# Prompt-based Model for Acronym Disambiguation via Negative Sampling

Taiqiang Wu[1], Xingyu Bai[2] and Yujiu Yang[1]

[1]Tsinghua Shenzhen International Graduate School Tsinghua University, P. R. China

**Abstract**

Acronym Disambiguation (AD) task aims to map the acronym in sentences to the corresponding expansion among candidate expansions. However, these models based on domain agnostic knowledge might perform insufficient when directly applied to the data in some specific areas such as science and law. To track these issues, we propose a prompt-based acronym disambiguation system with special negative sampling. Specially, we design a prompt to combine the input sentences and candidate expansions, followed by a Pre-train Language Model (PLM) to calculate the score. Moreover, negative expansions are randomly sampled for better training, and an additional hinge loss is added to improve the robustness of our system. Experiments show the effectiveness of our system, and we get competitive results in the SDU@AAAI-22-Shared Task 2: Acronym Disambiguation.

**Keywords**

prompt learning negative sampling hinge loss

## 1. Introduction

Acronyms are abbreviations formed from the initial components of words or phrases [1]. They are widely used in our daily life especially on social media. By using acronyms, people can avoid frequently repeating long phrases; thus, the sentences could be shorter and more readable. For example, we use NASA to replace the National Aeronautics and Space Administration.

However, for people without domain knowledge, acronyms might be confusing at some time, such as "PPP" can be Paycheck Protection Program or Public-Private Partnership. It is necessary to build an acronym disambiguation system that can identify the correct meaning of acronyms in a different context to track this issue. As shown in Figure 1, given several sentences containing acronym *POS*, we need to find out the corresponding expansion among candidate expansions in the given dictionary. Moreover, understanding the correlation between acronyms and their expansion is beneficial for several tasks in natural language processing, including question answering and machine reading comprehension.

Acronym disambiguation is usually considered as a sequence classification task [2], the goal is to map the given acronym in context to the corresponding expansion from the candidate expansion dictionary. Previous works mainly focused on the feature construction of acronym context to better understand semantics, such as hand crafted rules and patterns [1], word embeddings [3], graph structures [2], machine learning based methods such as CRF and SVM [4], and deep learning based methods [5, 6]. The experiments on this task were further extended to learn richer semantics features using Transformer [7], BERT [8] and SciBERT [9]. Although these efforts have achieved significant performance in this task, most of them ignored modeling the semantic relationship between acronym context and candidate expansions.

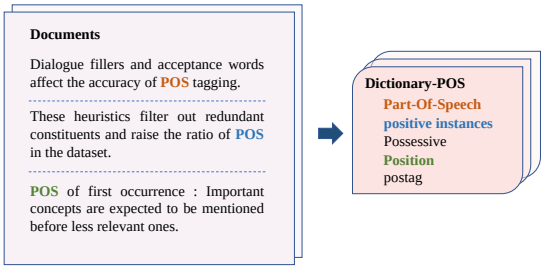


**Figure 1:** An example of acronym disambiguation for *POS*. The expansion in the same color is just the corresponding expansion for each *POS*.

Furthermore, large-scale data during training brings an extremely long-tail problem. The size of the original candidate expansions in the dictionary varies, making it hard to batch the samples during training. To address this issue, previous works [10] dynamically add extra expansions into the candidate expansion set. However, they ignore the fact that the original negative candidate expansions are related to the acronym word in semantic meaning while the added expansions are unrelated.

In this paper, we proposed a prompt-based acronym

*AAAI'22: Scientific Document Understanding*
wtq16@mails.tsinghua.edu.cn (T. Wu);
bxy16@mails.tsinghua.edu.cn (X. Bai);
yang.yujiu@sz.tsinghua.edu.cn (Y. Yang)

disambiguation framework with a specially designed negative sampling strategy. Firstly, we design a prompt template and use the template to concatenate the acronym context and candidate expansions. Secondly, we utilize a pre-trained language model such as BERT [8] to encode the combined context separately, followed by a linear layer to map the context vectors into logits. Since the size of candidate expansions for each acronym varies, we try to sample negative samples, thus padding the candidate expansions randomly. Finally, we consider the original negative expansions as hard negative samples and the added ones as easy negative samples, which can calculate an extra loss to build a more robust system. The main contributions of this work are summarized as follows:

- We design a prompt-based framework to resolve the acronym disambiguation problem, which can be easily modified to solve other NLP tasks such as Entity Linking.
- We propose a simple yet effective dynamic negative sampling strategy and adopt a novel hinge loss to help train a robust model. The strategy can benefit other matching problems.
- We conduct experiments on the SDU@AAAI22 shared task 2 dataset and achieve competitive performance, demonstrating our framework's effectiveness.

## 2. Related Work

In this section, we mainly introduce the related studies for prompt-based models, especially the BERT-based models. We first review the existing researches on word sense disambiguation, which is more generalized than the acronym disambiguation.

### 2.1. Prompt-based Learning

Prompt is suggestive information to enhance the knowledge that PLMs (Pre-trained Language Models) learned during pre-training, containing the description of task answers and corresponding answers. Prompt-based learning is a slot-filling method based on language models, which aims to probabilistically construct the final prompt as the prediction of the task. Previous exploration in prompt-based learning mainly focuses on prompt construction, including prompt engineering and answer engineering. Prompt engineering creates a prompt function applicable to corresponding downstream tasks [11, 12], While answer engineering searches for a unified answer space to which the original answers are mapped [11]. Multi-prompt learning, an ensemble of these two engineering prompts, aims to improve the generalization of models [13]. Based on multi-prompt learning, various combinations of prompts have been explored, such as prompt augmentation [14], prompt composition [15] and prompt decomposition [16]. In this work, we construct different forms of prompts manually, to enrich the knowledge enhancement methods.

### 2.2. Word Sense Disambiguation

Word Sense Disambiguation(WSD) is divided into supervised, unsupervised and semi-supervised methods.

In supervised WSD methods, classic machine learning-based methods, such as decision tree, SVM, ANN and naive Bayes models, have been combined to improve the complexity of classifier [17]. WSD model based on evolutionary game theory was designed to determine the prediction of ambiguous words by calculating distribution and semantic similarity [18]. Supervised neural network with LKB graph embedding was proposed for transferring the pre-trained embeddings of synset to predict ones [19].

Unsupervised WSD methods mainly cluster the unlabeled corpus to predict the category of ambiguous words. The classic hybrid model consists of self-adaptive genetic, max-min ant and any colony algorithms [20]. WSD models based on polysemy vector representation adopted statistical polysemy, word sense numbers, and K-means to finish disambiguation [21]. Word sense mapping graph network can be combined with multilinguistic and multi-knowledge resources to integrate rich information in unsupervised scenario [22].

In semi-supervised WSD models, the classifier is trained by the integration of annotated and unannotated corpora. PageRank-based WSD algorithm combined pIWordNet and semantic links from valency lexicon, Wikipedia articles and SUMO ontology [23]. Clustering and labeling strategy was used to generate labeled data for subjectivity WSD semi-automatically and further combined with original annotated data [24].

However, all these methods ignore the interaction between ambiguous word explanation and its context. In this work, we propose a prompt-based model to integrate better the semantic relationship between acronym context and candidate expansions.

## 3. Methodology

In this section, we present the overall architecture of our proposed framework, which uses the prompt-based model to solve the acronym disambiguation problem and adopt a dynamic negative sampling strategy to improve the robustness of our model.

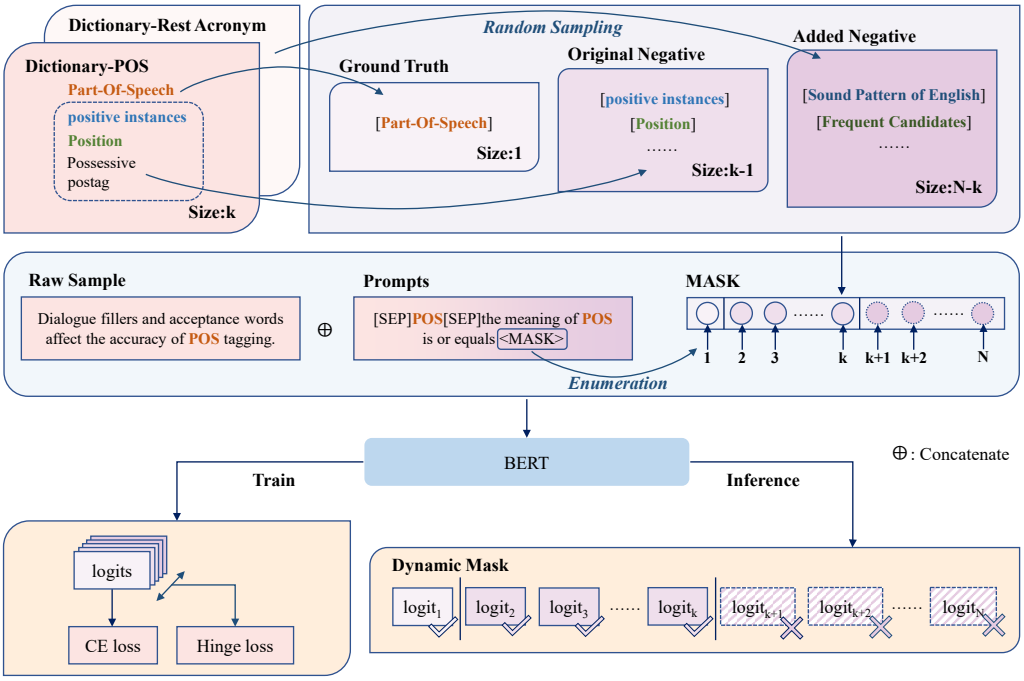


**Figure 2:** Overview of our proposed framework. For acronym *POS*, there are $k$ expansions in the dictionary. we adopt a sample strategy to sample $N-k$ samples. We design a prompt template: [SEP] *acronym* [SEP] the meaning of *acronym* is or equals *expansion*. After that, a BERT-based model is employed as encoder to calculate the logits. For inference, we will generate a dynamic mask to ignore the logits from added expansions.

### 3.1. Problem Statement

Formally, given an input sentence $s = w_1, w_2, ..., w_n$ and acronym $a = w_i$ at position $i$, the goal is to disambiguate the corresponding expansions $e_j$ among $n$ candidate expansions $\{ce_1, ce_2, ..., ce_n\}$. The candidate expansions are given in advance and their size vary. Specifically, in this paper, we treat this task as a classification problem by padding the candidate expansions set to fix length with randomly chosen unrelated expansions. We will dynamically mask the logit of added expansion in the testing phase and choose the largest one among original candidate expansions as the final prediction.

### 3.2. Overview

As shown in Figure 2, given the acronym *POS* in the sentence, there are $k$ candidate expansions which can be divided into a positive sample set of size 1 and a hard negative sample set of size $k-1$. Firstly, in the expansions of other acronyms, we randomly sample $N-k$ samples as the easy negative sample set to pad the candidate expansions into fix size $N$. Secondly, we design a prompt strategy to combine the acronym and candidate expansions. [SEP] token is inserted before and after the acronym, followed by a string: the meaning of *acronym* is or equals *expansion*. Finally, BERT with an additional linear layer is employed as our encoder. For training, we will calculate the cross-entropy loss and adopted hinge loss [25]. For inference, a dynamic mask strategy is adopted, in which we will drop the logits of added expansions. Specially, we will drop the logits from the added negative samples, which can not be the answer.

### 3.3. Prompt Design

To build a prompt template effectively, we consider a two-stage strategy. We hope the model to be aware of two tasks: **finding out the acronym** and **finding out the corresponding expansion**. Thus, we employ the token [SEP] to highlight the acronym, which can help the model to understand where the acronym is. For second task, previous works[26, 27] show that a longer prompt usually performs better. To add more tokens, we use the template: the meaning of *acronym* is or equals *expansion*. For French and Spanish, we employ the corresponding translation as the prompt templates.

### 3.4. Negative Sampling

The size of candidate expansions in the dictionary varies, making it hard to train an efficient model. Moreover, we consider the negative samples in the original candidate expansions as **related to but not exactly the ground truth**. To improve the robustness and convergence of the model, we adopt a negative sampling strategy. We set the size of the padded candidate set as $N$ and randomly sample expansions from the candidate expansions of other acronyms as needed. For example, $N$ is set to 6, and the number of original candidate expansions is 2. We need to pick up 4 additional expansions. We note that [10] also proposed a similar negative sampling strategy. The difference is that we divided the negative samples into hard negative samples and easy negative samples, thus designing extra loss.

### 3.5. Loss Function

For the model, we consider two goals: 1) the ground truth expansion gets the highest score; 2) the original negative expansions get higher scores than the additional negative expansions. For the first goal, we employ the cross entropy loss function. Note the predict label as $Y_{pred}$ and the ground truth label as $Y_{gt}$.

$$Loss_{ce} = CE(Y_{pred}, Y_{gt}) \tag{1}$$

where the $CE$ means cross entropy loss function. For the second goal, we follow the idea of hinge loss and we want the minimum of the original expansion scores $S_{ori} = \{score_1, score_2, ..., score_{k-1}\}$ is higher than the maximum of the additional expansion scores $S_{add} = \{score_1, score_2, ..., score_{N-k}\}$ by a margin.

$$Loss_{hin} = \max(\lambda - \min(S_{ori}) + \max(S_{add}), 0) \tag{2}$$

where $\max(\cdot)$ and $\min(\cdot)$ mean the maximum and minimum function while $\lambda$ is a learnable margin. Hence we get our final loss.

$$Loss = Loss_{ce} + \mu Loss_{hin} \tag{3}$$

where $\mu$ is also a learnable hyperparameter to control the ratio of hinge loss.

## 4. Experiments

In this section, we first introduce the experimental dataset and evaluation metrics and then conduct comprehensive experimental studies to verify the effectiveness of our method.

| Dataset | Sentences | Tokens | Acronyms | Expansions |
|---|---|---|---|---|
| Legal English | 3,717 | 174,997 | 303 | 625 |
| Scientific English | 9,000 | 245,558 | 497 | 1,551 |
| French | 9,573 | 502,461 | 669 | 1,622 |
| Spanish | 7,947 | 426,404 | 546 | 1,277 |

**Table 1**
Statics of all datasets.

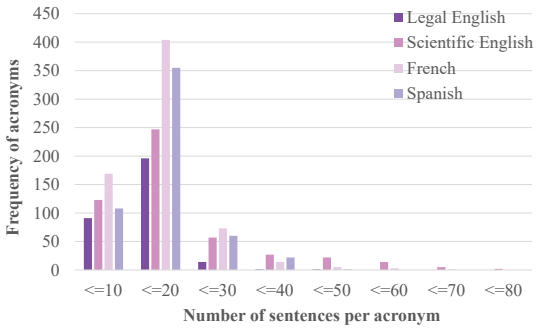


**Figure 3:** Distribution of acronyms based on sentence numbers per acronym.

### 4.1. Dataset

We evaluate all models based on the dataset provided by SDU@AAAI-22 [28]. As shown in Table 1, the dataset [29] contains training and development datasets in English (both scientific and legal domain), Spanish, and French consisting of 497 English Scientific, 303 English legal, 546 Spanish, and 669 French acronyms. For each language, a diction containing acronyms and their candidate expansions is provided. For Legal English, there are 3717 sentences containing 174997 tokens and 625 candidate expansions in the diction. The average expansion length of all acronyms is 3.1. The acronyms in the testing set would not appear in the training set.

For Exploratory Data Analysis(EDA), we analyze the statistical features in the dataset. As shown in Figure 3 and Figure 4, we can see that: 1) for most acronyms, the corresponding sentences are more than 10, indicating that the samples are highly similar. 2) for most acronyms, the corresponding candidate expansions are less than 4.

### 4.2. Evaluation Metrics

Given the acronyms in sentences, candidate expansions and ground truth labels, we can calculate the macro-averaged precision, recall and F1 score.

### 4.3. Implement

All models are implemented based on the open-source transformers library of Huggingface [30]. For all datasets,

| Dataset | Model | Epoch | | | | |
|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 |
| **Legal English** | bert-large-cased | 49.38 | **53.27** | **60.39** | 55.79 | 57.43 |
| | spanbert-large-cased | **49.69** | 51.64 | 56.98 | **56.96** | **60.78** |
| **Scientific English** | scibert-scivocab-cased | **63.40** | **66.37** | **71.90** | 69.13 | 68.95 |
| | scibert-scivocab-cased ($\mu = 1.5$) | 58.62 | 64.90 | 68.51 | **69.81** | **69.82** |
| **French** | bert-base-french-europeana-cased | 59.57 | 62.97 | 63.50 | 63.58 | 62.69 |
| | camembert-large | **66.31** | **65.90** | **72.05** | **67.76** | **72.70** |
| **Spanish** | bert-base-spanish-wwm-cased | 50.27 | 52.69 | 53.37 | 54.90 | 53.39 |
| | bert-base-multilingual-cased | **53.05** | **57.71** | **62.89** | **65.31** | **67.02** |

**Table 2**
Evaluation with various models on the valid set of all datasets. $F1(\%)$ represent F1-score.

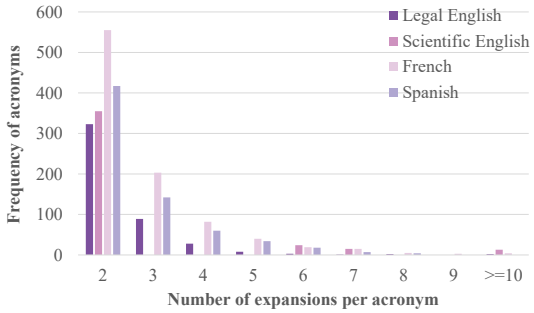


**Figure 4:** Distribution of acronyms based on expansion numbers per acronym.

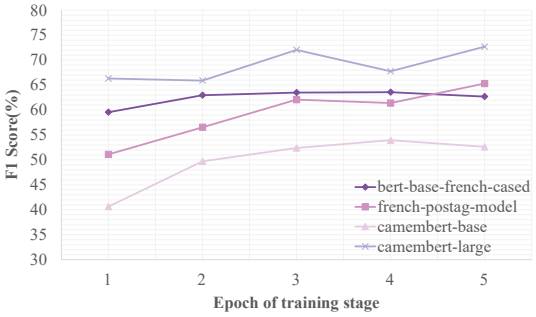


**Figure 5:** F1-score curve on French development dataset.

we set the $\lambda = 0.1$ and $\mu = 1$. The batch size is 2 and the size of expected expansion $N = max([k_i]) + 2$. For example, for French dataset, the maximum of candidate expansions for all acronyms is 12, thus we set $N = 12 + 2 = 14$. As for other parameters, we set the learning rate as $3e - 5$ and random seed as 10086. We pad or cut the input into 128 length. For French dataset, the prompt is: el significado de *acronym* es o igual a *expansion*. For Spanish dataset, we use: la signification de *acronym* est ou est égale à *expansion*. We train our model in one V100 GPU and evaluate the result using the official script.

## 4.4. Comparison

### 4.4.1. Overall Performance

The overall performance results on the validation set are shown in Table 2. For Legal English, we choose the bert-large-cased [31] and spanbert-large-cased as the PLM. For Scientific English, we choose scibert-sci-vocab-cased [32]. For French, we choose bert-base-french-europeana-cased and camembert-large [33]. For Spanish, we choose bert-base-spanish-wwm-cased [34] and bert-base-multilingual-cased [31]. As shown in Table 2, we can observe that most models suffer from over-fitting after 3 epochs. Moreover, we find that the BERT trained on the specialized corpus performs better than trained on the common corpus.

### 4.4.2. The Effect of PLM

We change the BERT type to study the influence of different backbones on the French dataset. As shown in Figure 5, we can see that the larger models usually get better results. Another interesting observation is that all models suffer from over-fitting at epoch 4.

### 4.4.3. The Effect of Margin $\lambda$

We change the $\lambda$ to 0.0 and 1.0 and conduct our experiments in the English science dataset. According to Figure 6, we can find that a large $\lambda$ brings a considerable change during training. Actually, a large $\mu$ means a large gap is required, leading to the oscillation in the loss.

### 4.4.4. The Effect of Ratio $\mu$

We change the $\mu$ to 0.5 and 1.5 and conduct our experiments in the English Science dataset. As shown in Figure 7, the larger $\mu$ leads to a lower result. Actually, a large $\mu$ means a large hinge loss, which pushes the model to

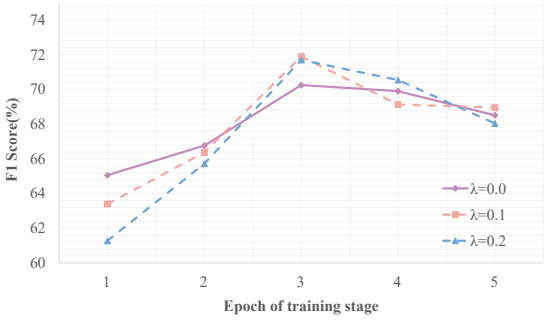


**Figure 6:** F1-score curve on Scientific English development dataset with $\lambda$.

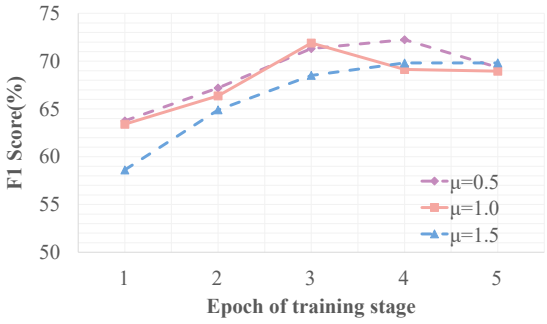


**Figure 7:** F1-score curve on Scientific English development dataset with $\mu$.

be more over-fitting on the training data. With more epochs trained, the gap between different results becomes smaller. It indicates that the hinge loss becomes smaller as trained for more epochs.

## 5. Conclusion and Future Work

In this paper, we proposed a novel prompt-based model, which shows promising and competitive performance in SDU@AAAI-22 - Shared Task 2. We design an effective, prompt template that helps the model utilize the implicit knowledge in the pre-trained language model. A dynamic negative sampling strategy is employed to improve the robustness and performance of our model.

For future work, we will try to adopt a learned prompt template rather than a fixed template following the CoOp [26]. Moreover, the acronym disambiguation under a zero-shot setting would be another interesting and valuable topic. Utilizing the graph information in given sentence [35] may also help.

## 6. Acknowledgments

This research was supported in part by the National Key Research and Development Program of China (No. 2018YFB1601102) and the Shenzhen Key Laboratory of Marine IntelliSense and Computation under Contract ZDSYS20200811142605016. We thank the organizers of acronym identification and disambiguation competitions and the reviewers for their valuable comments and suggestions.

## References

[1] Y. Li, B. Zhao, A. Fuxman, F. Tao, Guess me if you can: Acronym disambiguation for enterprises, in: ACL2018, 2018, pp. 1308–1317.

[2] A. P. B. Veyseh, F. Dernoncourt, Q. H. Tran, T. H. Nguyen, What does this acronym mean? introducing a new dataset for acronym identification and disambiguation, in: Proceedings of the COLING 2020, International Committee on Computational Linguistics, 2020, pp. 3285–3301.

[3] M. R. Ciosici, T. Sommer, I. Assent, Unsupervised abbreviation disambiguation contextual disambiguation using word embeddings, CoRR abs/1904.00929 (2019). URL: http://arxiv.org/abs/1904.00929. arXiv:1904.00929.

[4] J. Liu, C. Liu, Y. Huang, Multi-granularity sequence labeling model for acronym expansion identification, Inf. Sci. 378 (2017) 462–474.

[5] J. Charbonnier, C. Wartena, Using word embeddings for unsupervised acronym disambiguation, in: Proceedings of the COLING 2018, Association for Computational Linguistics, 2018, pp. 2610–2619.

[6] Q. Jin, J. Liu, X. Lu, Deep contextualized biomedical abbreviation expansion, in: Proceedings of the BioNLP@ACL, Association for Computational Linguistics, 2019, pp. 88–96.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems 30: NeurIPS 2017, 2017, pp. 5998–6008.

[8] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the NAACL-HLT 2019, Volume 1 (Long and Short Papers), Association for Computational Linguistics, 2019, pp. 4171–4186.

[9] I. Beltagy, K. Lo, A. Cohan, Scibert: A pretrained language model for scientific text, in: Proceedings of the EMNLP-IJCNLP 2019, Association for Computational Linguistics, 2019, pp. 3613–3618.

[10] C. Pan, B. Song, S. Wang, Z. Luo, Bert-based

acronym disambiguation with multiple training strategies, in: Proceedings of the SDU@AAAI 2021, volume 2831 of *CEUR Workshop Proceedings*, 2021.

[11] T. Shin, Y. Razeghi, R. L. L. IV, E. Wallace, S. Singh, Autoprompt: Eliciting knowledge from language models with automatically generated prompts, in: Proceedings of the EMNLP 2020, Association for Computational Linguistics, 2020, pp. 4222–4235.

[12] Z. Jiang, F. F. Xu, J. Araki, G. Neubig, How can we know what language models know, Trans. Assoc. Comput. Linguistics 8 (2020) 423–438.

[13] W. Yuan, G. Neubig, P. Liu, Bartscore: Evaluating generated text as text generation, CoRR abs/2106.11520 (2021).

[14] T. Gao, A. Fisch, D. Chen, Making pre-trained language models better few-shot learners, in: Proceedings of the ACL/IJCNLP 2021, (Volume 1: Long Papers), Association for Computational Linguistics, 2021, pp. 3816–3830.

[15] X. Han, W. Zhao, N. Ding, Z. Liu, M. Sun, PTR: prompt tuning with rules for text classification, CoRR abs/2105.11259 (2021). URL: https://arxiv.org/abs/2105.11259. arXiv:2105.11259.

[16] L. Cui, Y. Wu, J. Liu, S. Yang, Y. Zhang, Template-based named entity recognition using BART, in: Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, volume ACL/IJCNLP 2021 of *Findings of ACL*, Association for Computational Linguistics, 2021, pp. 1835–1845.

[17] A. R. Pal, D. Saha, N. S. Dash, S. K. Naskar, A. Pal, A novel approach to word sense disambiguation in bengali language using supervised methodology, Sādhanā 44 (2019) 1–12.

[18] R. Tripodi, M. Pelillo, A game-theoretic approach to word sense disambiguation, Comput. Linguistics 43 (2017) 31–70.

[19] M. Bevilacqua, R. Navigli, Breaking through the 80% glass ceiling: Raising the state of the art in word sense disambiguation by incorporating knowledge graph information, in: Proceedings of the ACL 2020, Association for Computational Linguistics, 2020, pp. 2854–2864.

[20] W. Alsaeedan, M. E. B. Menai, S. A. Al-Ahmadi, A hybrid genetic-ant colony optimization algorithm for the word sense disambiguation problem, Inf. Sci. 417 (2017) 20–38.

[21] G. Wiedemann, S. Remus, A. Chawla, C. Biemann, Does BERT make any sense? interpretable word sense disambiguation with contextualized embeddings, in: Proceedings of the 15th Conference on Natural Language Processing, KONVENS 2019, 2019.

[22] W. Lu, F. Meng, S. Wang, G. Zhang, X. Zhang, A. Ouyang, X. Zhang, Graph-based chinese word sense disambiguation with multi-knowledge integration, Comput. Mater. Continua 61 (2019) 197–212.

[23] A. Janz, M. Piasecki, A weakly supervised word sense disambiguation for polish using rich lexical resources, Poznan Studies in Contemporary Linguistics 55 (2019) 339–365.

[24] C. Akkaya, J. Wiebe, R. Mihalcea, Iterative constrained clustering for subjectivity word sense disambiguation, in: Proceedings of the EACL 2014, The Association for Computer Linguistics, 2014, pp. 269–278.

[25] C. Gentile, M. K. Warmuth, Linear hinge loss and average margin, in: Advances in Neural Information Processing Systems 11, [NIPS 1998, The MIT Press, 1998, pp. 225–231.

[26] K. Zhou, J. Yang, C. C. Loy, Z. Liu, Learning to prompt for vision-language models, CoRR abs/2109.01134 (2021). URL: https://arxiv.org/abs/2109.01134. arXiv:2109.01134.

[27] B. Lester, R. Al-Rfou, N. Constant, The power of scale for parameter-efficient prompt tuning, CoRR abs/2104.08691 (2021). URL: https://arxiv.org/abs/2104.08691. arXiv:2104.08691.

[28] 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.

[29] 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, 2022.

[30] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, A. M. Rush, Transformers: State-of-the-art natural language processing, in: Proceedings of the EMNLP 2020 - Demos, Association for Computational Linguistics, 2020, pp. 38–45.

[31] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, CoRR abs/1810.04805 (2018). URL: http://arxiv.org/abs/1810.04805. arXiv:1810.04805.

[32] I. Beltagy, K. Lo, A. Cohan, Scibert: A pretrained language model for scientific text, in: EMNLP, Association for Computational Linguistics, 2019. URL: https://www.aclweb.org/anthology/D19-1371.

[33] L. Martin, B. Muller, P. J. O. Suárez, Y. Dupont, L. Romary, É. V. de la Clergerie, D. Seddah, B. Sagot, Camembert: a tasty french language model, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020.

[34] J. Cañete, G. Chaperon, R. Fuentes, J.-H. Ho, H. Kang, J. Pérez, Spanish pre-trained bert model

and evaluation data, in: PML4DC at ICLR 2020, 2020.

[35] L. Ding, Z. Lei, G. Xun, Y. Yang, FAT-RE: A faster dependency-free model for relation extraction, J. Web Semant. 65 (2020) 100598. URL: https://doi.org/10.1016/j.websem.2020.100598. doi:10.1016/j.websem.2020.100598.