T5 Encoder Based Acronym Disambiguation with Weak Supervision

Gwangho Song[1], Hongrae Lee[2] and Kyuseok Shim[1,3]

[1] Seoul National University, Seoul, South Korea
[2] Google, Mountain View, CA, USA
[3] Corresponding author

Abstract

An acronym is a word formed by abbreviating a phrase by combining certain letters of words in the phrase into a single term. Acronym disambiguation task selects the correct expansion of an ambiguous acronym in a sentence among the candidate expansions in a dictionary. Although it is convenient to use acronyms, identifying the appropriate expansions of an acronym in a sentence is a difficult task in natural language processing. Based on the recent success of the large-scale pre-trained language models such as BERT and T5, we propose a binary classification model using those language models for acronym disambiguation. To overcome the limited coverage of a training data, we use a weak supervision approach to increase the training data. Specifically, after collecting sentences containing an expansion of an acronym from Wikipedia, we replace the expansion with its acronym and label the sentence with the expansion. By conducting extensive experiments, we show the effectiveness of the proposed model. Our model is placed in the top 3 models for three of four categories in SDU@AAAI-22 shared task 2: Acronym Disambiguation.

Keywords

acronym disambiguation, natural language processing, deep learning, weak supervision

1. Introduction

An acronym is a word formed by abbreviating a phrase which is called a long-form or an expansion (e.g., AAAI for Association for the Advancement of Artificial Intelligence). Due to its brevity, its usage is ubiquitous in many literature and documents, especially in scientific and biomedical fields [1, 2, 3, 4, 5]. A report found that more than 63% of the articles in English Wikipedia contain at least one abbreviation [1]. Furthermore, among more than 24 million article titles and 18 million article abstracts published between 1950 and 2019, there is at least one acronym in 19% of the titles and 73% of the abstracts [2].

Acronyms frequently have multiple long-forms, and only one of them is valid for a specific context. For example, in a 2001 version of the WWWAAS (World-Wide Web Acronym and Abbreviation Server) database, 47.97% of acronyms have multiple expansions [6]. As another example, in the SciAD dataset released by SDU@AAAI 2021 Shared Task: Acronym Disambiguation [5], an acronym has 3.1 long-forms on average and up to 20 long-forms. When sufficient context is not available, this leads to the ambiguity of the meaning of acronyms and creates serious understanding difficulties [2, 7, 8, 9]. Thus, acronym disambiguation task is important and challenging.

Input:

- **Sentence:** Since our generative models are based on **DP** priors, they are designed to favor a small number of unique entities per image.
- **Dictionary:** DP { Dynamic Programming; Dependency Parsing; Dirichlet Process }

Output: Dirichlet Process

Figure 1: An example of acronym disambiguation

The goal of acronym disambiguation(AD) is to select the correct long-form of an ambiguous acronym in a sentence among the candidate long-forms in a dictionary. Figure 1 shows an example of acronym disambiguation. A sentence containing an ambiguous acronym "DP" and a dictionary with the long-forms of "DP" are given as the input. In the dictionary, the acronym "DP" has three possible long-forms: "Dynamic Programming", "Dependency Parsing" and "Dirichlet Process". According to the context of the input sentence, since "DP" stands for "Dirichlet Process", a model outputs "Dirichlet Process" as its expansion.

The problem of acronym disambiguation is usually cast as a classification problem whose goal is to determine whether a long-form has the same meaning with the acronym in an input sentence. Early approaches [10, 11,

Scientific Document Understanding Workshop at AAAI 2022, March 1
ghsong@kdd.snu.ac.kr (G. Song); mr.hongrae.lee@gmail.com (H. Lee); kshim@snu.ac.kr (K. Shim)
0000-0002-9450-5629 (G. Song); 0000-0002-6138-3071 (H. Lee); 0000-0001-8818-0963 (K. Shim)

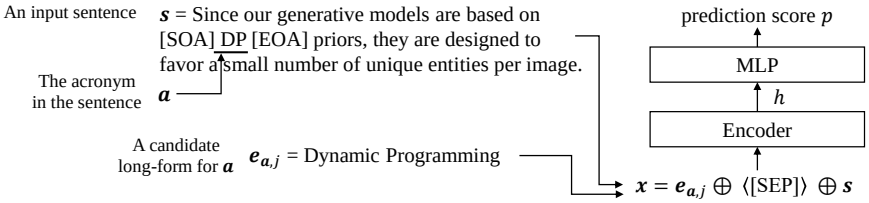

Figure 2: An illustration of the proposed model

12, 6] rely on the traditional classification models such as SVMs, decision trees and naive Bayes classifiers. As deep learning becomes more mainstream in natural language processing, several works employ contextualized word embeddings to create semantic representations of long-forms and context [9, 13, 14, 15, 16]. Moreover, with the recent success of the pre-trained language models such as BERT [17] and T5 [18] in natural language processing, classification models for acronym disambiguation are developed based on the pre-trained language models [4, 19, 20, 21].

To study multilingual acronym disambiguation, we develop a binary classification model by utilizing T5 [18], which is one of the most popular pre-trained language models, as well as mT5 [22] which is a multilingual variant of T5. We evaluate the proposed model on the dataset released by SDU@AAAI 2022 Shared Task: Acronym Disambiguation [23]. Since the acronyms in the test dataset do not appear in the training dataset, the training dataset provided in the competition may not be sufficient to solve the problem. Thus, we use a weak supervision approach to increase the training dataset. By training on the provided training dataset as well as the weakly labeled training dataset generated by our weak supervision method, the proposed model ranks in the top 3 place for three of four categories in SDU@AAAI-22 shared task 2: Acronym Disambiguation.

The remainder of this paper is organized as follows. We provide related work in Section 2 and present our proposed model in Section 3. In Section 4, we describe the datasets used for training the model, including weakly labeled datasets generated by weak supervision. Finally, we discuss the experimental results in Section 5 and summarize the paper in Section 6.

2. Related Work

In this section, we present the previous works on acronym disambiguation. We also summarize the pre-trained language models widely adopted in various natural language processing. In addition, we introduce weak supervision approaches to construct additional data.

2.1. Acronym Disambiguation

Early approaches [10, 11, 12, 6] rely on the traditional classification models such as SVMs, decision trees and naive Bayes classifiers. As deep learning becomes more mainstream in natural language processing, several works employ contextualized word embeddings to create semantic representations of long-forms and context [9, 13, 14, 15, 16]. The works in [13, 14] study the use of word embeddings [24, 25] to build classifiers for clinical abbreviation disambiguation. The UAD model proposed in [15] creates word embeddings by using additional unstructured text. The work in [9] compares the averaged context vector of the words in a long-form of an acronym with the weighted average vector of the words in the context of the acronym based on word embeddings trained on a domain-specific corpus. In [26], the proposed model is trained to compute the similarity between a candidate long-form and the context surrounding the target acronym.

Many works utilize deep neural architectures to construct a classifier [16, 8, 4, 19, 20, 21]. At the AAAI-21 Workshop on Scientific Document Understanding (SDU@AAAI-21), the top ranked participants [20, 19, 21] present models for acronym disambiguation based on pre-trained language models such as RoBERTa [27] and SciBERT [28]. In [20], the problem of acronym disambiguation is treated as a span prediction problem, and the proposed model predicts the span containing the correct long-form from the concatenation of an input sentence and candidate long-forms of the acronym in the sentence. The hdBERT model proposed in [21] combines RoBERTa and SciBERT to capture both domain agnostic and domain specific information. The work in [19], which is the winner of the shared task of acronym disambiguation held under the workshop SDU@AAAI 2021, incorporates training strategies such as adversarial training [29] and task-adaptive pre-training [30]. Following a similar strategy to the recent works [19, 21], we develop a binary classification model for acronym disambiguation.

Category	# Sentences			# Acronyms			Avg. # Sentences per acronym		
	Train	Dev	Test	Train	Dev	Test	Train	Dev	Test
Legal English	2,949	385	383	242	31	30	12.186	12.419	12.767
Scientific English	7,532	894	574	405	52	40	18.598	17.192	14.350
French	7,851	909	813	541	68	60	14.512	13.368	13.550
Spanish	6,267	818	862	437	56	53	14.341	14.607	16.264
Total	24,599	3,006	2,632	1,625	207	183	15.138	14.522	14.383

Table 1
Statistics of the labeled datasets

2.2. Pre-trained Language Models

There has been significant progress across many natural language processing (NLP) tasks by the pre-trained language models trained on large-scale unlabeled corpora. Based on the transformer architecture [31], a set of large-scale pre-trained language models are developed, including BERT [17], RoBERTa [27], GPT [32] and T5 [18]. Since these models are pre-trained on datasets primarily consisting of English text, multilingual models such as mBERT [33] and mT5 [22] are presented. To process multilingual texts in the datasets published in the shared task for acronym disambiguation in the workshop SDU@AAAI-22, we use both T5 and mT5 to encode input texts.

2.3. Weak Supervision

Modern machine learning models generally need a large amount of hand-labeled training sets for performance improvement [34]. Since creating hand-labeled training datasets is time-consuming and expensive, recent works rely on weak supervision to generate noisy datasets [35, 36, 37, 38, 39, 40, 41, 42]. Distant supervision, one of the most popular techniques for weak supervision, utilizes external knowledge bases to produce noisy labels [35, 36, 43] Other works obtain noisy labels by using crowdsourcing [40, 41, 42] or simple heuristic rules [44, 37]. The system proposed in [39] automatically generates the heuristics to assign training labels to a large-scale unlabeled data. Similar to the works in [35, 36, 43] based on distant supervision, we use the relationships between acronyms and their possible long-forms as the weak supervision sources.

3. Acronym Disambiguation Model

We first provide the problem definition of acronym disambiguation. We next present the overall architecture and details of our proposed model.

3.1. Problem Definition

The problem of acronym disambiguation is defined as a classification problem [5]. Given a dictionary $\mathcal{A}$ which is a mapping of acronyms to candidate long-forms (or expansions), let $\mathcal{A}(\boldsymbol{a}) = \{\boldsymbol{e}_{\boldsymbol{a},1}, \ldots, \boldsymbol{e}_{\boldsymbol{a},m(\boldsymbol{a})}\}$ be the set of all candidate long-forms of an acronym $\boldsymbol{a}$, where $m(\boldsymbol{a})$ is the size of the set. Then, for an input sentence $\boldsymbol{s} = \langle w_1, w_2, \ldots, w_n \rangle$ consisting of n tokens (i.e., $w_1, \ldots, w_n$) and an acronym $\boldsymbol{a} = \langle w_i, \ldots, w_j \rangle$ with $1 \leq i \leq j \leq n$ which is a contiguous subsequence of $\boldsymbol{s}$, we want to predict the correct long-form of the acronym $\boldsymbol{a}$ among the candidate long-forms in $\mathcal{A}(\boldsymbol{a})$. Note that we represent a text as a sequence of tokens by using a tokenizer such as WordPiece [45] and SentencePiece [46]. Following the existing works [19, 21], we simplify the problem as a binary classification problem. In other words, given an input sentence $\boldsymbol{s}$, an acronym $\boldsymbol{a}$ appearing in $\boldsymbol{s}$ and a candidate long-form $\boldsymbol{e}_{\boldsymbol{a},k}$ in $\mathcal{A}(\boldsymbol{a})$, we predict the label y which is 1 if $\boldsymbol{e}_{\boldsymbol{a},k}$ is the correct long-form of $\boldsymbol{a}$ in the context of $\boldsymbol{s}$, and 0 otherwise.

3.2. Model Architecture

We provide an illustration of the proposed model in Figure 2. The model consists of an encoder, which transforms an input token sequence into a vector representation, and a multi-layer perceptron (MLP) with a sigmoid activation function to output the prediction. We use the pre-trained language models such as T5 [18] or mT5 [22] encoder layers to encode the input tokens, and take the hidden state of the first token as the encoder output. The encoder takes as input the concatenation of the input long-form $\boldsymbol{e}_{\boldsymbol{a},j}$ and the sentence $\boldsymbol{s}$ [19]. A separator symbol (i.e., [SEP]) is used to separate them. In other words, by using the symbol $\oplus$ to represent the concatenation of two token sequences, the input token sequence $\boldsymbol{x}$ of the encoder is defined as

$$\boldsymbol{x} = \boldsymbol{e}_{\boldsymbol{a},j} \oplus \langle \text{[SEP]} \rangle \oplus \boldsymbol{s}. \quad (1)$$

We also insert two special tokens [BOA] and [EOA] before and after the acronym $\boldsymbol{a}$ in $\boldsymbol{s}$ to highlight the position of the acronym. For example, consider the input

sentence containing the acronym "DP" and one of its candidate long-form "Dynamic Programming" in Figure 1. As shown in Figure 2, the encoder takes as input the token sequence obtained by concatenating "Dynamic Programming", [SEP] and the input sentence. The encoder converts the input token sequence $\boldsymbol{x}$ into a vector representation $h \in \mathbb{R}^d$, where d is the number of hidden units. The MLP layer is used to compute the prediction score p from h. That is,

$$p = \text{sigmoid}(W^T h + b), \tag{2}$$

where $W \in \mathbb{R}^d$ and $b \in \mathbb{R}$ are parameters of the MLP layer. We interpret p as the probability that the input long-form $\boldsymbol{e}_{\boldsymbol{a},j}$ is the correct long-form of the acronym $\boldsymbol{a}$ in $\boldsymbol{s}$.

Given a set of N sentences $\mathcal{S} = \{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_N\}$, let $\boldsymbol{a}_i$ be the acronym contained in the sentence $\boldsymbol{s}_i$. For every pair of a sentence $\boldsymbol{s}_i \in \mathcal{S}$ and a long-form $\boldsymbol{e}_{\boldsymbol{a}_i,j} \in \mathcal{A}(\boldsymbol{a}_i)$, we obtain an input token sequence $\boldsymbol{x}_{i,j}$ by Equation (1), as well as its corresponding label $y_{i,j}$. Thus, from the sentence set $\mathcal{S}$, we can build a training dataset $\mathcal{D} = \{(\boldsymbol{x}_{i,j}, y_{i,j}) \mid 1 \leq i \leq N, 1 \leq j \leq m(\boldsymbol{a}_i)\}$. We train the model on the training dataset $\mathcal{D}$. Let us denote the prediction score for $x_{i,j}$ by $p_{i,j}$. Then, we use the cross-entropy loss to train the model on the training dataset $\mathcal{D}$. In other words, the loss is defined as

$$\mathcal{L} = -\sum_{i=1}^{N} \sum_{j=1}^{m(\boldsymbol{a}_i)} (y_{i,j} \log p_{i,j} + (1 - y_{i,j}) \log (1 - p_{i,j})). \tag{3}$$

At the inference stage, for an input sentence $\boldsymbol{s}$ with an acronym $\boldsymbol{a}$, we compute the prediction score for each candidate long-form in $\mathcal{A}(\boldsymbol{a})$ and choose the one with the highest prediction score.

4. Datasets

We describe the labeled datasets published for the shared task on acronym disambiguation in the workshop SDU@AAAI-22 [47]. Moreover, we present the details of additional datasets generated by our weak supervision method.

4.1. Labeled Datasets

The detailed statistics of the labeled datasets is provided in Table 1. The datasets consist of four categories (i.e., Legal English, Scientific English, French and Spanish). In total, there are 24,599, 3,006 and 2,632 sentences in the training, development and test datasets, respectively. Every sentence in the datasets has a single ambiguous acronym which is to be disambiguated. On average, an acronym appears in 14 or 15 sentences. As mentioned in the web page (https://sites.google.com/view/sdu-aaai22/shared-task) of the competition on acronym disambiguation, for each category, there is no overlap of acronyms between any pair of the training, development and test datasets. Table 2 shows the statistics of the dictionary for every category. In the table, the "Avg. Fanout" indicates the average number of candidate long-forms for an acronym. A dictionary contains a mapping from an acronym to the set of its candidate long-forms. The number of occurrences of an acronym in the datasets of all categories is 2.866 on average.

Category	# LFs	# ACs	Avg. Fanout
Legal English	1,126	456	2.469
Scientific English	2,275	671	3.390
French	2,578	926	2.784
Spanish	1,859	682	2.726
Total	7,838	2,735	2.866

* LF: long-form, AC: acronym

Table 2
Statistics of the dictionaries

Category	L	L+W_1	L+W_5	L+W_{10}	L+W_{20}
Legal English	2,949	3,366	4,640	5,921	8,048
Scientific English	7,532	8,337	10,688	12,875	16,264
French	7,851	8,575	10,479	12,135	14,609
Spanish	6,267	6,980	9,036	10,922	13,788
Total	24,599	27,258	34,843	41,853	52,709

Table 3
Statistics of the labeled and weakly labeled datasets

4.2. Weakly Labeled Datasets

Among the acronyms in the dictionaries, 40.6% of them do not appear in the training dataset. To train the proposed model for such acronyms, we collect additional data by incorporating a weak supervision method [35]. Specifically, we first extract the sentences containing a long-form in the dictionaries from English, French and Spanish Wikipedia dump dated November 7, 2021. For each language, we do not use the long-form of every acronym whose number of occurrences is at least 1,000 in the Wikipedia dump, since the pre-trained language models are likely to be well-trained for such frequent long-forms. For each extracted sentence from Wikipedia, we replace the long-form in the sentence with its acronym. We next assign 1 as the label for the pair of the extracted sentence and the long-form, and 0 for every pair of the sentence and each of the other long-forms of the acronym.

Let N_s be the maximum allowed number of sentences extracted from the Wikipedia dumps for a long-form. For

Encoder	# Params	Legal English	Scientific English	French	Spanish	All
BERT-base-cased [17]	108M	69.74±3.21	65.37±0.79	64.68±0.98	66.64±0.97	66.02±0.42
T5E-base [18]	110M	66.94±1.60	64.31±1.02	66.42±0.72	68.14±1.08	66.32±0.73
BERT-large-cased [17]	334M	70.35±1.57	66.48±0.90	66.11±0.63	66.90±0.76	66.95±0.52
mT5E-base [22]	277M	67.47±3.37	62.47±0.62	69.09±1.24	72.88±2.50	67.90±1.59
RoBERTa-base [27]	125M	70.94±2.30	67.82±2.75	67.10±1.68	71.64±0.77	68.98±0.37
mBERT-base-cased [33]	178M	73.18±2.46	66.74±1.32	69.98±1.28	76.74±2.62	71.18±0.91
hdBERT [21]	472M	71.03±1.24	75.69±0.49	67.81±0.53	74.17±0.79	72.25±0.17
T5E-large [18]	335M	75.62±1.39	72.85±0.65	70.57±0.46	72.91±2.23	72.49±0.22
mT5E-large [22]	564M	72.83±0.90	69.62±0.37	72.11±1.18	78.35±1.00	73.09±0.51
mT5E-xlarge [22]	1,670M	75.44±2.03	70.92±0.88	72.49±0.51	**78.95**±0.88	74.08±0.57
T5E-xlarge [18]	1,241M	**78.73**±1.10	**77.56**±0.63	**72.69**±1.40	77.88±0.73	**76.24**±0.79

Table 4
F1 score with varying the encoder

each value of N_s in $\{1, 5, 10, 20\}$, we create a weakly labeled dataset. Let L and W_k denote the labeled dataset provided in the competition and the weakly labeled dataset generated with $N_s = k$. Then, we refer to the combination of the labeled dataset (L) and each of the weakly labeled datasets as L+W_1, L+W_5, L+W_{10} and L+W_{20}, respectively. The statistics of the combined datasets are presented in Table 3. As an example, when $N_s = 10$, we obtain 17,254 additional sentences containing an acronym in the dictionaries by weak supervision, and the ratio of unseen acronyms in the training dataset is reduced from 40.6% to 21.6%.

5. Experiments

We first present the experimental setup and next report the results of experiments including the competition for acronym disambiguation.

5.1. Experimental Setup

We conduct all experiments on a single machine with an AMD EPYC Rome 7402P 24-Core CPU and two NVIDIA GeForce RTX 3090 GPUs under PyTorch framework [48]. For each sentence, we consider a window of 64 tokens where the acronym in the sentence is located in the middle of the window, and use the sequence of tokens in that window for training. We set the batch size to 16 and use Adam optimizer [49]. Furthermore, we use the union of the training datasets with all categories to train implementations of the proposed model for 10 epochs with a learning late of 10^{-5}. Moreover, we apply dropout [50] to the encoder of the model with a dropout probability of 0.1.

To evaluate the performance of the model, we use macro-averaged precision (P), recall (R) and F1 score (F1) computed for each long-form [15, 5] on the development and test datasets. Specifically, we first compute precision, recall and F1 score for each long-form and then report the average value of all long-forms for each measure. Furthermore, for the development data, we report the average value with its standard deviation by training the models three times with different random seeds.

5.2. Experimental Results

Pre-trained models We compare the performance of the implementations of the proposed model with varying the pre-trained model of the encoder. We use BERT [17], mBERT [33], RoBERTa [27], hdBERT [21], T5 [18] and mT5 [22] as the encoder. Since pre-trained models with various model sizes are available for BERT and T5, we test them with varying the model size, too. While the default learning rate is 10^{-5}, we use a learning rate of 10^{-6} for hdBERT since we get a better performance with 10^{-6}.

Table 4 shows the F1 score on the development dataset for each category. The results show that the implementation with T5-xlarge achieves the highest performance in terms of the F1 score in every category except Spanish. The second best in terms of the F1 score for all categories is the implementation with mT5-xlarge as the encoder. Note that although T5 is pre-trained using English corpora, we can see that the model with the encoder of T5 is generalized well to the other languages. As the size of a model increases, the accuracy of the model tends to be improved. However, the implementation with T5-xlarge performs better than that with mT5-xlarge since T5 is pre-trained with supervised training, while mT5 is not. Note that we cannot evaluate the pre-trained models with a larger size such as T5-xxlarge and mT5-xxlarge models due to GPU memory limitations used in our experiment.

Weak supervision To confirm the effectiveness of the weakly labeled datasets, we train the proposed model

Data	P	R	F1
L	79.43 ± 0.68	73.30 ± 0.89	76.24 ± 0.79
L+W_1	81.05 ± 0.48	75.11 ± 0.50	77.97 ± 0.47
L+W_5	81.54 ± 0.61	74.50 ± 0.15	77.86 ± 0.32
L+W_{10}	**81.78** ± 0.76	**74.66** ± 0.77	**78.06** ± 0.76
L+W_{20}	81.14 ± 0.70	73.98 ± 0.33	77.40 ± 0.47

Table 5
Performance with the weakly labeled datasets

which uses T5-xlarge as the encoder on both the labeled and weakly labeled datasets with varying $N_s = 1, 5, 10, 20$. We provide the results in Table 5. Recall that we use L and Wk to denote the labeled dataset and the weakly labeled dataset generated with $N_s = k$ respectively, as described in Section 4. The table shows that the F1 score becomes larger with increasing the value of N_s for $N_s = 1, 5, 10$. However, when $N_s = 20$, the accuracy is degraded since the skewness of the number of sentences containing an acronym increases. In other words, as N_s increases, the number of the extracted sentences containing a frequent long-form becomes large, while that of the extracted sentences containing rare long-form does not. Since the model performs the best when $N_s = 10$, we set N_s to 10 as the default value.

Table 6 presents some examples which are classified incorrectly with the labeled dataset only, but are classified correctly after training on both labeled and weakly labeled datasets. The two rightmost columns show the prediction scores generated by the model trained using only the labeled dataset and using both the labeled and weakly labeled dataset with $N_s = 10$ (i.e., L+W_{10}), respectively. Without the weakly labeled dataset, as shown in the table, the model fails to find the correct long-forms for the sentences. However, by using the weakly labeled dataset, the prediction scores for the correct long-forms increase significantly.

Performance on the test dataset We evaluate the implementations of our model with T5-xlarge and mT5-xlarge as the encoder after training them on both the labeled and weakly labeled dataset. When we use T5-xlarge, we set the learning rate to 9×10^{-6} since we find that the model performs the best with that learning rate by a hyperparameter search. As shown in Table 7, in terms of the F1 score on the test dataset, the model with T5-xlarge performs the best for both Legal English and Scientific English datasets. On the other hand, the model with mT5-xlarge shows better performance than that with T5-xlarge for French and Spanish datasets. To further improve the performance of the best model in each category, we additionally train the best model by using only the dataset of the category for 5 epochs with a learning rate of 10^{-6}. The results show that the category-wise fine-tuning improves the accuracy for every category.

SDU@AAAI-22 Shared Task: Acronym Disambiguation In the competition, for each category, we use the model performed the best on the test dataset as shown in Table 7. The bolded numbers in the table are the scores of our model. The results show that our model ranks the 2nd place for Legal English and 3rd place for Scientific English and French.

6. Conclusion

We propose a binary classification model for acronym disambiguation by utilizing large-scale pre-trained language models. To increase the size of the training datasets, we use a weak supervision approach to generate weakly labeled datasets. Experimental results show that training on both labeled and weakly labeled datasets is beneficial to the accuracy of the proposed model. For the shared task on acronym disambiguation in the AAAI-22 Workshop on Scientific Document Understanding (SDU@AAAI-22), our model ranks within the 3rd place in three of four categories.

Acknowledgments

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. 2020-0-00857, Development of cloud robot intelligence augmentation, sharing and framework technology to integrate and enhance the intelligence of multiple robots). It was also supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. NRF-2020R1A2C1003576).

References

[1] W. Ammar, K. Darwish, A. El Kahki, K. Hafez, Icetea: in-context expansion and translation of english abbreviations, in: International Conference on Intelligent Text Processing and Computational Linguistics, Springer, 2011, pp. 41–54.

[2] A. Barnett, Z. Doubleday, Meta-research: The growth of acronyms in the scientific literature, Elife 9 (2020) e60080.

[3] R. Islamaj Dogan, G. C. Murray, A. Névéol, Z. Lu, Understanding pubmed® user search behavior through log analysis, Database 2009 (2009).

[4] Q. Jin, J. Liu, X. Lu, Deep contextualized biomedical abbreviation expansion, arXiv preprint arXiv:1906.03360 (2019).

Category	Sentence	Acronym	Correct expansion	Prediction (L)	Prediction (L+W_{10})
Legal English	Slovakia welcomes the establishment of UN Women – the UN-Women.	UN-Women	United Nations Entity for Gender Equality and Empowerment of Women	0.678190	0.929378
Legal English	There is no answer to the hopelessness and despair of the more than 30 million unemployed in the countries of the OECD.	OECD	Organization for Economic Cooperation and Development	0.202852	0.999734
Scientific English	The SGD is adopted to optimize the parameters.	SGD	stochastic gradient descent	0.368887	0.998205
Scientific English	Specifically, we will interpolate the translation models as in Foster and Kuhn (2007), including a MAP combination (Bacchiani et al 2006).	MAP	maximum a posteriori	0.184368	0.629905
French	Il est entouré au Nord par l'Ouganda, à l'Est par la Tanzanie, au Sud par le Burundi et à l'Ouest par la RDC.	RDC	République Démocratique du Congo	0.844930	0.999477
French	De plus, il y a un représentant spécial adjoint du Secrétaire général résident à Chypre avec le rang de SSG.	SSG	sous-secrétaire général	0.956114	0.998696
Spanish	En cuanto al FMAMI se sugirió que sería apropiado esperar hasta que se completara el debate actual sobre su reforma.	FMAM	Fondo para el Medio Ambiente Mundia	0.000304	0.999739
Spanish	El Gobierno del Japón acoge con beneplácito la NEPAD África que ha sido lanzada por los países africanos.	NEPAD	Nueva Alianza para el Desarrollo de	0.944804	0.990742

Table 6
Examples classified correctly by weak supervision in the development dataset

Category	Model	Development			Test		
		P	R	F1	P	R	F1
Legal English	T5-xlarge	86.13 ± 0.55	76.11 ± 1.67	80.80 ± 0.88	84.64	76.71	80.48
	mT5-xlarge	81.49 ± 1.62	72.22 ± 0.68	76.57 ± 0.36	82.95	72.80	77.54
	T5-xlarge-finetune	86.35 ± 0.21	78.16 ± 0.32	82.05 ± 0.24	**85.52**	**77.12**	**81.11**
Scientific English	T5-xlarge	81.72 ± 0.50	75.59 ± 1.15	78.54 ± 0.82	87.21	81.36	84.18
	mT5-xlarge	77.10 ± 2.58	67.00 ± 1.85	71.70 ± 2.16	82.85	75.62	79.07
	T5-xlarge-finetune	82.38 ± 0.40	76.23 ± 0.50	79.18 ± 0.44	**88.36**	**81.85**	**84.98**
French	T5-xlarge	79.00 ± 0.07	70.35 ± 0.14	74.43 ± 0.11	79.98	69.29	74.25
	mT5-xlarge	77.66 ± 1.26	68.17 ± 1.91	72.60 ± 1.48	80.71	70.42	75.21
	mT5-xlarge-finetune	77.39 ± 0.38	67.99 ± 0.45	72.39 ± 0.38	**80.79**	**72.20**	**76.25**
Spanish	T5-xlarge	86.08 ± 0.39	77.97 ± 1.57	81.83 ± 1.01	84.31	75.36	79.58
	mT5-xlarge	84.63 ± 3.29	78.83 ± 2.26	81.63 ± 2.69	86.27	76.16	80.90
	mT5-xlarge-finetune	86.55 ± 0.39	80.89 ± 0.17	83.63 ± 0.27	**86.33**	**76.51**	**81.12**

Table 7
Performance on the test dataset of each category

Legal English				Scientific English				French				Spanish			
Model	P	R	F1	Model	P	R	F1	Model	P	R	F1	Model	P	R	F1
Rank1	0.94	0.87	0.90	Rank1	0.97	0.94	0.96	Rank1	0.89	0.79	0.84	Rank1	0.91	0.85	0.88
Rank2	**0.86**	**0.77**	**0.81**	Rank2	0.95	0.90	0.93	Rank2	0.85	0.73	0.78	Rank2	0.88	0.79	0.83
Rank3	0.82	0.80	0.81	**Rank3**	**0.88**	**0.82**	**0.85**	**Rank3**	**0.81**	**0.72**	**0.76**	Rank3	0.86	0.80	0.83
Rank4	0.79	0.64	0.70	Rank4	0.81	0.77	0.79	Rank4	0.76	0.70	0.73	Rank4	0.83	0.80	0.81
Rank5	0.75	0.61	0.67	Rank5	0.81	0.69	0.75	Rank5	0.73	0.64	0.68	**Rank5**	**0.86**	**0.77**	**0.81**

Table 8
Leaderboard

[5] 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, arXiv preprint arXiv:2010.14678 (2020).

[6] M. Zahariev, Automatic sense disambiguation for acronyms, in: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, 2004, pp. 586–587.

[7] H. L. Fred, T. O. Cheng, Acronymesis: the exploding misuse of acronyms, Texas Heart Institute Journal 30 (2003) 255.

[8] A. G. Ahmed, M. F. A. Hady, E. Nabil, A. Badr, A language modeling approach for acronym expansion disambiguation, in: International Conference on Intelligent Text Processing and Computational Linguistics, Springer, 2015, pp. 264–278.

[9] J. Charbonnier, C. Wartena, Using word embeddings for unsupervised acronym disambiguation (2018).

[10] S. Pakhomov, T. Pedersen, C. G. Chute, Abbreviation and acronym disambiguation in clinical discourse, in: AMIA annual symposium proceedings, volume 2005, American Medical Informatics Association, 2005, p. 589.

[11] S. Moon, S. Pakhomov, G. B. Melton, Automated disambiguation of acronyms and abbreviations in clinical texts: window and training size considerations, in: AMIA annual symposium proceedings, volume 2012, American Medical Informatics Association, 2012, p. 1310.

[12] S. Moon, B. McInnes, G. B. Melton, Challenges and practical approaches with word sense disambiguation of acronyms and abbreviations in the clinical domain, Healthcare informatics research 21 (2015) 35–42.

[13] Y. Wu, J. Xu, Y. Zhang, H. Xu, Clinical abbreviation disambiguation using neural word embeddings, in: Proceedings of BioNLP 15, 2015, pp. 171–176.

[14] R. Antunes, S. Matos, Biomedical word sense disambiguation with word embeddings, in: International Conference on Practical Applications of Computational Biology & Bioinformatics, Springer, 2017, pp. 273–279.

[15] M. Ciosici, T. Sommer, I. Assent, Unsupervised abbreviation disambiguation contextual disambiguation using word embeddings, arXiv preprint arXiv:1904.00929 (2019).

[16] I. Li, M. Yasunaga, M. Y. Nuzumlalı, C. Caraballo, S. Mahajan, H. Krumholz, D. Radev, A neural topic-attention model for medical term abbreviation disambiguation, arXiv preprint arXiv:1910.14076 (2019).

[17] 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).

[18] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, arXiv preprint arXiv:1910.10683 (2019).

[19] C. Pan, B. Song, S. Wang, Z. Luo, Bert-based acronym disambiguation with multiple training strategies, arXiv preprint arXiv:2103.00488 (2021).

[20] A. Singh, P. Kumar, Scidr at sdu-2020: Ideas–identifying and disambiguating everyday acronyms for scientific domain, arXiv preprint arXiv:2102.08818 (2021).

[21] Q. Zhong, G. Zeng, D. Zhu, Y. Zhang, W. Lin, B. Chen, J. Tang, Leveraging domain agnostic and specific knowledge for acronym disambiguation., in: SDU@ AAAI, 2021.

[22] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, C. Raffel, mt5: A massively multilingual pre-trained text-to-text transformer, arXiv preprint arXiv:2010.11934 (2020).

[23] A. P. B. Veyseh, N. Meister, S. Yoon, R. Jain, F. Dernoncourt, T. H. Nguyen, Multilingual acronym extraction and disambiguation shared tasks at sdu 2022, in: Proceedings of SDU@AAAI-22, 2022.

[24] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa, Natural language processing (almost) from scratch, Journal of machine learning research 12 (2011) 2493–2537.

[25] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in neural information processing systems, 2013, pp. 3111–3119.

[26] K. Kirchhoff, A. M. Turner, Unsupervised resolution of acronyms and abbreviations in nursing notes using document-level context models, in: Proceedings of the Seventh International Workshop on Health Text Mining and Information Analysis, 2016, pp. 52–60.

[27] 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).

[28] I. Beltagy, K. Lo, A. Cohan, Scibert: A pretrained language model for scientific text, arXiv preprint arXiv:1903.10676 (2019).

[29] T. Miyato, A. M. Dai, I. Goodfellow, Adversarial training methods for semi-supervised text classification, arXiv preprint arXiv:1605.07725 (2016).

[30] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, N. A. Smith, Don't stop pretraining: adapt language models to domains and

tasks, arXiv preprint arXiv:2004.10964 (2020).
[31] 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.
[32] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, arXiv preprint arXiv:2005.14165 (2020).
[33] J. Devlin, Multilingual bert readme, https://github.com/google-research/bert/blob/master/multilingual.md, 2018.
[34] C. Sun, A. Shrivastava, S. Singh, A. Gupta, Revisiting unreasonable effectiveness of data in deep learning era, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 843–852.
[35] 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, 2009, pp. 1003–1011.
[36] A. Go, R. Bhayani, L. Huang, Twitter sentiment classification using distant supervision, CS224N project report, Stanford 1 (2009) 2009.
[37] A. Ratner, S. H. Bach, H. Ehrenberg, J. Fries, S. Wu, C. Ré, Snorkel: Rapid training data creation with weak supervision, in: Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases, volume 11, NIH Public Access, 2017, p. 269.
[38] A. Ratner, B. Hancock, J. Dunnmon, R. Goldman, C. Ré, Snorkel metal: Weak supervision for multi-task learning, in: Proceedings of the Second Workshop on Data Management for End-To-End Machine Learning, 2018, pp. 1–4.
[39] P. Varma, C. Ré, Snuba: Automating weak supervision to label training data, in: Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases, volume 12, NIH Public Access, 2018, p. 223.
[40] N. Dalvi, A. Dasgupta, R. Kumar, V. Rastogi, Aggregating crowdsourced binary ratings, in: Proceedings of the 22nd international conference on World Wide Web, 2013, pp. 285–294.
[41] Y. Zhang, X. Chen, D. Zhou, M. I. Jordan, Spectral methods meet em: A provably optimal algorithm for crowdsourcing, Advances in neural information processing systems 27 (2014) 1260–1268.
[42] M. Joglekar, H. Garcia-Molina, A. Parameswaran, Comprehensive and reliable crowd assessment algorithms, in: 2015 IEEE 31st International Conference on Data Engineering, IEEE, 2015, pp. 195–206.
[43] E. Alfonseca, K. Filippova, J.-Y. Delort, G. Garrido, Pattern learning for relation extraction with hierarchical topic models (2012).
[44] A. J. Ratner, C. M. De Sa, S. Wu, D. Selsam, C. Ré, Data programming: Creating large training sets, quickly, Advances in neural information processing systems 29 (2016) 3567–3575.
[45] R. Sennrich, B. Haddow, A. Birch, Neural machine translation of rare words with subword units, arXiv preprint arXiv:1508.07909 (2015).
[46] T. Kudo, J. Richardson, Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing, arXiv preprint arXiv:1808.06226 (2018).
[47] A. P. B. Veyseh, N. Meister, S. Yoon, R. Jain, F. Dernoncourt, T. H. Nguyen, Macronym: A large-scale dataset for multilingual and multi-domain acronym extraction, arXiv preprint arXiv:1412.6980 (2022).
[48] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, Advances in neural information processing systems 32 (2019) 8026–8037.
[49] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).
[50] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, The journal of machine learning research 15 (2014) 1929–1958.