AI-NLM exploration of the Acronym Identification Shared Task at SDU@AAAI-21

Willie Rogers, Alastair Rae, Dina Demner-Fushman
National Library of Medicine
8600 Rockville Pike
Bethesda, MD, 20894
{wjrogers,raear,ddemner}@mail.nih.gov

Abstract

National Library of Medicine has developed systems for recognition of named entities in biomedical and clinical text. The systems are primarily leveraging the Unified Medical Language System (UMLS) to recognize the terms and link them to the terminology part of the UMLS (Metathesaurus.) Biomedical and clinical texts are rife with acronyms and abbreviations. Acronym identification and disambiguation play, therefore, an important role in processing of the text using the UMLS-based approaches. To test the existing rule-based approaches developed at NLM and to explore the state-of-the-art DL approaches, we participated in the SDU Acronym Identification shared task. Not surprisingly, our existing rule-based approach achieved high precision (over 96%), but had very low recall, whereas, the LSTM and BERT-based approaches had almost equal recall and precision and achieved F1 scores in the low 90s.

Introduction

One of the major problems in machine understanding of the biomedical and clinical text is disambiguation of acronyms and abbreviations. In the scientific literature, acronyms are often introduced along with the full form of the term, for example, Coronavirus Disease 2019 (COVID-19). Full terms are not provided when the term is well known and relatively unambiguous, such as HIV or NSAIDs. This observation led to implementation of algorithms that leverage the full term to derive the meaning of the acronym in the local context (Schwartz and Hearst 2002; Aronson 1996; Zhou, Torvik, and Smalheiser 2006). Clinical notes, on the other hand, almost never contain full terms and, unlike the scientific papers, can use the same acronyms for different terms in different parts of a single note. For example, BS could denote Breath Sound, Bowel Sound or Blood Sugar levels, and only the sections of the note can help disambiguate the acronym. Not surprisingly, there is a large body of work on resolving the biomedical acronyms, most recently, employing neural approaches (Li et al. 2019; Joopudi, Dandala, and Devarakonda 2018; Wu et al. 2017).

Our participation in the task was motivated by the general purpose biomedical named entity recognition system,

Parameter	Value
dimension	300
dimension char	100
dropout	0.5
num oov buckets	1
training epochs	25
batch size	20
buffer	15000
char lstm size*	25
kernel size**	3
lstm size	100
minimum steps	8000

Table 1: Hyperparameters for training the Bi-directional LSTM models. ***** Used only in Stacked Bi-LSTM-CRF with and without EMA; ****** Used to set kernel size in the convolutional models.

MetaMap, developed at the National Library of Medicine. There are currently two implementations of MetaMap: a Prolog-based version (Aronson and Lang 2010) that has accrued multiple processing options over the years and a lightweight Java implementation (Demner-Fushman, Rogers, and Aronson 2017) intended to facilitate inclusion of the tool in the local clinical text processing. Both versions of MetaMap implement a rule-based acronym disambiguation algorithm that relies on the presence of the full form of the term. Participating in the shared task gave us an opportunity to evaluate this algorithm, and also to explore the state-of-the-art approaches that we previously used for word sense disambiguation and other tasks. To train and validate our approaches, we used the data developed for the Acronyms Identification task (Pouran Ben Veyseh et al. 2020b). The task is described in the overview provided by the organizers (Pouran Ben Veyseh et al. 2020a).

Methods

We tried both the above algorithmic and machine learning methods in the acronym identification task.

We initially attempted acronym identification using the MetaMap (Aronson and Lang 2010) implementation of an

Method	Development Set			Test Set		
	Precision	Recall	F1	Precision	Recall	F1
Bi-LSTM-CRF	91.19	89.17	90.17			
Bi-LSTM-CRF w/convolution	93.11	91.11	92.10			
Stacked Bi-LSTM-CRF	92.65	91.99	92.32	90.73	91.96	91.34
Bi-LSTM-CRF w/EMA	91.85	89.48	90.65			
Bi-LSTM-CRF w/convolution & EMA	92.86	91.34	92.09			
Stacked Bi-LSTM-CRF w/EMA	93.68	91.70	92.68	91.10	92.30	91.70
BERT	90.98	91.91	91.44			
RoBERTa	92.31	92.34	92.32	89.93	92.05	90.98
DistilBERT	90.72	91.03	90.87			
BioBERT	92.49	91.94	92.22	89.30	92.01	90.64

Table 2: Results obtained on the development and test sets for the acronym identification task (in percentages)

Model	Time	GPU Memory	GPU
Stacked Bi-LSTM-CRF	32m34s	4 Gb	NVIDIA GTX 1050Ti
Stacked Bi-LSTM-CRF w/EMA	41m25s	4 Gb	NVIDIA GTX 1050Ti
RoBERTa	17m55s	24Gb	NVIDIA K80
BioBERT	16m50s	24Gb	NVIDIA K80

Table 3: Time and resources needed to train our best-performing models.

author-defined abbreviation detection algorithm that only detects acronyms where the author definitions occur in the same document. The SDU@AAAI-21 Task 1 corpus contains acronyms both with and without definitions; mostly without. To deal with the acronyms without local definitions we applied two deep learning approaches, Bi-directional LSTM with CRF and Transformer models.

Three variations of the Bidirectional LSTM-CRF approach (Genthial 2020) were applied to the Acronym Identification corpus: Bi-directional LSTM with CRF (Huang, Xu, and Yu 2015), Stacked Bi-directional LSTM and CRF (Lample et al. 2016), and Bi-directional LSTM and CRF with convolution and max-pooling (Ma and Hovy 2016). All three variations used GloVe embeddings. We also used the Exponential Moving Average (EMA) with all three Bi-LSTMs. The Exponential Moving Average of the weights was used to determine weights for next iteration during training. This approach improves the effectiveness of the above methods by a small margin (NIST/SEMATECH 2020).

The hyperparameters used for all three models with and without EMA are shown in Table 1.

We submitted runs for Stacked Bi-directional LSTM and CRF and Stacked Bi-directional LSTM and CRF with Exponential Moving Average, the two highest performing runs on the development set.

We also applied to the task the Simple Transformers NER implementation (version 0.49.5) (Rajapaksee 2020). We used four transformer models fine-tuned on the SDU Task 1 training set: BERT (Devlin et al. 2018), BioBERT (Lee et al. 2020), DistilBERT (Sanh et al. 2019), and RoBERTa (Liu et al. 2019). Only the batch size was modified for the task, the other parameters are the defaults for the Simple Transformers NER implementation. For BioBERT, we used the batch size 16 suggested in the Simple Transformers documentation. For RoBERTa, we used the default batch size 8. We submitted runs for RoBERTa and BioBERT transformer models.

Results

As expected, the rule based approach had high precision and low recall on the development set, we therefore did not submit the results of that approach on the test set. MetaMap achieved 96.24% precision, 17.24% recall (F1=29.24%) on the training set and 96.55% precision, 59.36% recall (F1=73.52%) on the development set. Interestingly, the recall on the development set is much higher for this rule-based approach, but not high enough to make it competitive.

We submitted test results for the Bi-LSTM and BERT-based approaches that performed best on the development set. All results are shown in Table 2.

The models using Simple Transformers are quite a bit more resource intensive than the Bi-LSTM models. The Bi-LSTM models were trained on a computer with a 4GB GTX-1050 Ti graphics card. Fine-tuning the Transformer models, however, required much more memory and it was necessary to train them on an Nvidia Tesla K80 with 24GB memory. The Bi-LSTM results are comparable to the Transformer results using less resources with a similar training time (see Table 3.)

Discussion

Our results clearly demonstrate that although the algorithmic approaches currently implemented in our biomedical named entity recognition tools have higher precision than the explored transformer-based approaches, they clearly miss many important terms. We are looking forward to learning more about the approaches explored by the other

participants of the shared task. We believe that implementing the best approaches in our tools will significantly improve recognition of named entities in clinical notes. We hope to improve recall, while maintaining the precision of our algorithmic approach, which is not far behind the human performance reported by the task organizers.

Acknowledgments

References

Aronson, A. R. 1996. MetaMap Technical Notes. Technical report, NLM. URL https://ii.nlm.nih.gov/Publications/Papers/metamap.tech.pdf.

Aronson, A. R.; and Lang, F.-M. 2010. An overview of MetaMap: historical perspective and recent advances. *Journal of the American Medical Informatics Association* 17(3): 229–236.

Demner-Fushman, D.; Rogers, W. J.; and Aronson, A. R. 2017. MetaMap Lite: an evaluation of a new Java implementation of MetaMap. *Journal of the American Medical Informatics Association* 24(4): 841–844.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* .

Genthial, G. 2020. *Named Entity Recognition with Tensorflow*. URL https://github.com/guillaumegenthial/tf_ner.

Huang, Z.; Xu, W.; and Yu, K. 2015. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991* .

Joopudi, V.; Dandala, B.; and Devarakonda, M. 2018. A convolutional route to abbreviation disambiguation in clinical text. *Journal of biomedical informatics* 86: 71–78.

Lample, G.; Ballesteros, M.; Subramanian, S.; Kawakami, K.; and Dyer, C. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360* .

Lee, J.; Yoon, W.; Kim, S.; Kim, D.; Kim, S.; So, C. H.; and Kang, J. 2020. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* 36(4): 1234–1240.

Li, I.; Yasunaga, M.; Nuzumlalı, M. Y.; Caraballo, C.; Mahajan, S.; Krumholz, H.; and Radev, D. 2019. A Neural Topic-Attention Model for Medical Term Abbreviation Disambiguation. *arXiv preprint arXiv:1910.14076* .

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

Ma, X.; and Hovy, E. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354* .

NIST/SEMATECH. 2020. *e-Handbook of Statistical Methods: Single Exponential Smoothing, National Institute of Standards and Technology,*. URL https://www.itl.nist.gov/div898/handbook/pmc/section4/pmc431.htm.

Pouran Ben Veyseh, A.; Dernoncourt, F.; Nguyen, T. H.; Chang, W.; and Celi, L. A. 2020a. Acronym Identification and Disambiguation shared tasksfor Scientific Document Understanding. In *Proceedings of the AAAI-21 Workshop on Scientific Document Understanding*.

Pouran Ben Veyseh, A.; Dernoncourt, F.; Tran, Q. H.; and Nguyen, T. H. 2020b. What Does This Acronym Mean? Introducing a New Dataset for Acronym Identification and Disambiguation. In *Proceedings of the 28th International Conference on Computational Linguistics*, 3285–3301. Barcelona, Spain (Online): International Committee on Computational Linguistics. doi:10.18653/v1/2020.coling-main.292. URL https://www.aclweb.org/anthology/2020.coling-main.292.

Rajapaksee, T. 2020. *Simple Transformers*. URL https://simpletransformers.ai/.

Sanh, V.; Debut, L.; Chaumond, J.; and Wolf, T. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108* .

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.

Wu, Y.; Denny, J. C.; Trent Rosenbloom, S.; Miller, R. A.; Giuse, D. A.; Wang, L.; Blanquicett, C.; Soysal, E.; Xu, J.; and Xu, H. 2017. A long journey to short abbreviations: developing an open-source framework for clinical abbreviation recognition and disambiguation (CARD). *Journal of the American Medical Informatics Association* 24(e1): e79–e86.

Zhou, W.; Torvik, V. I.; and Smalheiser, N. R. 2006. ADAM: another database of abbreviations in MEDLINE. *Bioinformatics* 22(22): 2813–2818.