

Ensemble Approach to Acronym Extraction using Transformers

Prashant Sharma^{*1}, Hadeel Saadany^{*2}, Leonardo Zilio²,
Diptesh Kanojia², Constantin Orăsan²

¹Hitachi CRL, Japan.

²Centre for Translation Studies, University of Surrey, United Kingdom.
prashaantsharmaa@gmail.com, h.saadany@surrey.ac.uk

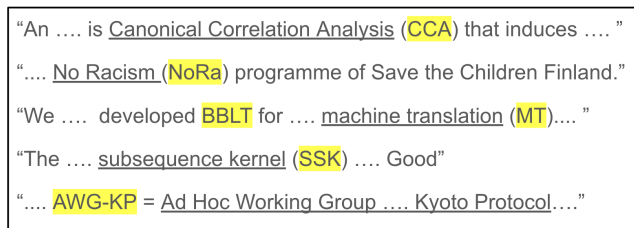
Abstract

Acronyms are abbreviated units of a phrase constructed by using initial components of the phrase in a text. Automatic extraction of acronyms from a text can help various Natural Language Processing tasks like machine translation, information retrieval, and text summarisation. This paper discusses an ensemble approach for the task of Acronym Extraction, which utilises two different methods to extract acronyms and their corresponding long forms. The first method utilises a multilingual contextual language model and fine-tunes the model to perform the task. The second method relies on a convolutional neural network architecture to extract acronyms and append them to the output of the previous method. We also augment the official training dataset with additional training samples extracted from a series of open-access journals to help improve the task performance. Our dataset analysis also highlights the noise within the current task dataset. Our approach achieves the following macro-F1 scores on test data released with the task: Danish (0.74), English [Legal] (0.72), English [Scientific] (0.73), French (0.63), Persian (0.57), Spanish (0.65), Vietnamese (0.65). We release our code and models publicly¹.

1 Introduction

Acronyms are commonly used to shorten known units of text in various domains such as scientific (Pustejovsky et al. 2001), medical (Dannélls 2006), business (Ménard and Ratté 2011) and legal (Tsimpouris, Sgarbas, and Panagiotopoulou 2015). Humans can usually identify acronyms in a text without too much difficulty by relying on various surface clues. Without knowing the meaning of acronyms, translators and interpreters cannot translate a text reliably.

Automatic identification of acronyms and their corresponding long forms is a relevant issue in the domain of Natural Language Processing (NLP) as it can help tasks such as information extraction and retrieval (Sánchez and Isern 2011; Ballesteros and Croft 1996), machine translation (Kirchhoff and Turner 2016) and also machine interpreting (Braun 2019). Acronyms represent relevant parts of



"An is Canonical Correlation Analysis (CCA) that induces "

".... No Racism (NoRa) programme of Save the Children Finland."

"We developed BBLT for machine translation (MT).... "

"The subsequence kernel (SSK) Good"

".... AWG-KP = Ad Hoc Working Group Kyoto Protocol...."

Figure 1: Examples of Long Forms and their Acronyms

the text and can confuse translation models. Thus, the task of automatically identifying and extracting acronyms from a text can be challenging for a machine. The shared task-1 for acronym extraction (AE) under the Scientific Document Understanding (SDU) workshop allows researchers to tackle this challenge and propose novel ways to solve it. The task requires participants to submit systems that can automatically identify acronyms and their long forms from a given piece of text. Figure 1 shows two examples of acronyms and their long forms from the English dataset (Scientific domain) and English dataset (Legal domain). These examples are from the task dataset provided as-is to the participants. The acronyms in parenthesis are highlighted in yellow, and their respective long forms are underlined. Automatic acronym extraction can be particularly challenging and based on our analysis of the training data for this task (Amir Pouran Ben Veyseh 2022b), we observe:

- Acronyms are known to be present in uppercase letters in the text, but there can be instances where they can contain lowercase letters (*e.g.*, CATiB, SEqui, OEql, SRals ing)
- There can be instances where the long forms of the acronyms are not present in the text (*e.g.*, We originally developed BBLT for ourselves..)
- There can be different long forms for the same acronym (*e.g.*, CT has been used as an acronym for both "contract" and "certain")
- Acronyms can have special characters present in them (*e.g.*, Communist Party of the Philippines-New Peoples Army-National Democratic Front (CPPNPA-NDF))
- Multiple acronymised letters are part of the same word (*e.g.*, subsequence kernel (SSK), maximum entropy (MaxEnt))

^{*}These authors contributed equally.

Copyright © 2022, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

¹<https://github.com/dipteshkanojia/PR-AAAI22-SDU-ST1-AE>

Apart from the challenges discussed here, we also noted noise in the dataset, which we discuss in the dataset analysis subsection later (Section 3). However, for the challenges discussed here, rule-based approaches fail, in particular, as they try to generalise over a pattern or a regular expression to detect acronyms from the text. There are multiple outliers that cannot be detected with the help of such approaches, as can be seen in the results of the rule-based approach implemented by the organisers of this task as a baseline. We discuss this approach in brief in Section 4. For this reason, we focused our efforts to develop a data-driven approach (more precisely a deep learning-based approach) to extract acronyms from the multilingual dataset prepared by the organisers of the shared task.

In this paper, we describe our efforts to create a system that can extract acronyms and their long forms from a multilingual dataset. We model the task of acronym extraction as a sequence labelling problem and perform token classification considering each dataset sample as a sequence. After experimenting with several architectures, we decided to use an Ensemble approach which relies on two methods. The first method utilises the Transformer-architecture-based multilingual language model to perform fine-tuning and extract acronyms. We also perform the acronym extraction task with the help of a Convolutional Neural Network (CNN), which employs word embeddings from a different source, as described later in our work. The resultant outputs from both these methods are then combined to create an ensemble output which we submit to obtain our scores for the task.

2 Related Work

The task of extracting acronyms from text has been performed in different domains for English, with most of the approaches being rule-based approaches (Taghva and Gilbreth 1999; Yeates 1999; Park and Byrd 2001; Larkey et al. 2000). Schwartz and Hearst (2002) implemented an algorithm for identifying acronyms by using parenthetical expressions as a marker of a short form. Their work is based on a previous work by Pustejovsky et al. (2001) which also extracts acronyms using a similar method. Dannélls (2006) performs the extraction of acronym-definition pairs from Swedish medical texts by primarily using a rule-based approach to extract acronyms and then a memory-based supervised machine learning approach to compare and evaluate the results. A rule-based approach was also implemented by Okazaki and Ananiadou (2006) for term recognition, and it discusses the extraction of acronyms and their long forms. This system mines acronyms based on parenthetical expressions as a marker of a short form as previous methods had described. However, for mining long forms, they created a candidate list based on frequent co-occurrences of word sequences. Movshovitz-Attias and Cohen (2012) investigate the use of Hidden Markov Model (HMM) for the extraction of acronyms from text. Ehrmann et al. (2013) show how acronym recognition patterns, initially developed for medical terms, can be adapted to the more general news domain. Their efforts led to automatically merging the numerous long-form variants referring to the same short form while maintaining non-related long forms separately. Their

	Training	Development	Test
Danish	3082	385	386
English (Legal)	3564	445	446
English (Scientific)	3980	497	498
French	7783	973	973
Persian	1336	167	168
Spanish	5928	741	741
Vietnamese	1274	159	160

Table 1: Dataset Statistics, in terms of number of dataset samples, for the Acronym Extraction task as provided by the task organizers.

work is based on the algorithm developed by Schwartz and Hearst (2002), but they perform the task of acronym extraction for 22 languages.

Machine learning-based approaches for the extraction of acronyms have been utilised in many previous studies (Nadeau and Turney 2005; Kuo et al. 2009). With the advancement of research in NLP, various methods to extract word embeddings for text have been proposed, the most recent of them being contextual language models. To detect acronyms without local definitions, Rogers, Rae, and Demner-Fushman (2021) applied two deep learning approaches: bi-directional LSTM with CRF and Transformer models. Li et al. (2021) utilise transformer-based architecture for modelling the task of acronym identification as a sentence-level sequence labelling problem. Zhu et al. (2021) incorporate the FGM adversarial training strategy for fine-tuning BERT for robust and generalised acronym identification. This was the winning system for the Acronym Extraction Shared Task held at SDU workshop in 2021.

In this paper, we employ the previously proposed approach of fine-tuning a language model for the task of acronym extraction (Kubal and Nagvenkar 2021). However, we extract additional data from PLOS journals and perform additional data analysis. We describe data augmentation and pre-processing techniques in the upcoming sections.

3 Dataset

The dataset provided by the task organisers consists of text data in Danish, English, French, Persian, Spanish, and Vietnamese languages in the JSON format (Amir Pouran Ben Veyseh 2022a). The English dataset is further divided into two different domains: legal and scientific. The dataset statistics can be seen in Table 1. However, each text sample can contain multiple lines of the text, thus containing up to 1050 words in each sample as observed from the training dataset.

Dataset Preprocessing

Our approach to model the task as a sequence labelling problem required us to convert the text provided into a BIO (short for beginning, inside, outside) format. BIO is a common format for tagging a token in a chunking or a named entity recognition task in computational linguistics (Ramshaw

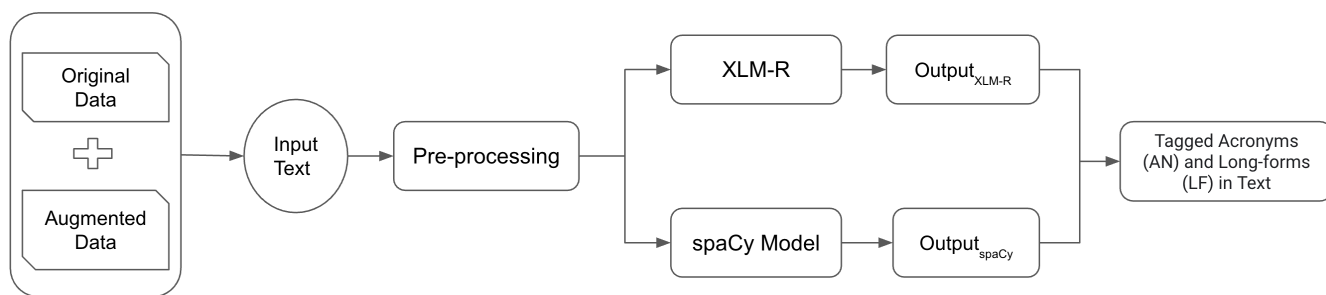


Figure 2: System Architecture for Our Ensemble Approach

and Marcus 1999). We convert each JSON task dataset into the BIO format with the help of a custom script written in Python. The custom tags we use to convert each token in the sequence resemble Named Entity tags and result in the following list of tags: [O (Outside), B-AN (Begin Acronym), I-AN (Inside Acronym), B-LF (Begin Long Form), I-LF (Inside Long Form)].

Dataset Analysis

We analysed the dataset provided for the task and observed the following issues:

- There are instances with missing or incomplete annotation (e.g. the acronym SDI has associated with the long-form “selective dissemination” instead of “selective dissemination of information”, which is present in the instance).
- Segments with wrong annotation (e.g. “US\$ 3” was annotated as an acronym of “US\$ 3,000”, when these are actually conversions between two currencies).
- Segments with over-annotation (e.g. the acronym “(GHS)” had the brackets included in the annotation).

In an analysis performed on the first 100 instances of the English scientific training dataset, we found 28 instances with such issues. These issues were also found in 21 instances among the first 100 instances of the English legal training dataset. The presence of such a high number of errors in the dataset poses some serious challenges to any data-driven method.

Data Augmentation

To increase the amount of training data, a separate training dataset was created by using information extracted from the PLOS² open-access journal publications. The XML versions of these publications are freely distributed along with the PMC Open Access Subset³ and amount to 305,445 texts.

The XML files in the PLOS corpus have a section for abbreviations, where all abbreviations used in a paper are made explicit along with their long forms. This information was used to extract sentences from research articles within the PLOS corpus. For the augmented dataset, only

sentences containing at least one abbreviation with a related long-form were included (other abbreviations with or without long forms could be present as well). This augmented training dataset contained a total of approx. 93k samples for English (Scientific domain).

4 Our Approach

The baseline approach provided with the task uses a single rule, i.e., if the word inside a parenthesis contains more than 60% uppercase letters, it is to be identified as an acronym. Moreover, the number of uppercase letters in this acronym form a sliding window for words before/after the acronym. If each uppercase letter matches the first characters of words in the sliding window, the words constitute the long-form phrase. This approach clearly fails to address the challenges discussed in Section 1 of the paper. The resulting macro-F1 scores for the **baseline approach on the development** data are: English (Legal) - 0.1258, English (Scientific) - 0.1084, Danish - 0.0950, French - 0.0806, Spanish - 0.0831, Persian - 0.4437, Vietnamese - 0.3538. Hence, we needed to look for alternative methods for identifying acronyms and their long forms. After considering a number of options, we decided to experiment with a deep-learning-based ensemble approach.

Our Ensemble Approach: We concatenate all the multilingual task data into a single training dataset. This was then concatenated with the PLOS data described above to increase the train data size to approx. 113k dataset samples. This data is used to fine-tune the multilingual language model described below to perform the acronym extraction (AE) task. We obtain the test output from two different methods described below and concatenate them. The fine-tuning method described below is able to label both acronyms and long forms. The spaCy blank model-based method described below is only able to obtain acronyms for the task but helps achieve improved F1 scores for the task. The architecture for our approach is shown in Figure 2.

Fine-tuning with XLM-RoBERTa

XLM-RoBERTa (XLM-R) (Conneau et al. 2019) is a multilingual contextualised Language Model (LM) pre-trained on filtered CommonCrawl data from 100+ languages. Each language dataset from this task is included in this model⁴.

²<https://plos.org/>.

³<https://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/>.

⁴<https://github.com/facebookresearch/XLM>

Our approach utilises this transformer architecture-based pre-trained LM and fine-tunes it for the downstream sequence labelling task. The LM used for our approach is XLM-R_{base}, and has approximately 270M parameters with 12-layers, 768 hidden states, 3072 feed-forward hidden states, 8 heads; and is pre-trained on CommonCrawl data in over 100 languages. The fine-tuning process adds a hidden linear layer on top of the pre-trained LM and projects the output to a *softmax* layer for token classification. We perform further hyperparameter tuning as described below.

We observed that during the inference phase, the output token length was truncated to 128 tokens as it was the default ‘maximum sequence length’ parameter. To preserve the entire token length, the two parameters, namely, ‘sliding window’ and ‘maximum sequence length’ were being used with the original model. The sliding window prevents the truncation of sentences by splitting the input sequence into multiple windows if it exceeds the default maximum sequence value. The sliding window problem represents the broken contextual information while predicting token class, and hence it was not used. We carried out multiple experiments with the ‘maximum sequence length’ parameter and observed that the model performed the best when it was limited to 512. The other values for maximum sequence length we experimented with were 128, 256, 350, 450, and 512. The language model was fine-tuned by using the fairSeq (Ott et al. 2019) library. We used an NVIDIA Quadro RTX 5000 GPU with 16 GB of memory for carrying out these experiments. This approach could classify tokens as B-AN, I-AN, B-LF and I-LF in the text for us. With the help of further post-processing using a custom script in Python, we were able to convert the predictions in the JSON format as required for the evaluation phase of the task.

spaCy Blank Model

The other model that we used for AE was a spaCy v3.2⁵ blank NER model. The spaCy v3.2 model is based on predicting context-sensitive vectors for each word in the input by a token-to-vector model. The embeddings in this model are obtained from the Bloom embeddings where each sub-word is transferred into a string of fixed symbols (e.g. 0-9 integer transferred to letter d, capital letter to W and lower case letter to w) (Serrà and Karatzoglou 2017). This strategy has proven to be significantly effective in handling out-of-vocabulary (OOV) tokens; instead of dumping all OOVs in one bucket, each OOV is given a unique representation. The method offers a variety of neural architectures for building a blank NER model to predict task-tailored entities. We opted for the trigram-CNN architecture learning via a transition-based approach which takes a window of the embeddings on either side of each word in the sentence and concatenates them in a multi-layered perceptron followed by an attention layer (Lample et al. 2016). A Maxout Unit (Goodfellow et al. 2013) is used as an activation function that calculates the ‘maximum’ of the inputs. These architecture parameters have performed well for NER tasks⁶.

⁵<https://spacy.io/models>

⁶<https://v2.spacy.io/usage/facts-figures>

	F1	P	R
Danish	0.74	0.78	0.70
English (Legal)	0.72	0.75	0.69
English (Scientific)	0.73	0.77	0.69
French	0.63	0.68	0.59
Persian	0.57	0.64	0.51
Spanish	0.65	0.65	0.65
Vietnamese	0.65	0.64	0.66

Table 2: Results obtained using our ensemble approach over the test data as provided for the task where P is Precision, R is Recall, and F1 is the Macro-F1 score as used for the task.

For training, we use the English (Scientific) and English (Legal) training sets consisting of 7523 instances and evaluate our method on the English (Scientific) and English (Legal) development sets. We use a limited training dataset as these experiments are performed with CPU cores. Before training, we pre-processed the data to conform with the ‘spacy’ format, where each positive instance was assigned the NER label acronym along with its specific indices. For the spaCy pipeline parameters, we chose the spaCy en_core_web_sm model, which is the small model trained on written English language web text (blogs, news, comments) including vocabulary, vectors, syntax and entities. We trained this model on a CPU with 100 iterations and with a batch size of 1000.

We used this model as a zero-shot model for the AE task with the test sets of the other four languages during the final stage of the shared task. A note on the reason for choosing this blank spaCy model for AE in this task is that it has a CPU-optimised pipeline, and it is much cheaper to run than pre-trained models. Due to its competitive results to the more expensive pre-trained models, we plan to explore training a spaCy model with more data for future AE task.

5 Results and Discussion

Using the fine-tuned XLM-R model, we obtained acronyms and long forms for the test data provided for this task. We then concatenated this output with the output from the spaCy blank model. The results obtained for the final test set output are present in Table 2. Our approach was outranked by several other systems submitted for the task, but we show a significantly improved set of results over the baseline method proposed for the task. Our results also show how the AE task can be modelled as a sequence labelling problem, thus utilising pre-existing architecture for the NER problem in NLP. The performance of our approach in comparison to other systems submitted at the task was comparatively lower, which can be attributed to the fact that we use a single multilingual training model for all the languages. We also use a rather simple fine-tuning based approach and do not add a more sophisticated neural networks-based architecture.

The output obtained using the fine-tuning method described above outperforms the rule-based baseline approach

by a significant margin. This method helps our approach gain significant percentage points for both acronyms and long forms as compared to the baseline approach. However, we observed that this method did not recognise many acronyms, resulting in low recall values. We also observed that due to the noise present in the data, this method tagged special characters like ‘)’ as a part of the acronym. We also observed that the fine-tuning process tagged a lot of *stop-words* as long forms even when they were not a part of any long-form sequence (e.g., for, the, of). We post-process the output of this model to rectify such errors.

The performance of the spaCy method, however, is significantly better at extracting acronyms as it uses a CNN architecture and performs well for the English language. However, it needs to be pointed out that, despite the fact that the dataset was multilingual in nature, most instances of long forms and acronyms were present in the English language. The results of the spaCy model for the extraction of only acronyms against the development set were: English Scientific - Precision: 0.8847, Recall: 0.7990, F1: 0.8397; and English Legal - Precision: 0.9168, Recall: 0.7354 and F1: 0.8161. As noted, this method only extracts acronyms and does not work well with long forms.

In most of the cases, we observed that our approach obtains higher precision than recall. This observation is expected as the fine-tuning process expects a lot more ‘O’ tokens compared to the AN or the LF classes. Our approach, which classifies each token, confuses a lot of ‘AN’s as ‘O’s. In fact, when the output of the spaCy model was deconcatenated, we observed that the XLM-R based method for the Persian language had only achieved an F1 of 0.27 compared to the overall F1 of 0.57. The task performance of our approach on the Spanish test data, however, is an exception as it shows a steady Precision, Recall, and F1 scores of 0.65.

6 Conclusions and Future Work

In this paper, we propose a deep-learning-based approach to extract acronyms and long forms from the data provided for the task. We discuss the problem of acronym extraction and show how challenging it is to accomplish this task automatically. The dataset provided for the task is multilingual in nature, and our approach attempts to build a single model which can handle all the languages. However, our dataset analysis shows that the data provided for the task should indeed be manually validated first to remove the noise. We also augmented this dataset with more samples from the PLOS open-access journal to improve the dataset size, but it is unclear how much this helped improve the performance. We modelled this AE task as a sequence labelling problem and used an ensemble approach which utilises two different methods: (1) based on fine-tuning the XLM-R model to extract ANs and LFs, and (2) based on using a CNN architecture provided by spaCy blank modelling method. Our results significantly outperform the baseline results and show that this approach does work for the AE task. We release the code and the models used for this task⁷.

⁷<https://github.com/dipteshkanojia/PR-AAAI22-SDU-ST1-AE>

For the future, we aim to perform this multilingual AE task by separating the models individually for each language. We plan to use the data from PLOS to augment each training dataset and perform further experiments. We also plan to collect more data for each language for the task and augment it with the training data for each of the models. With this augmented resource, we plan to perform an extensive analysis of the acronym extraction task and present our findings in the near future. Our eventual goal is to perform exhaustive experimentation with various datasets/methods and empirically find the best performing approach for this task.

References

- Amir Pouran Ben Veyseh, S. Y. R. J. F. D. T. H. N., Nicole Meister. 2022a. MACRONYM: A Large-Scale Dataset for Multilingual and Multi-Domain Acronym Extraction. In *arXiv*.
- Amir Pouran Ben Veyseh, S. Y. R. J. F. D. T. H. N., Nicole Meister. 2022b. Multilingual Acronym Extraction and Disambiguation Shared Tasks at SDU 2022. In *Proceedings of SDU@AAAI-22*.
- Ballesteros, L.; and Croft, B. 1996. Dictionary methods for cross-lingual information retrieval. In *International Conference on Database and Expert Systems Applications*, 791–801. Springer.
- Braun, S. 2019. Technology and interpreting. In *The Routledge Handbook of translation and technology*, 271–288. Routledge.
- Conneau, A.; Khandelwal, K.; Goyal, N.; Chaudhary, V.; Wenzek, G.; Guzmán, F.; Grave, E.; Ott, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.
- Dannélls, D. 2006. Automatic Acronym Recognition. In *Demonstrations*, 167–170.
- Ehrmann, M.; Della Rocca, L.; Steinberger, R.; and Tanev, H. 2013. Acronym recognition and processing in 22 languages. *arXiv preprint arXiv:1309.6185*.
- Goodfellow, I.; Warde-Farley, D.; Mirza, M.; Courville, A.; and Bengio, Y. 2013. Maxout networks. In *International conference on machine learning*, 1319–1327. PMLR.
- Kirchhoff, K.; and Turner, A. M. 2016. 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*, 52–60.
- Kubal, D. R.; and Nagvenkar, A. 2021. Effective Ensembling of Transformer based Language Models for Acronyms Identification. In *SDU@ AAAI*.
- Kuo, C.-J.; Ling, M. H.; Lin, K.-T.; and Hsu, C.-N. 2009. BIOADI: a machine learning approach to identifying abbreviations and definitions in biological literature. In *BMC bioinformatics*, volume 10, 1–10. BioMed Central.
- Lample, G.; Ballesteros, M.; Subramanian, S.; Kawakami, K.; and Dyer, C. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

- Larkey, L. S.; Ogilvie, P.; Price, M. A.; and Tamilio, B. 2000. Acrophile: an automated acronym extractor and server. In *Proceedings of the fifth ACM conference on Digital libraries*, 205–214.
- Li, F.; Mai, Z.; Zou, W.; Ou, W.; Qin, X.; Lin, Y.; and Zhang, W. 2021. Systems at SDU-2021 Task 1: Transformers for Sentence Level Sequence Label. In *SDU@ AAAI*.
- Ménard, P. A.; and Ratté, S. 2011. Classifier-based acronym extraction for business documents. *Knowledge and information systems*, 29(2): 305–334.
- Movshovitz-Attias, D.; and Cohen, W. 2012. Alignment-HMM-based extraction of abbreviations from biomedical text. In *BioNLP: Proceedings of the 2012 Workshop on Biomedical Natural Language Processing*, 47–55.
- Nadeau, D.; and Turney, P. D. 2005. A supervised learning approach to acronym identification. In *Conference of the Canadian Society for Computational Studies of Intelligence*, 319–329. Springer.
- Okazaki, N.; and Ananiadou, S. 2006. A Term Recognition Approach to Acronym Recognition. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, 643–650. Sydney, Australia: Association for Computational Linguistics.
- Ott, M.; Edunov, S.; Baevski, A.; Fan, A.; Gross, S.; Ng, N.; Grangier, D.; and Auli, M. 2019. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.
- Park, Y.; and Byrd, R. J. 2001. Hybrid text mining for finding abbreviations and their definitions. In *Proceedings of the 2001 conference on empirical methods in natural language processing*.
- Pustejovsky, J.; Castano, J.; Cochran, B.; Kotecki, M.; and Morrell, M. 2001. Automatic extraction of acronym-meaning pairs from MEDLINE databases. In *MEDINFO 2001*, 371–375. IOS Press.
- Ramshaw, L. A.; and Marcus, M. P. 1999. *Text Chunking Using Transformation-Based Learning*, 157–176. Dordrecht: Springer Netherlands. ISBN 978-94-017-2390-9.
- Rogers, W.; Rae, A. R.; and Demner-Fushman, D. 2021. AI-NLM exploration of the Acronym Identification Shared Task at SDU@ AAAI-21. In *SDU@ AAAI*.
- Sánchez, D.; and Isern, D. 2011. Automatic extraction of acronym definitions from the Web. *Applied Intelligence*, 34(2): 311–327.
- Schwartz, A. S.; and Hearst, M. A. 2002. A simple algorithm for identifying abbreviation definitions in biomedical text. In *Biocomputing 2003*, 451–462. World Scientific.
- Serrà, J.; and Karatzoglou, A. 2017. Getting deep recommenders fit: Bloom embeddings for sparse binary input/output networks. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, 279–287.
- Taghva, K.; and Gilbreth, J. 1999. Recognizing acronyms and their definitions. *International Journal on Document Analysis and Recognition*, 1(4): 191–198.
- Tsimpouris, C.; Sgarbas, K.; and Panagiotopoulou, S. 2015. Acronym identification in Greek legal texts. *Digital Scholarship in the Humanities*, 30(3): 440–451.
- Yeates, S. A. 1999. Automatic Extraction of Acronyms from Text. In *New Zealand Computer Science Research Students' Conference*, 117–124. Citeseer.
- Zhu, D.; Lin, W.; Zhang, Y.; Zhong, Q.; Zeng, G.; Wu, W.; and Tang, J. 2021. AT-BERT: Adversarial Training BERT for Acronym Identification Winning Solution for SDU@ AAAI-21. *arXiv preprint arXiv:2101.03700*.