

Applying Multi-Task Reading Comprehension in Acronym Disambiguation

Yunpeng Tai¹, Xiaoyu Zhang² and Xuefeng Xi (Corresponding Author)¹

¹Suzhou University of Science and Technology

²Shanghai Etrump Information Technology Co.,Ltd

Abstract

Acronym Disambiguation (AD) task is designed to find the exact expansion of the acronym in a given sentence. Since little work has been done in a Machine Reading Comprehension (MRC) way, this paper presents a novel model which leverages the advantages of both MRC and sequence tagging. First, AD is regarded as a multi-choice task and all the candidate expansions are options. We design useful question-answer pairs where Question can be seen as the combination of sentence and acronym while Context consists of the candidate expansions. Second, we apply adversarial learning (i.e. FGM) and normalization methods such as Gradient Centralization (GC) to further improve the robustness and generalization of the model. Third, the final answer is jointly predicted by two tasks which can enhance model's understanding towards AD. Besides, the model also infers the test set to construct pseudo-labelling set to make the most of data. The model we put forward provides a novel way to handle AD and the performance can be competitive.

Keywords

Acronym Disambiguation, Machine Reading Comprehension, Sequence Tagging, Multi-Task Learning, Adversarial Learning, Gradient Centralization

1. Introduction

Acronyms are built in part from the first letters of word components and pronounced like a word (e.g. NASA). Due to their convenience, acronyms are of widespread use in many scenarios where long words show frequently. For instance, an original sentence is, "Here we present a non-linear method based on a deep convolutional neural network and this convolutional neural network is quite powerful". Undoubtedly, this sentence is long and tedious. By replacing the initial long part with the acronym, the sentence can be brief and explicit. "Here we present a non-linear method based on a deep CNN and this CNN is quite powerful".

Although using acronyms in documents seems like a favorable choice, people who don't know much about a specific field can suffer from the ambiguity of acronyms. Therefore, it is beneficial to figure out the relationship between acronyms and their correct expansions for the further aim of eliminating the ambiguity of the acronym. However, given that acronyms are widely used in considerable fields, it is hard for people to clarify the real meanings of acronyms one by one. Thus, it is necessary to build a model which can automatically find the accurate expansions of acronyms used in documents.

As shown in Table 1, each instance in Acronym Disambiguation (AD) task includes a sentence which contains

Table 1

An example of acronym disambiguation task

Sentence	A Maximum Entropy Approach to NERn.
Acronym	NER
Candidate	Named Entity Recognito, ...
Label	Named Entity Recognitio


an acronym, the specific acronym part, the candidate expansions of a given acronym and its true label [1]. For the sentence in table 1, we need to choose appropriate expansion for the acronym from the candidate expansions (e.g. "Named Entity Recognitio", "named entity recognition", "Named Entity Recognition", "named entity taggers", "nition", "named entity recogniser", "Named Entity Recognizer", "Name Entity Recognizer", "Named entity recognition"). Understanding so many acronyms in different scenarios is still hard for someone who is not a native speaker of English let alone the poor machine.

Consequently, AD can be seen as a classification task. At the beginning, much attention is paid on the rule-based methods [2], [3] and they do work. For example, an inexact pattern matching algorithm is proposed and play a role in the past [4]. Due to the complexity of different acronyms' meanings in various scenarios, rule-based methods can always fail to catch the subtle relationship between an acronym and the according expansion [5]. Thus, an increased number of research shifts focus to exploiting the contextualized information. Based on lexical knowledge, the method computes the similarity between the acronym and words near it [6]. On the other hand, unsupervised methods are put forward to break the limit

Proceedings of the Workshop on Scientific Document Understanding (SDU 2022), Remote, 2022.

✉ yunpengtai.typ@gmail.com (Y. Tai); zhangxiaoyu@etrump.net (X. Zhang); xfxi@usts.edu.cn (X. X. (. Author))

© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

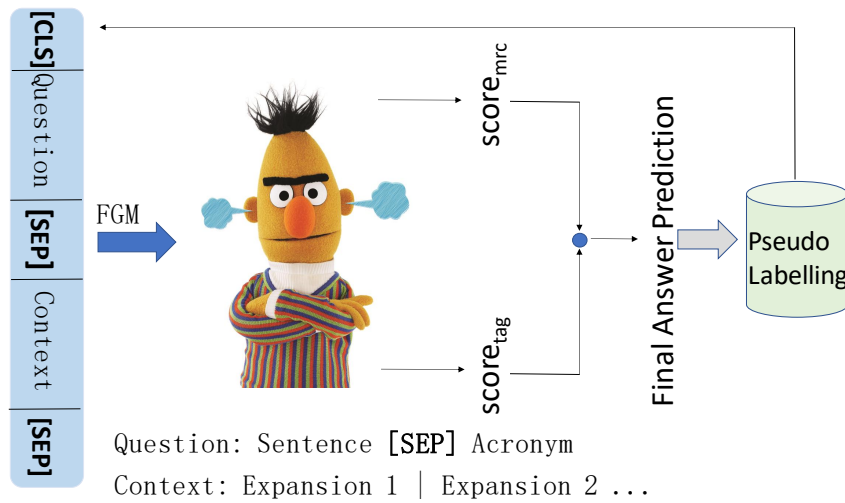


Figure 1: The left object represents the input of the model where Question is “Context” : sentence [SEP] “Acronym”: acronym. Note that “Acronym” and “Context” are the prompts. Context is all the candidate expansions split by “|”: “Option” : Expansion 1 | Expansion 2 ... where “Option” is also a prompt. FGM is the acronym of Fast Gradient Methods. Interestingly, the middle character is “BERT” from the Sesame Street (<https://www.keywordbasket.com>) and it stands for our base model. Our final answer prediction is produced by jointly training sequence tagging and reading comprehension. By means of inferring the test set, we can construct pseudo-labelling set and use it in training the model to realize full potential of all the data.

of annotated data. Via clustering word embedding [7], [8] into different groups, machine can learn how to distinguish one expansion from another. Each group just represents an expansion [9].

Given that supervised methods always have a better performance [10], more researchers tend to apply semi-supervised approaches [11] to further combine both advantages.

Over the past few years, pre-training models such as BERT [12] have already been proven to be powerful. BERT firstly masks the words in contexts at a certain possibility like “Beijing is the caption of [MASK] (China)” and then predicts the masked part to gain the representations of the corpus. Also, BERT is pre-trained at a binarized next sentence prediction task. When it is pre-trained done, BERT is finetuned at the annotated data. By fully making use of the unlabeled and annotated data, BERT outperforms all the models before it at many tasks on GLUE [13]. Afterwards, a large number of BERT’s variants come out RoBerta [14], ERNIE [15], SpanBERT [16], etc.

BERT_{LARGE} is our base model (L=24, H=1024, A=16, Total Parameters=340M). Different from other BERT models, this model is pre-trained by applying Whole Word Masking technique and then fine-tuned on the SQuAD dataset. Our whole model architecture is in Fig 1. To let the model exploit the relationships between the acronym

and sentence, we design the question as “Context” : sentence [SEP] “Acronym”: acronym. Note that the lower-case word just represents the instance in the dataset. Context is composed of all the candidate expansions separated by “|” which just highlights different expansions to make the model easy to learn the differences. Eventually, our model is jointly trained on two tasks: sequence tagging and machine reading comprehension. The result of inferring the test set can be used to construct the pseudo-labelling set to make full use of all the data to further improve the model’s performances.

The main contributions of this paper are summarized as 4 points:

- We introduce Acronym Disambiguation to Reading Comprehension Task naturally and observe something interesting about the components of question-answer pairs.
- To the best of our knowledge, we should be the first to train two tasks: sequence tagging and reading comprehension jointly on AD.
- Our end-to-end model does not need any extra operations and it is environmentally friendly compared to the ensemble of many models.
- Adversarial training is smoothly combined with Gradient Centralization to improve the performance.

This paper is organized as follows: Related work is included in Section 2. Then comes with model structure and experiment which has 4 subsections: The task, fine-tuning results and training details. And the last two sections are the conclusion and references.

2. Related Work

2.1. Adversarial Training

Since neural networks are fragile and vulnerable to perturbations, adversarial training is a good way to enhance model’s robustness by training the model on extra adversarial examples. Based on the observation that the direction of the perturbation (i.e. the gradient) matters most, the Fast Gradient Sign Method (FGSM) is originally yielded to produce adversarial examples [17]. Afterwards, the Fast Gradient Method (FGM) is added on the word embedding to improve the generalization of the model [18]. Different from them, the Projected Gradient Descent (PGD) does a range of attacks on the model and can map the perturbation to a specified range every time [19]. It is obvious that PGD performs better at the cost of high computation complexity. By restricting most of the forward and back propagation within the first layer during the adversarial training, YOPO reduces the cost of computation [20]. FreeAT recycles the gradient information when updating the model parameters to cut down the cost [21].

Given that the model is likely to overfit the dataset, we improve the model’s robustness by adding perturbations to the word embedding (e.g. FGM).

2.2. Reading Comprehension and Sequence Tagging

Machine Reading Comprehension (MRC) plays a role in the development of Artificial Intelligence (AI) and still faces complicated problems at present. The early Question Answering (QA) system is rule-based and works really bad [22]. After that, the system is made up of rules [23] which compute the similarity of question-answer pairs and Bag-Of-Words Model (BOW) [24], [25]. But rule-based methods always fail. Soon the dataset MCTest is put forward whose instance contains a passage and question [26]. And the answer must be chosen from the four choices. Machine Learning approaches come up (e.g. to minimize the max-margin loss function) to perform better on the MCTest [27], [28], [29]. With the development of deep learning, the Long Short-Term Memory (LSTM) [30] with attention is proposed and achieves good results [31]. Since the ratio of noise in the CNN/Daily Mail dataset is high, SQuAD comes up to further boost the development of MRC [32]. Nowadays, BERT-based

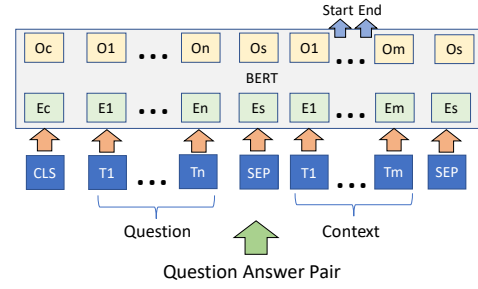


Figure 2: This is the process of predicting the answer span. The green cubes mean they have crossed an embedding block which contains three embedding layers: token embedding, segment embedding and position embedding. The yellow cubes mean they are already encoded by BERT.

methods have exceeded the performance of humans.

Inspired by the habit of humans that we first verify if the answer exists and then we can choose to answer it or not, Retrospective Reader is proposed to better tackle the complex problems [33]. Motivated by their work, we consider the acronyms as the named entity and other parts are not. By designing this strategy, we can also do sequence tagging task just like verifying the acronym.

3. Method

3.1. Problem Definition

The objective of Machine Reading Comprehension (MRC) is to output the distribution $p(a|q, c)$ where $q, c, a \in \mathcal{V}^*$ represent the given question, supporting context and the prediction answer respectively and are composed of tokens in the vocabulary \mathcal{V} (Fig 2). Since the context here is all the candidate expansions split by “|”, we exactly focus on extractive reading comprehension task because the answer can be found in the context. We denote $c_{i, \dots, j}$ as the answer span where $i < j$. The answer span is predicted by maximizing the sum of the possibility of the start of the answer span $p(start = i|q, c)$ and the possibility for the end $p(end = j|q, c)$.

The goal of Sequence Tagging is to predict the possibility of every position being the special token $p(\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n)$ such as the named entity based on the given sequence $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$.

3.2. Multi-Task Fine-Tuning

Our multi-task model gets improved by leveraging the both advantages of MRC and Sequence Tagging task. L_{span}, L_{tag} represent the loss of MRC and Sequence Tagging task respectively. The whole loss is computed by

equation 1 where α just indicates the importance of sequence tagging task.

$$L = L_{mrc} + \alpha L_{tag} \quad (1)$$

3.2.1. MRC

Following [12], the input sequence X is encoded by multi-layer Transformer [34]. Let $H = \{h_1, h_2, \dots, h_n\}$ denote the last-layer hidden states of X . The start possibility s can be computed by Equation 3. And the analogous rule for the end possibility e .

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \quad (2)$$

$$s = \text{Sigmoid}(\text{FFN}(H)) \quad (3)$$

And the aim of MRC is to fit a model to the examples drawn from the training dataset \mathcal{D} and θ refers to all the parameters.

$$\arg \min \mathbb{E}_{q,c,a \sim \mathcal{D}} [-\log p_{\theta}(a|q, c)] \quad (4)$$

We can turn the problem into minimizing the binary cross-entropy loss for the start and end predictions where y_i^s, y_i^e are respectively ground-truth start and end positions of example i and N is the size of \mathcal{D} .

$$L_{mrc} = -\frac{1}{N} \sum_{i=1}^N [\log(p_{y_i^s}^s) + \log(p_{y_i^e}^e)] \quad (5)$$

3.2.2. Sequence Tagging

We first label every token in the sequence 0 and 1 for the right expansion part which means other expansions are labeled 0. For instance:

- input = “[CLS]Acronym[SEP]A Maximum Entropy Approach to NERn.[SEP]Named Entity Recognition|named entity recognition...”
- labeled = [0,...,0,1,...,1,0,...,0]

We can train the model by minimizing the cross-entropy loss where \hat{y}_i is predicted by the model and y_i is the label just like the example’s.

$$\hat{y}_i = \text{Softmax}(\text{FFN}(H)) \quad (6)$$

$$L_{tag} = -\frac{1}{N} \sum_{i=1}^N [y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)] \quad (7)$$

3.3. Adversarial Training

Adversarial Training (AT) is a good way to enhance the model’s robustness by training the model on the generated adversarial examples. By adding small perturbations

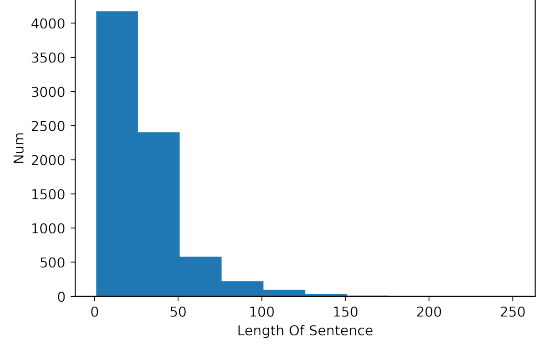


Figure 3: This shows the distribution of the length of sentences. More specifically, the length ranges from 1.00 to 251.00 and the mean length is 28.90.

to the features of training examples, AT can generate adversarial examples which are likely to induce the model to make wrong predictions. The model is first trained on the original training instances and then generate adversarial examples. Finally, the adversarial examples are used in the training process. The unified paradigm is summarized below [19].

$$\arg \min \mathbb{E}_{(x,y) \sim \mathcal{D}} [\max_{r_{adv} \in \mathcal{S}} L(\theta, x + r_{adv}, y)] \quad (8)$$

r_{adv}, \mathcal{S} are the perturbation and perturbation space respectively. L is the common loss function. The intuition of Equation 8 is to find the appropriate perturbation r_{adv} which can maximize the loss. By finding the best parameter θ to minimize the loss resulted from r_{adv} , our model can be more robust to the unseen perturbations in the real world. Following [18], we apply the same method to generate r_{adv} where ϵ is the hyper parameter which is default as 1. and g is the gradient of common training on the original dataset.

$$r_{adv} = \epsilon \cdot \frac{g}{\|g\|_2} \quad \text{where } g = \nabla_x L(\theta, x, y) \quad (9)$$

$$\|g\|_2 = \sqrt{|g_1|^2 + |g_2|^2 + \dots + |g_n|^2} \quad (10)$$

3.4. Gradient Centralization

Although the performance of Neural Networks like Transformer can be impressive, it is hard to train them because of the oscillation of training process and risks of being trapped in a local minimum. Performing normalization on activation or weights can to some degree improve this situation. Similar to normalization methods, gradient centralization (GC) centralizes the column vectors of weights so that the mean value of the column vectors

Table 2
Different Designs Of QA Pairs

P	R	F1	Question	Answer
0.5368 \pm 1.2%	0.4502 \pm 0.7%	0.4897 \pm 0.8%	[CLS] sentence [SEP]	choice
0.5569 \pm 2.1%	0.4728\pm1.4%	0.4949 \pm 1.3%	[CLS] acronym [SEP] sentence [SEP]	choice
0.5608 \pm 3.4%	0.4529 \pm 2.8%	0.5010 \pm 3.0%	[CLS] "Acronym": acronym [SEP] "Context": sentence [SEP]	"Option": choice
0.5812\pm2.0%	0.4644 \pm 0.4%	0.5162\pm1.0%	[CLS] "Context": sentence [SEP] "Acronym": acronym [SEP]	"Option": choice

is 0 [35]. Specifically, Φ_{GC}, \mathcal{L} are the GC operation and loss respectively. $\nabla_{\mathbf{w}_i} \mathcal{L}$ is the i th column of the gradient matrix and $\mu_{\nabla_{\mathbf{w}_i} \mathcal{L}}$ is the mean value of the i th column of the gradient matrix. By removing the mean value from every column vector of the gradient matrix, GC can make the optimization landscape more smoother which contributes to the more efficient training and the better generalization.

$$\Phi_{GC}(\nabla_{\mathbf{w}_i} \mathcal{L}) = \nabla_{\mathbf{w}_i} \mathcal{L} - \mu_{\nabla_{\mathbf{w}_i} \mathcal{L}} \quad (11)$$

4. Experiment

In this section, we present our model’s fine-tuning results on shared task 2: Acronym Disambiguation.

4.1. The Task

This task is designed to find the exact meaning of the ambiguous acronym in a given sentence. The input is a sentence which includes the acronym and the systems is going to figure out the expanded form of the acronym. For instance:

- **Input Sentence:** Here we present a non-linear method based on a deep CNN.
- **Input Candidate Long-forms:** convolutional neural network, Convolutional Neural Network, convolutional neural networks
- **Output:** convolutional neural network

Also, this task covers several languages: English, French and Spanish. Interestingly, only the English version contains different domains: legal and scientific [36]. And we choose the scientific one for it is more related to our field. Besides, there is no overlap among the acronyms in the training set, the development set and the evaluation set. The model’s performance is evaluated by the macro-averaged precision (P), recall (R) and F1.

4.2. Fine-Tuning Results

In this subsection, we present different plans for the modelling and the according results on the official dev test. To reduce the effect of the seed, every experiment is carried out with 3 different seeds:2021, 2022 and 2023.

Table 3
Adversarial Training and Gradient Centralization

Operation	P	R	F1
baseline	0.5812\pm2.0%	0.4644 \pm 0.4%	0.5162 \pm 1.0%
+ FGM	0.5707 \pm 1.3%	0.4634 \pm 1.2%	0.5113 \pm 0.3%
+ GC	0.58 \pm 2.9%	0.4682 \pm 1.4%	0.5181 \pm 1.9%
+ FGM,GC	0.5794 \pm 2.2%	0.4770\pm1.5%	0.5232\pm1.8%

4.2.1. Training Details

Since the case of words differs, we set do lower case as False. For the training process, we use 5 epochs and the batch size of both training and validating is 32. The optimizer is AdamW [37] which applies weight decay on Adam [38] in a different way. And the learning rate, adam epsilon, max grad norm and weight decay are $2e-5$, $1e-8$, 1.0 and 0.0 respectively. Also, we use the linear schedule for warming up and ratio is 0.1. Due to the observation in the dataset (Fig 3), the max sequence length is 160.

All the experiments are done on the GPU RTX6000 which has 48GB. And time for every independent experiment in Table 2, 3, 4 take 7m 51s, 13m 41s and 14m 59s respectively. Besides, the space for every experiment is 19GB.

4.2.2. Options of QA pairs

Different designs of QA pairs matter. In Table 2, the choice stands for the candidate expansions split by “|”. Note that the dropout ratio here is 0.0 to better observe the effect of different designs. Although the sentence contains the specific acronym such as CNN, concatenating it to the question still improves the model’s performances. “Acronym”, “Context” and “Option” are the prompts. Interestingly, just adding prompts to the question and answer works. Also, exchanging the position of sentence and acronym can lead to a better score. However, it should make no difference in humans’ thinking. Maybe the focus of the machine needs to be further explored.

4.2.3. Better Generalization

Since the model has the risk of overfitting the training dataset, we apply adversarial Learning such as FGM and Gradient Centralization (GC) to enhance the model’s

Table 4
Multi-Task Learning

α	P	R	F1
0.0	0.5812 \pm 2.0%	0.4644 \pm 0.4%	0.5162 \pm 1.0%
0.2	0.5963 \pm 2.4%	0.4823\pm1.6%	0.5333 \pm 1.9%
0.4	0.5747 \pm 3.3%	0.4618 \pm 1.3%	0.5120 \pm 2.1%
0.6	0.6055\pm1.5%	0.4810 \pm 1.7%	0.5361\pm0.6%
0.8	0.5758 \pm 0.8%	0.4675 \pm 0.5%	0.5160 \pm 0.2%
1.0	0.5906 \pm 2.1%	0.4733 \pm 1.5%	0.5255 \pm 1.8%

Table 5
Data Augmentation and Pseudo-labelling

	P	R	F1
baseline	0.5812 \pm 2.0%	0.4644 \pm 0.4%	0.5162 \pm 1.0%
shuffle	0.5843 \pm 1.4%	0.4685 \pm 1.7%	0.5201 \pm 1.6%
pseudo-labelling	0.6158\pm2.3%	0.4998\pm1.5%	0.5517\pm1.7%

generalization. In this process, we observe the dropout ratio can influence the training so we choose the best ratio from [0.1, 0.2, 0.3] (0.2). It turns out the combination of FGM and GC leads to the largest improvement (Table 3).

4.2.4. Multi-Task Learning

Here we do experiments on the effect of the value of α in Equation 1. Since it may take more time for convergence in multi-task learning, the epochs here is 10. Also, the dropout ratio is 0.0. We can draw conclusions from Table 4 that the joint training does help. Furthermore, appropriate α such as 0.6 can be a better choice.

4.2.5. Data Augmentation and Pseudo-labelling

We also use data augmentation such as shuffling the options during training, which can improve the performance (Table 5). Finally, we combine everything together and construct pseudo-labelling set for second training which leads to the comprehensive improvement.

5. Conclusion

By leveraging the advantages of both Machine Reading Comprehension task (MRC) and sequence tagging, our multi-task model gets improved in Acronym Disambiguation task. The combination of adversarial training and gradient centralization can further improve the model’s performance. And extra improvement can be made via designing useful prompts related to the specific task. For future work we plan to focus on the interesting phenomena observed in the experiments.

6. Acknowledgments

We thank the organizers of this Acronym Disambiguation task for sharing such an interesting topic with us and valuable advice from friends and reviewers. And this research has been supported by the National Natural Science Foundation of China under grants 61876217, 62176175; the Innovative Team of Jiangsu Province under grant XYDXX-086; the Science and Technology Development Project of Suzhou under grants SGC2021078.

References

- [1] S. Y. R. J. F. D. T. H. N. Amir Pouran Ben Veyseh, Nicole Meister, Multilingual Acronym Extraction and Disambiguation Shared Tasks at SDU 2022, in: Proceedings of SDU@AAAI-22, 2022.
- [2] A. S. Schwartz, M. A. Hearst, A simple algorithm for identifying abbreviation definitions in biomedical text, in: Biocomputing 2003, World Scientific, 2002, pp. 451–462.
- [3] N. Okazaki, S. Ananiadou, Building an abbreviation dictionary using a term recognition approach, *Bioinformatics* 22 (2006) 3089–3095.
- [4] K. Taghva, J. Gilbreth, Recognizing acronyms and their definitions, *International Journal on Document Analysis and Recognition* 1 (1999) 191–198.
- [5] C. G. Harris, P. Srinivasan, My word! machine versus human computation methods for identifying and resolving acronyms, *Computación y Sistemas* 23 (2019).
- [6] M. Billami, A knowledge-based approach to word sense disambiguation by distributional selection and semantic features, *arXiv preprint arXiv:1702.08450* (2017).
- [7] N. J. Vickers, Animal communication: when i’m calling you, will you answer too?, *Current biology* 27 (2017) R713–R715.
- [8] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, *arXiv preprint arXiv:1802.05365* (2018).
- [9] J. Charbonnier, C. Wartena, Using word embeddings for unsupervised acronym disambiguation (2018).
- [10] S. Melacci, A. Globo, L. Rigutini, Enhancing modern supervised word sense disambiguation models by semantic lexical resources, in: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), 2018.
- [11] S. Sousa, E. Milios, L. Berton, Word sense disambiguation: an evaluation study of semi-supervised approaches with word embeddings, in: 2020 In-

- ternational Joint Conference on Neural Networks (IJCNN), IEEE, 2020, pp. 1–8.
- [12] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).
- [13] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, S. R. Bowman, Glue: A multi-task benchmark and analysis platform for natural language understanding, arXiv preprint arXiv:1804.07461 (2018).
- [14] 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, arXiv preprint arXiv:1907.11692 (2019).
- [15] Y. Sun, S. Wang, Y. Li, S. Feng, X. Chen, H. Zhang, X. Tian, D. Zhu, H. Tian, H. Wu, Ernie: Enhanced representation through knowledge integration, arXiv preprint arXiv:1904.09223 (2019).
- [16] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, O. Levy, Spanbert: Improving pre-training by representing and predicting spans, Transactions of the Association for Computational Linguistics 8 (2020) 64–77.
- [17] I. J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples. iclr’15, arXiv preprint arXiv:1412.6572 (2015).
- [18] T. Miyato, A. Dai, goodfellow, ij adversarial training methods for semi-supervised text classification, in: International Conference on Learning Representations (ICLR), Toulon, France, 2017.
- [19] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, arXiv preprint arXiv:1706.06083 (2017).
- [20] D. Zhang, T. Zhang, Y. Lu, Z. Zhu, B. Dong, You only propagate once: Accelerating adversarial training via maximal principle, arXiv preprint arXiv:1905.00877 (2019).
- [21] A. Shafahi, M. Najibi, A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, T. Goldstein, Adversarial training for free!, arXiv preprint arXiv:1904.12843 (2019).
- [22] W. G. Lehnert, The Process of Question Answering., Ph.D. thesis, USA, 1977. AAI7728146.
- [23] A. Peñas, E. Hovy, Semantic enrichment of text with background knowledge, in: Proceedings of the NAACL HLT 2010 First International Workshop on Formalisms and Methodology for Learning by Reading, Association for Computational Linguistics, Los Angeles, California, 2010, pp. 15–23. URL: <https://aclanthology.org/W10-0903>.
- [24] Z. S. Harris, Distributional structure, Word 10 (1954) 146–162.
- [25] P. Ruch, R. Baud, A. Geissbühler, Evaluating and reducing the effect of data corruption when applying bag of words approaches to medical records, International journal of medical informatics 67 (2003) 75–83. doi:10.1016/S1386-5056(02)00057-6.
- [26] M. Richardson, C. J. Burges, E. Renshaw, MCTest: A challenge dataset for the open-domain machine comprehension of text, in: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Seattle, Washington, USA, 2013, pp. 193–203. URL: <https://aclanthology.org/D13-1020>.
- [27] K. Narasimhan, R. Barzilay, Machine comprehension with discourse relations, 2015. doi:10.3115/v1/p15-1121, publisher Copyright: © 2015 Association for Computational Linguistics.; 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL-IJCNLP 2015 ; Conference date: 26-07-2015 Through 31-07-2015.
- [28] M. Sachan, K. Dubey, E. Xing, M. Richardson, Learning answer-entailing structures for machine comprehension, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Beijing, China, 2015, pp. 239–249. URL: <https://aclanthology.org/P15-1024>. doi:10.3115/v1/P15-1024.
- [29] H. Wang, M. Bansal, K. Gimpel, D. McAllester, Machine comprehension with syntax, frames, and semantics, 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), Association for Computational Linguistics, Beijing, China, 2015, pp. 700–706. URL: <https://aclanthology.org/P15-2115>. doi:10.3115/v1/P15-2115.
- [30] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (1997) 1735–1780.
- [31] K. M. Hermann, T. Kociský, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, P. Blunsom, Teaching machines to read and comprehend, CoRR abs/1506.03340 (2015). URL: <http://arxiv.org/abs/1506.03340>. arXiv:1506.03340.
- [32] P. Rajpurkar, J. Zhang, K. Lopyrev, P. Liang, SQuAD: 100,000+ questions for machine comprehension of text, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Austin, Texas, 2016, pp. 2383–2392. URL: <https://aclanthology.org/D16-1264>. doi:10.18653/v1/D16-1264.

- [33] Z. Zhang, J. Yang, H. Zhao, Retrospective reader for machine reading comprehension, 2020. [arXiv:2001.09694](https://arxiv.org/abs/2001.09694).
- [34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in neural information processing systems, 2017, pp. 5998–6008.
- [35] H. Yong, J. Huang, X. Hua, L. Zhang, Gradient centralization: A new optimization technique for deep neural networks, in: European Conference on Computer Vision, Springer, 2020, pp. 635–652.
- [36] S. Y. R. J. F. D. T. H. N. Amir Pouran Ben Veyseh, Nicole Meister, MACRONYM: A Large-Scale Dataset for Multilingual and Multi-Domain Acronym Extraction, in: [arXiv](https://arxiv.org/abs/2022.01.01), 2022.
- [37] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, [arXiv preprint arXiv:1711.05101](https://arxiv.org/abs/1711.05101) (2017).
- [38] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, [arXiv preprint arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014).