syntactic elements in the text, whereupon pattern rules are automatically generated. Pattern rules are error-prone and therefore require a high level of intervention by human experts.
- **Statistical relation extraction models.** Along with the rest of NLP, RE has evolved to statistical models that significantly reduce the need for human intervention and provide better coverage of the task. Methods include feature-based, kernel-based, graphical, and embedding-model inspired methods.
- **Neural relation extraction models.** With the increasing usability and popularity of **neural networks (NN)**, especially with the use of GPU [10], neural methods overtook the field of RE. The first phase of use included various NN architectures that attempted to capture the semantics of text, such as recursive neural networks, convolutional neural

networks (CNN), attention-based neural networks, and recurrent neural networks (RNN), which dominated the field until the advent of the transformer architecture and approaches relying on pretrained (large) language models.

In the Table 1 below several examples of extracted relations are shown. Note that the underlined parts represent the entities detected by the NER model or jointly in the relation extraction model as of recent trends.

Sentence	Relation
Relation extraction (RE) is the subtask of information extraction.	subtask_of
There is a house way down in New Orleans	location_of
The town blossomed in the 18th and 19th centuries with the development of roads to the seaside and waterways along the Kupa River.	near_body_of_water
Though Kid A divided listeners, it was later named the best album of the decade by multiple outlets.	"named the best album of the decade by"
Now that the first person interface has become the design of choice for the industry, Id will need to find new innovations.	"has become"

Table 1
Examples of relations in sentences: First three rows depict RE with finite set of relations, while latter two represent RE without previously defined schema.

Upon closer inspection, we can notice similarities in nomenclature between the first three examples, and between the bottom two examples. The Table 1 previews two different understandings of the relation extraction task: the first approach (used in this work) considers finite set of relations (i.e. closed relation extraction) and the second approach refers to the much harder task of identifying relations without a strictly predefined template (i.e. open relation extraction or unsupervised relation extraction) [11]. The former is erratically also specified as the relation classification (RC). For further insight into the definition of relation extraction, the reader is encouraged to examine the research of Bassignana and Plank[12], where the definition of the RE task is revisited along with a comprehensive survey on RE datasets. Relation extraction, as done by the majority of the research community, is usually tackled in two main setups: **Pipeline approach** and **Joint entity and relation approach**.

2.1. Related Work on Neural Relation Extraction

In the pipeline approach, the NER and RE tasks are trained separately, therefore the RE model expects already extracted entities in the input text, which may be of lower quality, propagating the error. Although the pipeline approach suffers from error propagation, it is easy to implement and yields good results, as shown in the work of Nguyen et al. [13], Ale et al. [14], and Zhou et al. [15].

In joint entity and relation extraction, the model is trained to perform both tasks simultaneously while benefiting from the use of interrelated signals. This approach has attracted a lot of

attention in recent years and has provided new state-of-the-art results in various benchmarks. Some examples are the work of P. L. Huguet Cabot and R. Navigli [16], which define the RE task as seq2seq generation, and Tang et al. [17] which uses matrix-like interaction maps to effectively represent relations and NE together.

In the work of Beltagy et al. [18], the SciBERT model is presented along with exploration of various fine-tuning tasks. These include an approach similar to ours, a relation classification task on the SciERC [19] dataset with gold entities[1]. The extraction of sentence-level relations is also discussed in the work of Baek et al. [20], where RoBERTa [21] is trained on the TACRED [22] dataset with the Minority Class Attention Module (MCAM) to tackle the low-frequency relation problem. Most datasets for relation extraction, e.g., the datasets discussed in Section 3.3, rely on existing knowledge bases, e.g., Wikidata. RE and knowledge bases (e.g., knowledge graphs (KG)) are two interconnected domains with a mutually beneficial relationship. In this sense, it is possible to automatically construct graph-like structures to represent information (knowledge) extracted from unstructured texts. Similarly, RE is often used to improve KGs, as in the work of Pingle et al. [23], where RE is used to improve cybersecurity KG. Also, some research, such as the work of Yu et al. [24], Li et al. [25], and Luan et al. [19] preview the implementation of relation extraction for KG construction. In this work, we experiment with encoder-based models (BERT and SciBERT) for the relation extraction as the relation classification problem in the domain of science (i.e. scientific relation classification) to asses the influence of pretraining/finetuning corpora.

3. Methodology

3.1. BERT and SciBERT

When considering language modeling for use on downstream tasks (i.e. fine-tuning the pre-trained language model), it is important to find the right objective for language modeling. In this light, Devlin et al. with "*BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*" [4] point out the shortcomings of left-to-right language modeling[2], such as work of Radford et al. [26]. Decoders trained as left-to-right language models are limiting the context of a token only to tokens that precede it. Therefore for the task on hand, the encoder is the preferred part of the transformer architecture. The next paragraph briefly introduces transformer architecture and its connection with Generative Pretrained Transformer (GPT) and BERT models.

As shown in Figure 1, depicting the original transformer encoder-decoder architecture, two pertinent models were created based on this architecture, the encoder only BERT [4] and the pure decoder GPT [26]. In this work, we used only the encoder part of the architecture, extracting contextualized representations from sentences to classify the present relations.

The encoder of a transformer consists of an arbitrary number of encoder blocks. Each encoder block starts with a self-attention layer, more specifically a multi-head self-attention that further

[1]Gold entities refer to relation extraction setup where the model has information about entities, as opposed to a setup where the entities are also extracted by the model.

[2]Left-to-right language modeling objective is a task where an LM is trained to predict the next part of the sequence, given the previous parts.

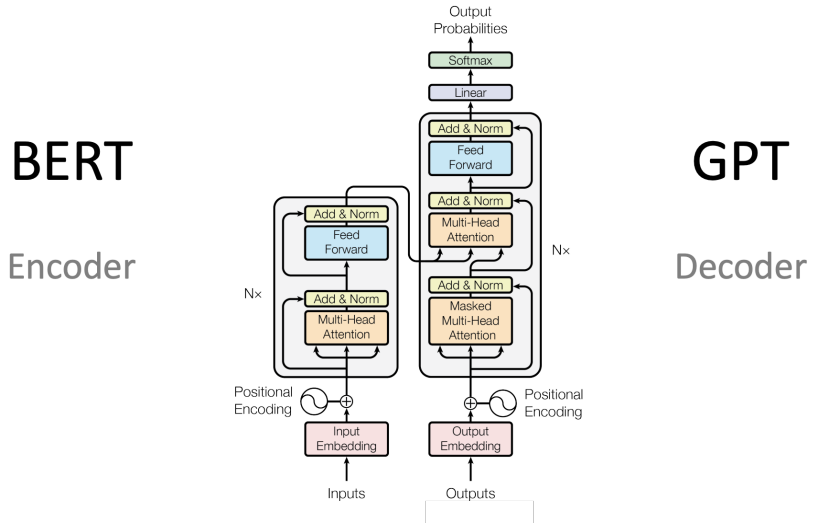

Figure 1: Transformer architecture: Architecture design used in the work of Vaswani et al. [27], consisting of a encoder and a decoder part, from which two models, BERT and GPT stem respectively. Adopted from work of Niklas Heidloff: *https://heidloff.net/article/foundation-models-transformers-bert-and-gpt/*

facilitates the ability to encode multiple relationships and nuances for each part of the input, i.e., **token**[3]. The output of the multi-head self-attention then proceeds through position-wise feed-forward network (FFN) consisting of a linear layer, ReLU, and another linear layer. After each of the steps (multi-head self-attention and position-wise FFN), the residual connection is added along with the layer normalization.

Following the original work of Vaswani et al. [27], BERT retains the encoder architecture with variations in the number of encoder blocks (L), hidden size (H), and self-attention heads (A). In the work, two main variants of BERT are presented: **BERT**$_{BASE}$ **(L=12, H=768, A=12)** and BERT$_{LARGE}$ (L=24, H=1024, A=16), the prior of which is used in this work. To enable BERT to handle a variety of downstream tasks, such as text classification, relation extraction, sentiment analysis, and question answering, two special tokens are used in the input and output representations:

- [CLS] token - First token of every sequence. The final hidden state corresponding to this token is used as aggregate sequence representation for classification tasks,
- [SEP] token - Sentence separator, in case the input consists of two sentences, e.g. for question answering task.

For further detail on the input and output representation of BERT models, the reader is advised to read the work of Devlin et al. [4].

Authors split BERT pretraining into two separate tasks, **masked language modeling (MLM)** and next sentence prediction (NSP). As argued by authors, it is possible to gain more useful

[3]Token can be considered as a useful semantic unit for processing, common tokens are subword, word, and sentence tokens.

information, i.e. get better-contextualized representations when a single token is contextualized by the rest of the sequence (sentence), as compared to left-to-right or right-to-left model[4]. It is not desirable for a standard LM to attend to "future" tokens (words) in the sentence, as the training for predicting the next word in the sentence becomes trivial. In this light, Devlin et al. propose a new objective, that hides (**masks**) only a single part of the sequence that needs to be predicted, calling it MLM (masked language model)[5]. BERT model was **pretrained on BookCorpus [28] and English Wikipedia**.

Following the same design as above, Beltagy et al. [18] train the BERT model on scientific corpora to support further scientific-based use-cases. Of multiple pretrained models yielded in this work, the **SciBERT** with new SciVocab[6] WordPiece vocabulary uncased[7] is used. All models presented in work of Beltagy et al. are pretrained on **random sample of 1.14M papers from Semantic Scholar** [30].

3.2. Relation Extraction Evaluation Metrics

Relation extraction and its associated named entity recognition task are relatively difficult to evaluate, especially when two tasks are trained in a joint scenario (sometimes referred to as end-to-end RE). Taillé et al. [31] discuss this problem further and address the common errors that occur even in state-of-the-art models for joint RE and NER tasks. In this work, the NER task is neglected, i.e., the model expects tagged inputs for further standard multiclass classification. To this end seven metrics are presented: Micro and Macro versions of precision, recall and F1-score, and the standard accuracy metric. These metrics are calculated from true positive (**TP**) and true negative (**TN**) values (TP-the number of times a model correctly predicts positive and TN-negative class), and false positive (**FP**) and false negative (**FN**) which count incorrectly predicted positive and negative classes respectively. An in-depth review of TP, TN, FP, and FN calculations on the multiclass classification is in Grandini et al. [32].

Accuracy, the metric that rewards models with the most correct predictions (TP and TN) is computed as: $accuracy = \frac{TP+TN}{TP+TN+FP+FN}$. This metric works desirably for balanced datasets, i.e. datasets that have a similar distribution of instances in all classes.

Precision is defined as the ratio of TP over the sum of TP and FP. Calculation on binary classification is straightforward: $precision = \frac{TP}{TP+FP}$. Precision can also be used on multiclass classification with two different approaches: for each class individually (observing binary classification for each class) and with averaging across all classes (micro and macro averaging). Of two, the latter is used both for precision and recall calculation, also extending to F1-score. Specifically, there are two standard approaches used for averaging along classes:

Macro precision is calculated for a selected class-positive, where all other classes are considered negative. After calculating precision for each class in this manner, per-class precision is macro averaged as:

[4] "*Models where every token can only attend to previous tokens in the self-attention layers of the Transformer*" - [4]

[5] To mitigate the problems regarding special [MASK] token that does not appear during fine-tuning of the LM, extra steps in the objective are added.

[6] Original BERT uses WordPiece-based [29] vocabulary consisting of 30,000 tokens.

[7] Uncased refers to the model trained on the lower-cased textual data.

$$precision_{MACRO} = \frac{precision_{C1} + precision_{C2} + ... + precision_{CN}}{N}$$

where N is the number of classes denoted by C1 to CN.

Micro precision is calculated for each class, by summing up all TP and FP values per class resulting in Total True Positive and a Total False Positive sums respectively. Based on these total sums, the micro precision is then calculated:

$$precision_{MICRO} = \frac{TP_1 + TP_2 + ... + TP_N}{TP_1 + FP_1 + TP_2 + FP_2 + ... + TP_N + FP_N}$$
$$= \frac{TP_{TOTAL}}{TP_{TOTAL} + FP_{TOTAL}}.$$

The **recall** is defined as the fraction of TP divided by the total number of sum of TP and FN: $recall = \frac{TP}{TP+FN}$. Working with multiple classes the **macro recall** is defined as:

$$recall_{MACRO} = \frac{recall_{C1} + recall_{C2} + ... + recall_{CN}}{N}$$

where N is the number of classes denoted by C1 to CN and **micro recall** is defined as:

$$recall_{MICRO} = \frac{TP_{TOTAL}}{TP_{TOTAL} + FN_{TOTAL}}$$

F1-score combines precision and recall of the model as their harmonic mean:

$$F1 = \frac{2(precision * recall)}{precision + recall}.$$

Since the metrics on which the F1-score relies have different approaches, mainly **micro** and **macro** averaging, the same classification is present for the F1-score. Thus, there exists a micro and macro version of the F1-score. F1-score is mainly used to compare models' performances while considering both, recall and precision.

3.3. Datasets

In this Section, the datasets that were used to train the BERT and SciBERT models to classify relations are discussed. First, each of the datasets is defined and an example is provided, then the work of Shimorina et al. [33] is presented. Based on this short survey of available datasets and their shortcomings, we construct the new dataset (**combo160**), containing the relations of interest in the scientific domain.

3.3.1. FewRel Dataset

Few-Shot Relation Classification Dataset (**FewRel**) by Han et al. [34] consists of 100 distinct relations, each accompanied by 700 instances. FewRel is created with the use of Wikipedia articles and Wikidata as a text corpus and the knowledge base respectively. After the gathering of the initial dataset through distant supervision, totaling in 122 relations and 122,000 instances, the dataset underwent human annotation that resulted in **70,000 instances**. A full list of relations, including their names and descriptions is available on the link[8] of the paper.

In Table 2 below the examples of the FewRel data are listed. ID here stands for the unique property (relation) in Wikidata (e.g., P26 represents the "spouse" relation between two entities), it is followed by the tokenized sentence, after which the details of **h**ead and **t**ail[9] entities are given, i.e., surface form (i.e. raw, textual representation of an entity), unique item ID, and indexes referring to the tokenized list.

ID	tokens	h	t
P26	[His, parents, are, Karl, von, Habsburg, and, Francesca, von, Habsburg, .]	"francesca von habsburg", Q1276954, [[7, 8, 9]]	"karl von habsburg", Q78515, [[3, 4, 5]]
P25	[Emmy, Acht\u00e9, was, the, mother, of, the, internationally, famous, opera, singers, Aino, Ackt\u00e9, and, Irma, Tervani, .]	"aino ackt\u00e9", Q259161, [[11, 12]]	"emmy acht\u00e9", Q4933685, [[0, 1]]

Table 2
FewRel examples: Head (**h**) and tail (**t**) are used interchangeably with subject and object annotation.

3.3.2. T-REx Dataset

Elsahar et al. [35] in "*T-REx: A Large Scale Alignment of Natural Language with Knowledge Base Triples*" address the problem of small RE datasets by utilizing Wikidata[10] and DBpedia [36], a dataset consisting of Wikipedia abstracts, to form a new dataset called T-REx. T-REx contains **11 million triple alignments** from 6.2 million sentences. In this case, the triple alignment refers to the process of mapping the extracted entities from the natural language sentence with the triple in knowledge graph (KG) or knowledge base (KB), to form a distantly supervised training instance. For brevity, the examples from the T-REx dataset are not included.

3.3.3. DocRED Dataset

Compared to the previous two datasets, which were primarily developed for extracting sentence-level (intra-sentence) relations, Yao et al. [37] create a DocRED dataset for extracting document-level (inter-sentence) relations. DocRED is based on a similar design to T-REx and FewRel,

[8]Not public due to details on the test data used for online evaluation. Available at: https://github.com/thunlp/FewRel/blob/master/data/pid2name.json
[9]Subject and object entities are sometimes considered head and tail entities in the literature.
[10]https://www.wikidata.org/

in that the main and only data source is the aforementioned combination of Wikidata and Wikipedia abstracts through distant supervision. It is important to note that the tedious process of human annotation of part of the data using crowdsourcing has made a significant contribution to the field. The result is 5,053 human-annotated documents with 40,276 sentences and a total of **96 distinct relations** and a large distantly supervised dataset of 101,873 documents with 828,115 sentences.

3.3.4. WikiFact Dataset

Goodrich et al. [38] explore new metrics for evaluating the factual accuracy of the generated text, primarily for the RE task. Similar to the previously presented datasets, under the distant supervision assumption [39], a new dataset based on Wikidata and Wikipedia, WikiFact, is created. WikiFact consists of two distinct parts based on the training objective, data for the relation classifier (relation extraction) and data for fact extraction (paragraph and sentence based). In this work, only the data for RE is used, which is 13 GB in total. Table 3 previews examples from the WikiFact dataset. Compared to the structure of the data in FewRel (Table 2), the authors mark the entities in untokenized sentences, which leaves room for task-specific preprocessing.

target	inputs	subject	object
P0	SUBJ{Art Nalls} was born in 1954 in OBJ{Alexandria}, Virginia just outside Washington, D.C. and grew up in that area.	Art Nalls	Alexandria
P0	SUBJ{Cerner} and executives at Girard agreed that Girard did not have adequate staff to manage the acquisition and OBJ{implementation} of the system.	Cerner	implementation

Table 3
WikiFact examples: Inputs (sentences) have annotated subject (SUBJ{ }) and object (OBJ{ }) spans.

3.3.5. Wiki20m Dataset

Comparable to the previously covered datasets, Wiki20m [9] utilizes distant supervision via Wikipedia and Wikidata. Wiki20 is originally constructed for bag-level relation extraction, a task that aims to extract relations from multiple sentences (i.e. bag-of-sentences). Wiki20m is the version of Wiki20 with a manually annotated test set. Each instance in the dataset resembles the structure of FewRel with tokenized sentence, head and tail data, and relation label.

3.3.6. WebRED Dataset

In "*WebRED: Effective Pretraining And Finetuning For Relation Extraction On The Web*" [40] Ormandi et al. point out generalization problems that concern Wikipedia-Wikidata[11] trained models, since the text on Wikipedia follows a certain structure and certain constraints. To mitigate generalization problems, a set of web domains with high linguistic quality and factually

[11]Relation extraction datasets constructed from Wikipedia and Wikidata through distant supervision.

correct content were ranked by human annotators. From the selected web domains, the large corpus was created and linked to Wikidata triples in a process similar to previous datasets. Then, the part of the data was subjected to a human annotation process similar to DocRED. The result is a two-part dataset with **523 unique (Wikidata) relations**, with the **117,717 human annotated** and **199,786,781 weakly supervised** examples.

3.3.7. Relation Extraction Database Based on Wikidata

While the datasets presented were created for seemingly different tasks, they share the knowledge base for defining and extracting relations, Wikidata. With this property in mind, Shimorina et al. [33] design a sentence-level RE database based on the aforementioned datasets (FewRel, T-REx, DocRED, WikiFact, Wiki20m, WebRED). First, the datasets are preprocessed, including deduplication and extraction of designated entities from unwanted formats. Then, the instances (sentences) or rather, relations are uniformly labeled, with OBJ and SUBJ representing the entities involved in a given relation. Results are **47,390,557** instances across **1,022 unique relations**, including "**P0**" (no relation) and " **NA** " (unknown relation). This database is used to maximize the number of relation types that are of interest to us and allows us to fine-tune the models to a broader range of scientific relationships.

3.4. Combo160 Dataset

Since this work is concerned with the domain of science, it is desirable to train the models on data that exhibit scientific relationships. To obtain such relations, the Wikidata list of properties in the science domain[12] was examined. Particularly, relations inherent on chemistry, physics, biology, mathematics, geology, and astronomy, are considered resulting in 341 applicable relations. Based on these relations, the database created by Shimorina et al. [33] was queried to filter out the ones of interest, resulting in a new dataset with 176 relations. The number of instances for each relation is limited to 10,000 to prevent further increase in disproportionality of the data, since the majority of the examples (66%) belong to the classes "P0" (no relation) and "NA" (unknown relation). After the preprocessing (cutting of the relation types with frequencies less then three and reshaping to a desirable input for OpenNRE [41] Toolkit), the dataset resulted in **161 unique relations**, containing **301,062 examples** in total. Hence we name the dataset combo160. In the Table 4 below, a summary of datasets information is given with respect to the work of Shimorina et al. [33]. Please note that statistics presented in Table 4 refer to sentence-level RE and are calculated after preprocessing procedures, such as deduplication.
Table 4 shows the total number of instances (# instances), relations (# R), and percentage of negative relations (% neg.) in each of the datasets. Relations "P0" (no relation) and "NA" (unknown relation) are considered negative in this context. Negative examples can be excluded if a binary classification (yes/no relation between entities) is performed before relation classification (RC). Essentially, the RE is further divided into two approaches: the first, binary classification before RC and the second, RC with the classes "no relation" and/or "unknown" relation classes. Here we treat "P0" and "NA" as regular relations although, usually, most entity co-occurrences are either undefined or not-relation, which is the main rationale behind high percentage of

[12]https://www.wikidata.org/wiki/Wikidata:List_of_properties/science

dataset	# instances	# R	% neg.
FewRel	56,000	80	0%
T-REx	12,081,023	652	0%
DocRED	778,914	96	0%
WikiFact	33,628,338	934	92%
Wiki20m	738,463	81	60%
WebRED	107,819	385	54%
Unified database	47,390,557	1,022	66%
Combo160	**301,062**	**161**	**6.6%**

Table 4
Dataset summary: with the number of instances, the number of relations (R), and % of negative relations.

negative examples in datasets such as WikiFact, Wiki20m, WebRED and the unified database [33]. The exploration of the impact of the negative examples ratio is left for future work. Hence, to wrap up Combo160 dataset is created through filtering and preprocessing in order to obtain the dataset fit for the relation extraction classification training for the domain of science.

4. Experimental Setup

The combo160 dataset is exported to csv format for further preprocessing using python with appropriate libraries (pandas [42, 43], scikit-learn [44]). First, the empty values and notation SUBJ and OBJ are cleaned up in the records. Clean sentences are then tokenized using the BasicTokenizer implemented in OpenNRE to make the input conform to the Toolkit standards. Following tokenization, low frequency relations (less than three times) are removed to allow splitting between training, test, and validation, with each relation occurring at least once per set. Using scikit-learn function `train_test_split()` with stratification based on relation ID (e.g., P1234) that corresponds to the class label, the dataset is split into 80:15:5 ratio to train, test, and validation subsets respectively.

4.1. OpenNRE Toolkit

OpenNRE is an open source and extensible toolkit that provides a unified framework for implementing relation extraction models, introduced with the work of Han et al. [41] "*OpenNRE: An Open and Extensible Toolkit for Neural Relation Extraction*". Toolkit enables RE extraction in a specific setup, consistent comparison, re-implementation, variation, deployment, and evaluation over different tasks.

The OpenNRE toolkit adresses the problem of code reusability as well, by providing extensible base implementations for most tasks that precede or follow RE, such as tokenization (word and subword level), common neural layers, encoder module, data processing, model training and evaluation. OpenNRE allows three approaches to RE:

- **Sentence-level RE** (RE from sentence, i.e. only the existence of relations inside a single sentence is assumed),

- **Bag-level RE** (RE from multiple sentences, i.e. relations can exist across multiple sentences that appear consecutively),
- **Document-level RE** (RE from the whole document).

In this work, OpenNRE is used due to its extensibility and ease of use, as the toolkit contains a complete procedure for training the BERT model for RE following the work of Soares et al. [5].

4.2. Training Setup

Soares et al. with "*Matching the Blanks: Distributional Similarity for Relation Learning*" investigate the capabilities of BERT in extracting relations as the classification problem and then train the BERT model on new objective specifically designed to capture relation representation, named **matching the blanks (MTB)**. In this work, we focus on the relation classification problem with fine-tuning procedures. A sequence of tokens (x), e.g. words, is defined as $x = [x_0, x_1, ..., x_n]$, where, similarly to original setup, $x_0 = [CLS]$ and $x_n = [SEP]$ are special start and end markers. Moreover, let $s_1 = (i, j)$ and $s_2 = (k, l)$ be pairs of integers such that $0 < i < j - 1, j < k, k \leq l - 1$, and $l \leq n$. Here, relation (r) is represented as $r = (x, s_1, s_2)$, where s_i represents the entity mentions in the sentence.

For this work, BERT models are fine-tuned for RC with **entity marker** tokens that incorporate s_1 and s_2 entity spans into the input via special tokens: [E1$_{start}$], [E1$_{end}$], [E2$_{start}$], and [E2$_{end}$]. Resulting in an augmented sequence ($\tilde{x}$) as:

$$\tilde{x} = \big[x_0, ..., [E1_{start}], x_i, ..., x_{j-1}, [E1_{end}], ..., [E2_{start}], x_k, ..., x_{l-1}, [E2_{end}], ..., x_n\big].$$

One exemplary sentence depicts the procedure:
If sentence (x) is:

$$x = \big[\textbf{\textit{he}}, is, seen, as, one, of, the, founders, of, modern, \textbf{\textit{archeology}}, in, czech, lands, "."\big];$$

and entity spans s_1 and s_2 are:

$$\boldsymbol{s_1} = (0, 0) \text{ and } \boldsymbol{s_2} = (10, 10);$$

then augmented sentence ($\tilde{x}$) is:

$$\tilde{x} = \big[\textbf{\textit{[E1}}_{\textbf{\textit{start}}}\textbf{\textit{]}}, he, \textbf{\textit{[E1}}_{\textbf{\textit{end}}}\textbf{\textit{]}}, is, ..., modern, \textbf{\textit{[E2}}_{\textbf{\textit{start}}}\textbf{\textit{]}}, archeology, \textbf{\textit{[E2}}_{\textbf{\textit{end}}}\textbf{\textit{]}}, in, czech, lands, "."\big].$$

With this set-up, two models BERT and SciBERT are fine-tuned with the combo160 dataset on the RC task with parameters set up as elaborated in Table 5. To enable training for the task of multi-label classification, outputs of the encoders (BERT and SciBERT) are forwarded to a neural network consisting of a linear layer, dropout layer, and softmax layer to output the probability distribution over the number of classes.
Each of the two models (BERT and SciBERT) was trained for 3 epochs with a total of ~ 12 hours of training on an Ubuntu 20.04 machine with a single NVIDIA GeForce GTX 1050 Mobile (3GB) GPU and Intel Core i7-7700HQ CPU. The results of both BERT$_{base}$ uncased and SciBERT SciVocab uncased models are compared following an identical training set-up to support the arguments discussed. It is important to note that BERT$_{base}$ and SciBERT have the same architecture and differ only in the corpora used to pretrain the model.

Model	$BERT_{base}$ uncased / SciBERT SciVocab uncased
Dataset	Combo160
Pooler	Entity
Entity masker	No
Batch size	8
Learning rate	2e-5
Maximum length	128
Maximum epochs	3
Seed	42
Optimizer	Adamw [45]

Table 5
Training parameters set-up

5. Results

In this Section we present obtained results. First, the inference of two models (BERT and SciBERT) is evaluated with standard multiclass classification metrics such as accuracy, precision and F1-score with micro and macro averaging. Second, the results are discussed.

The distribution of relations in the training data is in Figure 2. Comparing the top 20 relations (by the number of instances) in both datasets, we obtain a significant overlap with a difference in only one relation P706 (in train) versus P279 (in test). The top 20 set includes relations such as: *location* (P276), *father* (P22), *sibling* (P3373), *instance of* (P31), *located in the administrative territorial entity* (P131), *owned by* (P127), *field of work* (P101), and of course the negative classes *unknown* (NA /PNAN) and *no relation* (P0). The above relations are more general as compared to the scientifically specific ones. Hence, relations such as *chromosome* (P1057), *monomer of* (P4599), *pathogen transmission process* (P1060), *lymphatic drainage* (P2288), *research site* (P6153), *decreased expression in* (P1910), and *inflorescence* (P3739), are more prone to scientific domain. According to their occurrence in Combo160 dataset, they appear at the bottom 20 places. Arguably, it is expected to have a higher overall occurrence of ”general” relations, compared to ”scientific domain” relations in the dataset constructed mainly from Wikipedia as the source. With this in mind, next, we present the results of two trained models BERT and SciBERT in Table 6.

	BERT	SciBERT
Accuracy	**0.99893**	0.99877
Micro precision	**0.92370**	0.91423
Micro recall	**0.90931**	0.89514
Micro F1-score	**0.91645**	0.90458
Macro precision	0.74523	**0.77450**
Macro recall	0.70157	**0.72637**
Macro F1-score	0.71641	**0.74119**

Table 6
The results for BERT and SciBERT: in terms of accuracy, micro and macro averaged precision, recall and F1-score.

The first metric discussed, **accuracy**, yields comparable results for both models, with a performance difference of $1.54980406 \cdot 10^{-4}$ in favor of BERT. If we revisit the definition of accuracy (Section 3.2), it becomes clear that this accuracy only rewards correct predictions and attenuates the proportion of correctness per class. While intuitive, this metric yields biased results when the dataset is unbalanced, as is the case with Combo160 (Figure 2). Meaning, the model that more accurately predicts the dominant classes (presumably BERT) could have a better result according to the accuracy metric exclusively. To discuss the matter further, we explore micro and macro averaging. According to Grandini et al. [32] the idea of micro-averaging is to consider all the units together, without considering disproportion between classes.

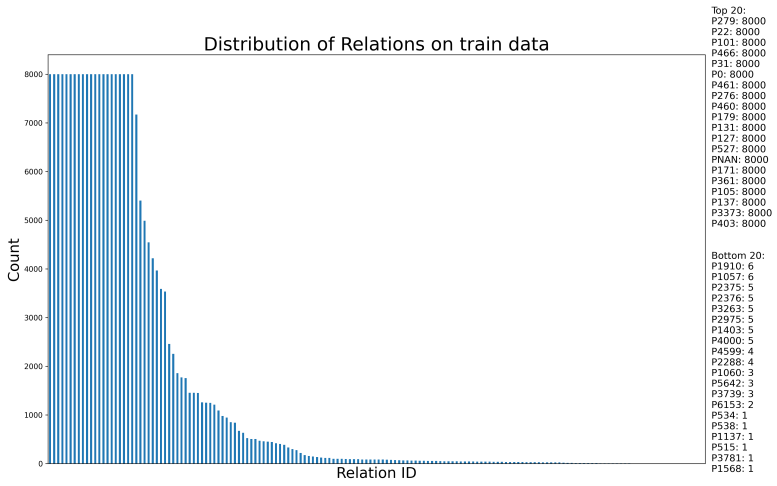

Figure 2: Relation distribution in train data: with Top 20 and Bottom 20 relation IDs

Similar to accuracy, BERT shows negligibly better performance in **micro-averaged precision** with $9.465608607 \cdot 10^{-3}$ advantage. This advantage is also to be expected since micro-averaging first sums all units, i.e., TP and FP, and then calculates precision based on the sums, again neglecting the inequality of classes in the dataset. The comparison of **micro-averaged recall** yields similar results with a more significant advantage of $1.417546431 \cdot 10^{-2}$ in the performance of BERT. **Micro F1-score**, as a harmonic mean between (micro) precision and recall, gives a better assessment of the overall performance of the model than the two previously mentioned metrics. Looking at the results of micro-averaged F1-score, a repeating trend is exhibit, with BERT performing with a lead of $1.186484641 \cdot 10^{-2}$. Thus, based on the micro-averaged metrics, it can be concluded that BERT performs better due to better results of classification in dominant classes (i.e. general ones) as compared to SciBERT.

Looking at the macro-averaged metrics, there is a significant decline in performance scores in general. This behaviour is stemming from the unbalanced nature of dataset, as macro-averaged metrics tend to neglect the correlation between class size and overall scores [32]. This means that macro-averaged metrics, by virtue of their calculations, have the desirable side effect of giving equal importance to each class (regardless of the number of instances in the class). Let

us now consider the macro-averaged scores of the models. A significant difference is exhibited when comparing **macro precision** of two models, as SciBERT achieves $2.92693784 \cdot 10^{-2}$ better score. This means that the SciBERT model classifies the relations more confidently. Similar results are also manifested in the **macro recall** and **macro F1-score**.

Two important conclusions can be drawn from the analysis of the provided metrics measured on SciBERT and BERT models in classifying relations:

- The BERT model performs better on more dominant classes overall (better results on micro-averaged metrics and accuracy) and, in particular, correctly classifies examples of dominant relations (greater macro recall).
- The SciBERT model is competitive with BERT when it comes to more dominant relations (general ones), but achieves better results on less dominant classes (specific to the domain of science), having better overall results on macro-averaged metrics.

As noted at the beginning of this section, the dominant (frequent) relations tend to be more ”general”, while the less dominant (infrequent) relations tend to be more ”scientific”. Given this, it can be concluded that SciBERT is a better fit to extract relations from the domain of science.

6. Conclusion

This research is concerned with the training (fine-tuning) of sentence-level relation extraction models BERT and SciBERT. The relation classification is modeled as the multi-class classification problem. For this task, the pretrained transformer-encoder models BERT and SciBERT are used and compared to observe the effects of the usage of different pretraining corpora.

Relevant datasets suitable for the relation extraction task are explored, resulting in the construction of a new dataset, **combo160** fit for the relation extraction in the domain of science. Two models, SciBERT and BERT, are trained on this newly constructed dataset with 161 relation types - combo160.

Finally, the results of the two models are presented and compared in order to draw conclusions about the pretraining corpora used. Using relevant metrics for classification tasks, such as accuracy and (micro- and macro-) averaged precision, recall and F1-score, it is shown that BERT performs marginally better on accuracy and micro-averaged metrics. This implies better performance on more dominant classes (which turn out to be less ”scientific domain” and more ”general”), while SciBERT model outperforms the BERT model when it comes to relations specific to the scientific domain, implied by better results on the macro-averaged metrics that account for class disproportions in the calculation. To conclude, we list two important contributions of this work:

- Creation (selection) of the new dataset for the relations in the domain of science **combo160**;
- Two fine-tuned models BERT and SciBERT treined for RE, are available for use through **OpenNRE toolkit** [41];

To further support the conclusion, the issue of defining the term ”scientific” relationship needs to be addressed in more detail. In addition, other models with different pretraining objectives and

architectures should be explored and evaluated against the combo160 dataset. The sole dataset, combo160, should be further analyzed to reduce and find the optimal and necessary number of relations. This and the study of the effects of relation distribution (especially negative relations) are the subject of future work. These results serve as the initial step in relation extraction which will be fine-tuned to specific scientific subdisciplines like physics or chemistry, aiming to proceed to the construction of a domain-specific knowledge graph.

7. Acknowledgments

This work has been partialy supported by the University of Rijeka under the project number uniri-drustv-18-20. AP is supported by Croatian Science Foundation under the project DOK-2021-02.

References

[1] J. Eisenstein, Introduction to Natural Language Processing (Adaptive Computation and Machine Learning series), MIT Press, 2019. URL: https://mitpress.mit.edu/books/introduction-natural-language-processing.

[2] S. Sarawagi, Information extraction, Foundations and Trends® in Databases 1 (2008) 261–377. URL: http://dx.doi.org/10.1561/1900000003. doi:10.1561/1900000003.

[3] S. Pawar, G. K. Palshikar, P. Bhattacharyya, Relation extraction : A survey, 2017. arXiv:1712.05191.

[4] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. arXiv:1810.04805.

[5] L. Baldini Soares, N. FitzGerald, J. Ling, T. Kwiatkowski, Matching the blanks: Distributional similarity for relation learning, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 2895–2905. URL: https://aclanthology.org/P19-1279. doi:10.18653/v1/P19-1279.

[6] X. Li, X. Sun, Y. Meng, J. Liang, F. Wu, J. Li, Dice loss for data-imbalanced NLP tasks, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 465–476. URL: https://aclanthology.org/2020.acl-main.45. doi:10.18653/v1/2020.acl-main.45.

[7] L. Miculicich, J. Henderson, Graph refinement for coreference resolution, in: Findings of the Association for Computational Linguistics: ACL 2022, Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 2732–2742. URL: https://aclanthology.org/2022.findings-acl.215. doi:10.18653/v1/2022.findings-acl.215.

[8] A. Poleksić, Relation extraction with deep learning methods, 2023. URL: https://urn.nsk.hr/urn:nbn:hr:195:836831, repository of Faculty of Informatics and Digital technologies, University of Rijeka.

[9] X. Han, T. Gao, Y. Lin, H. Peng, Y. Yang, C. Xiao, Z. Liu, P. Li, J. Zhou, M. Sun, More data, more relations, more context and more openness: A review and outlook for relation extraction, in: Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, Association for Computational Linguistics, Suzhou, China, 2020, pp. 745–758. URL: https://aclanthology.org/2020.aacl-main.75.

[10] A. Poleksić, Paralelizacija izračuna dubokih neuralnih mreža na grafičkim procesorima, 2021. URL: https://urn.nsk.hr/urn:nbn:hr:195:174813.

[11] R. Wu, Y. Yao, X. Han, R. Xie, Z. Liu, F. Lin, L. Lin, M. Sun, Open relation extraction: Relational knowledge transfer from supervised data to unsupervised data, 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), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 219–228. URL: https://aclanthology.org/D19-1021. doi:10.18653/v1/D19-1021.

[12] E. Bassignana, B. Plank, What do you mean by relation extraction? a survey on datasets and study on scientific relation classification, 2022. arXiv:2204.13516.

[13] T. H. Nguyen, R. Grishman, Relation extraction: Perspective from convolutional neural

networks, in: Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, Association for Computational Linguistics, Denver, Colorado, 2015, pp. 39–48. URL: https://aclanthology.org/W15-1506. doi:10.3115/v1/W15-1506.

[14] C. Alt, M. Hübner, L. Hennig, Improving relation extraction by pre-trained language representations, 2019. arXiv:1906.03088.

[15] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, B. Xu, Attention-based bidirectional long short-term memory networks for relation classification, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Berlin, Germany, 2016, pp. 207–212. URL: https://aclanthology.org/P16-2034. doi:10.18653/v1/P16-2034.

[16] P.-L. Huguet Cabot, R. Navigli, REBEL: Relation extraction by end-to-end language generation, in: Findings of the Association for Computational Linguistics: EMNLP 2021, Association for Computational Linguistics, Punta Cana, Dominican Republic, 2021, pp. 2370–2381. URL: https://aclanthology.org/2021.findings-emnlp.204. doi:10.18653/v1/2021.findings-emnlp.204.

[17] W. Tang, B. Xu, Y. Zhao, Z. Mao, Y. Liu, Y. Liao, H. Xie, UniRel: Unified representation and interaction for joint relational triple extraction, in: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022, pp. 7087–7099. URL: https://aclanthology.org/2022.emnlp-main.477.

[18] I. Beltagy, K. Lo, A. Cohan, SciBERT: A pretrained language model for scientific text, 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), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3615–3620. URL: https://aclanthology.org/D19-1371. doi:10.18653/v1/D19-1371.

[19] Y. Luan, L. He, M. Ostendorf, H. Hajishirzi, Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 3219–3232. URL: https://aclanthology.org/D18-1360. doi:10.18653/v1/D18-1360.

[20] H.-R. Baek, Y.-S. Choi, Enhancing targeted minority class prediction in sentence-level relation extraction, Sensors 22 (2022). URL: https://www.mdpi.com/1424-8220/22/13/4911. doi:10.3390/s22134911.

[21] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, 2019. arXiv:1907.11692.

[22] Y. Zhang, V. Zhong, D. Chen, G. Angeli, C. D. Manning, Position-aware attention and supervised data improve slot filling, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Copenhagen, Denmark, 2017, pp. 35–45. URL: https://aclanthology.org/D17-1004. doi:10.18653/v1/D17-1004.

[23] A. Pingle, A. Piplai, S. Mittal, A. Joshi, J. Holt, R. Zak, Relext: Relation extraction using deep learning approaches for cybersecurity knowledge graph improvement, in: Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM '19, Association for Computing Machinery, New York, NY, USA, 2020, p.

879–886. URL: https://doi.org/10.1145/3341161.3343519. doi:10.1145/3341161.3343519.
[24] H. Yu, H. Li, D. Mao, Q. Cai, A relationship extraction method for domain knowledge graph construction, World Wide Web 23 (2020) 735–753. URL: https://doi.org/10.1007/s11280-019-00765-y. doi:10.1007/s11280-019-00765-y.
[25] F.-L. Li, H. Chen, G. Xu, T. Qiu, F. Ji, J. Zhang, H. Chen, Alimekg: Domain knowledge graph construction and application in e-commerce, CIKM '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 2581–2588. URL: https://doi.org/10.1145/3340531.3412685. doi:10.1145/3340531.3412685.
[26] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, Improving language understanding by generative pre-training (2018).
[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, CoRR abs/1706.03762 (2017). URL: http://arxiv.org/abs/1706.03762. arXiv:1706.03762.
[28] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, S. Fidler, Aligning books and movies: Towards story-like visual explanations by watching movies and reading books, 2015. arXiv:1506.06724.
[29] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Łukasz Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, J. Dean, Google's neural machine translation system: Bridging the gap between human and machine translation, 2016. arXiv:1609.08144.
[30] W. Ammar, D. Groeneveld, C. Bhagavatula, I. Beltagy, M. Crawford, D. Downey, J. Dunkelberger, A. Elgohary, S. Feldman, V. Ha, R. Kinney, S. Kohlmeier, K. Lo, T. Murray, H.-H. Ooi, M. Peters, J. Power, S. Skjonsberg, L. L. Wang, C. Wilhelm, Z. Yuan, M. van Zuylen, O. Etzioni, Construction of the literature graph in semantic scholar, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers), Association for Computational Linguistics, New Orleans - Louisiana, 2018, pp. 84–91. URL: https://aclanthology.org/N18-3011. doi:10.18653/v1/N18-3011.
[31] B. Taillé, V. Guigue, G. Scoutheeten, P. Gallinari, Let's Stop Incorrect Comparisons in End-to-end Relation Extraction!, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 3689–3701. URL: https://aclanthology.org/2020.emnlp-main.301. doi:10.18653/v1/2020.emnlp-main.301.
[32] M. Grandini, E. Bagli, G. Visani, Metrics for multi-class classification: an overview, 2020. arXiv:2008.05756.
[33] A. Shimorina, J. Heinecke, F. Herledan, Knowledge extraction from texts based on Wikidata, in: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track, Association for Computational Linguistics, Hybrid: Seattle, Washington + Online, 2022, pp. 297–304. URL: https://aclanthology.org/2022.naacl-industry.33. doi:10.18653/v1/2022.naacl-industry.33.
[34] X. Han, H. Zhu, P. Yu, Z. Wang, Y. Yao, Z. Liu, M. Sun, FewRel: A large-scale super-

vised few-shot relation classification dataset with state-of-the-art evaluation, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 4803–4809. URL: https://aclanthology.org/D18-1514. doi:10.18653/v1/D18-1514.

[35] H. Elsahar, P. Vougiouklis, A. Remaci, C. Gravier, J. Hare, F. Laforest, E. Simperl, T-REx: A large scale alignment of natural language with knowledge base triples, in: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), European Language Resources Association (ELRA), Miyazaki, Japan, 2018. URL: https://aclanthology.org/L18-1544.

[36] M. Brümmer, M. Dojchinovski, S. Hellmann, DBpedia abstracts: A large-scale, open, multilingual NLP training corpus, in: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), European Language Resources Association (ELRA), Portorož, Slovenia, 2016, pp. 3339–3343. URL: https://aclanthology.org/L16-1532.

[37] Y. Yao, D. Ye, P. Li, X. Han, Y. Lin, Z. Liu, Z. Liu, L. Huang, J. Zhou, M. Sun, DocRED: A large-scale document-level relation extraction dataset, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 764–777. URL: https://aclanthology.org/P19-1074. doi:10.18653/v1/P19-1074.

[38] B. Goodrich, V. Rao, P. J. Liu, M. Saleh, Assessing the factual accuracy of generated text, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 166–175. URL: https://doi.org/10.1145/3292500.3330955. doi:10.1145/3292500.3330955.

[39] M. Mintz, S. Bills, R. Snow, D. Jurafsky, Distant supervision for relation extraction without labeled data, in: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, Association for Computational Linguistics, Suntec, Singapore, 2009, pp. 1003–1011. URL: https://aclanthology.org/P09-1113.

[40] R. Ormandi, M. Saleh, E. Winter, V. Rao, Webred: Effective pretraining and finetuning for relation extraction on the web, 2021. arXiv:2102.09681.

[41] X. Han, T. Gao, Y. Yao, D. Ye, Z. Liu, M. Sun, OpenNRE: An open and extensible toolkit for neural relation extraction, 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): System Demonstrations, Association for Computational Linguistics, Hong Kong, China, 2019, pp. 169–174. URL: https://aclanthology.org/D19-3029. doi:10.18653/v1/D19-3029.

[42] T. pandas development team, pandas-dev/pandas: Pandas, 2020. URL: https://doi.org/10.5281/zenodo.3509134. doi:10.5281/zenodo.3509134.

[43] Wes McKinney, Data Structures for Statistical Computing in Python, in: Stéfan van der Walt, Jarrod Millman (Eds.), Proceedings of the 9th Python in Science Conference, 2010, pp. 56 – 61. doi:10.25080/Majora-92bf1922-00a.

[44] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine

Learning Research 12 (2011) 2825–2830.
[45] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, 2019. arXiv:1711.05101.