Stud Health Technol Inform (Proc Medinfo) 2021:(submitted)
draft material -- please do not cite

Adding an Attention Layer Improves the Performance of a Neural Network Architecture for Synonymy Prediction in the UMLS Metathesaurus

Vinh Nguyen and Olivier Bodenreider

National Library of Medicine, National Institutes of Health, USA

Abstract

Background: Terminology integration at the scale of the UMLS Metathesaurus (i.e., over 200 source vocabularies) remains challenging despite recent advances in ontology alignment techniques based on neural networks. **Objectives**: To improve the performance of the neural network architecture we developed for predicting synonymy between terms in the UMLS Metathesaurus, specifically through the addition of an attention layer. **Methods**: We modify our original Siamese neural network architecture with Long-Short Term Memory (LSTM) and create two variants by (1) adding an attention layer on top of the existing LSTM, and (2) replacing the existing LSTM layer by an attention layer. **Results**: Adding an attention layer to the LSTM layer resulted in increasing precision to 92.38% (+3.63%) and F1 score to 91,74% (+1.13%), with limited impact on recall at 91.12% (-1.42%). **Conclusions**: Although limited, this increase in precision substantially reduces the false positive rate and minimizes the need for manual curation.

Keywords:

Unified Medical Language System; Neural Networks, Computer.

Introduction

The first version of the Unified Medical language System (UMLS) Metathesaurus was released 30 years ago [3]. Over the past 30 years, the size and complexity of the UMLS has grown tremendously, from integrating seven source vocabularies (grouping 208,000 terms into 64,000 Metathesaurus concepts) to a very large graph (13.7M terms from 218 sources grouped into 4.4 million concepts). Over the past three decades, this large-scale terminology integration resource has become ubiquitous in biomedical research projects and applications, where it supports not only crosswalks among standard terminologies, but also other tasks, such as natural language processing.

In contrast, the UMLS Metathesaurus development process has essentially remained unchanged over these three decades. Central to the Metathesaurus is the grouping of synonymous terms into a concept. Terms from source vocabularies are normalized [6] and lexically-similar terms become candidates for integration into the same concept. Lexically-suggested grouping of terms are then be reviewed by human Metathesaurus editors. Additional curation support includes source synonymy (i.e., synonymy asserted between terms in a source vocabulary tends to be conserved in the Metathesaurus) and source semantics (terms that do not share a common semantics are prevented from being grouped into the same concept even if they are lexically similar). Despite this algorithmic support, the curation of the Metathesaurus remains challenging and extremely labor-intensive.

We recently developed a synonymy prediction model for the UMLS Metathesaurus based on neural networks and showed that it largely outperformed the algorithms currently used for supporting Metathesaurus curation [7]. More specifically, we achieved the following performance: precision = 0.8875, recall = 0.9254 and F1 score = 0.9061. At the scale of the Metathesaurus, the number of false positive synonymy predictions for this system remains very high and improving the performance of our model remains a priority.

In recent years, the use of attention mechanisms has improved the performance of neural network models in a variety of tasks, from natural language processing to computer vision [10].

The objective of this work is to improve the performance of the neural network architecture we developed for predicting synonymy between terms in the UMLS Metathesaurus. More specifically, we explore whether the addition of an attention layer to our original Siamese neural network architecture with BioWordVec embeddings and Long-Short Term Memory (LSTM) yields additional performance. More specifically, we assess which of the two following variants performs better: (1) adding an attention layer on top of the existing LSTM, or (2) replacing the existing LSTM layer by an attention layer.

The specific contributions of this work include (1) a simple modification to our existing neural network architecture with an additional attention layer on top of the LSTM layer that yields +3.63% in precision and +1.13% in F1 score, and (2) confirmation that attention improves performance for predicting synonymy between UMLS Metathesaurus terms.

Background

Related work

Previous work on synonymy prediction in the UMLS Metathesaurus

We recently formalized synonymy prediction in the Metathesaurus as a vocabulary alignment problem (UVA). We developed a neural network model for predicting synonymy between terms in the UMLS Metathesaurus [3]. This simple model solely leverages the lexical features of terms. Our neural network architecture consists of BioWordVec embeddings fed to Long Short-Term Memory (LSTM) neural network. Because our goal is to compare two terms, we adopted a Siamese architecture, in which the two terms are processed in parallel and the output vectors compared using a Manhattan distance metric. Furthermore, we created different datasets with different degrees of lexical similarity among negative examples for training the neural networks and testing their generalization. Our experiments showed that the model trained with negative examples from different degrees of lexical similarity yielded the best performance for the UVA task. (See Datasets section below for details). The performance of this model was: accuracy = 0.9938, precision = 0.8875, recall = 0.9254 and F1 score = 0.9061.

Attention mechanisms in neural network models

Attention mechanisms were first used in neuroscience [8] and have gained popularity in other fields, especially in natural language processing. Self-attention mechanisms relating different word positions of an input to compute its context representation have succeeded in a variety of tasks including reading comprehension, abstractive summarization, textual entailment and learning task-independent sentence representations [1; 2; 4; 5; 9; 10]. BERT [2] is a great example among many successful projects that use a self-attention mechanism with multi-heads in both pretraining and fine-tuning tasks. Although BioBERT [4], further pretraining of BERT on PubMed abstracts, has shown performance improvements on several biomedical NLP tasks, the pretraining cost is substantial – it required 16x Tesla V100 GPUs running continuously for 23 days. Our preliminary work suggested that fine-tuning BERT or BioBERT for our UVA task will also be computational expensive in both the training and testing phases. Therefore, we are interested in evaluating a simpler attention mechanism that can be scalable for the UVA task. Compared to BERT and BERT-variants using self-attention mechanism with multi-heads, the dot-product attention mechanism is much faster and more space-efficient [1; 10]. As a result, we chose to implement and evaluate the dot-product attention mechanism for the UVA task in this paper.

Datasets

In our original work, we created multiple datasets with increasing levels of lexical similarity among negative pairs, because we hypothesized that it would be difficult to predict the absence of synonymy between lexically-similar terms. We showed that the best performing model was trained on the large dataset including the three variants in terms of lexical similarity ("ALL"). While the objective is to improve the performance with the proposed attention-based models, we also want to assess the generalization of our attention-based models on the three degrees of lexical similarity among negative examples (TOPN_SIM – high-level of lexical similarity, RAN_SIM – low level of lexical similarity and RAN_NOSIM – no lexical similarity). For training, we used the ALL dataset with 170,075,628 negative examples, and 22,324,834 positive examples. For testing the generalization of the models, we used the ALL dataset and the dataset variants, for which the numbers of positive and negative examples are shown in Table 1.

Table 1 – Training and generalization test datasets (number of pairs of terms)

Training	**Negative**	**Positive**	**Total**
ALL (training)	147,750,794	22,324,834	170,075,628
Testing	**Negative**	**Positive**	**Total**
TOPN_SIM	54,752,228	5,581,209	60,333,437
RAN_SIM	54,445,899	5,581,209	60,027,108
RAN_NOSIM	58,256,526	5,581,209	63,837,735
ALL (testing)	167,454,653	5,581,209	173,035,862

Methods

Architecture

We implemented a simple attention layer [1] where the context vector is created by (1) taking the dot product of inputs and weights followed by the addition of bias, (2) applying a *tanh* function followed by a softmax layer, and (3) taking the dot product of softmax outputs and the hidden states. This attention layer is used to create two model architecture variants, V1 and V2, depicted in Figure 1, along with the original model, V0.

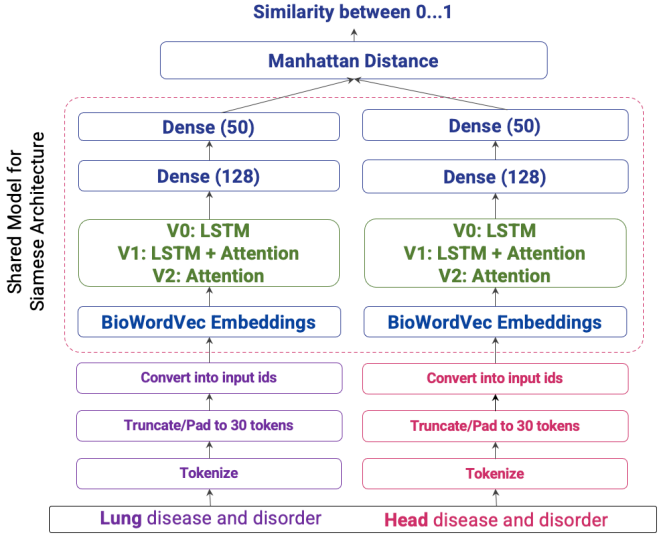

Figure 1: The proposed neural network architectures with three variants: V0 as the original architecture with LSTM alone, V1 with an attention layer on top of the LSTM layer, and V2 with the LSTM layer replaced by an attention layer.

Using LSTM alone (V0)

This is the original model used in [7], which we use as a baseline in this investigation. In this model, the original LSTM layer that only outputs the last hidden state.

Using attention in addition to LSTM (V1)

In this architecture, we add an attention layer with the same number of hidden states as in the original LSTM layer in V0. Unlike the V0 variant, however, the LSTM layer in this variant outputs all the hidden states. The output from this LSTM layer is fed to the attention layer described above.

Using attention in replacement of LSTM (V2)

In this model variant, we replace the LSTM layer by the attention layer with the same number of hidden states. The context vector output is passed to the remaining layers of the architecture.

Experimental Setup

We conduct the following experiments for each model variant: (1) we train the models (V1 and V2) using the ALL dataset, (2) we test these trained models using the ALL generalization test dataset, and (3) we test these trained models using the TOPN_SIM, RAN_SIM, and RAN_NOSIM generalization tests.

All the experiments are deployed to the Biowulf High-Performance Cluster at the National Institute of Health. We use Tesla V100x GPU with 32 GB of GPU RAM and 220 GB of system RAM for each experiment.

Training parameters

We use 50 hidden states for both LSTM and attention layers. The remaining hyperparameters are the same as in our original model. (For details, see [7].) We train the models with 100 epochs with a batch size 8192 and report the results in Table 2. Each epoch takes 27 minutes for training.

Quantitative evaluation

We compute the usual metrics (accuracy, precision, recall and F1 score) for the two models we developed (i.e., using attention in addition to LSTM [V1] and using attention in replacement of LSTM [V2]) and compare these results to those obtained with our initial model that does not leverage attention (V0). Additionally, we assess the effect of the best performing model on the three generalization test datasets with various degrees of lexical similarity.

Qualitative evaluation

In addition to comparing the performance of the models, we also assess their impact on the false positive rate (FPR). The false positive rate is important here given the predominance of negative cases in our main test dataset (ALL).

Results

Quantitative evaluation

The performance metrics for the two models (V1 - LSTM + Attention, and V2 - Attention alone) using the ALL generalization test dataset are shown in Table 2, along with the metrics for the baseline from prior work (V0 - LSTM alone).

Compared to the original architecture with LSTM alone (V0), the architecture with an additional attention layer on top of the LSTM layer (V1) yields better performance. It improves accuracy (+0.09%), precision (+3.63%), and F1 (+1.13%) while slightly reducing recall (-1.42%).

In contrast, the architecture with the LSTM layer replaced by the attention layer (V2) performs poorly on all the metrics.

Table 2 –Performance of the three models for testing using the ALL generalization test dataset

Variant	Accuracy	Precision	Recall	F1
V0	0.9938	0.8875	0.9254	0.9061
V1	**0.9947**	**0.9238**	**0.9112**	**0.9174**
V2	0.9928	0.8876	0.8908	0.8892

Performance improvement is more markedly observed on the test dataset with the highest level of lexical similarity (TOP_SIM).

Qualitative evaluation

The false positive rate of the baseline model (V0) is 0.39% (654,699 / 167,454,653) vs. 0.25% (419,487 / 167,454,653) with the best performing model (V1).

Discussion

Findings

In this experiment, we showed that adding an attention layer to the original LSTM neural network was beneficial in terms of precision (+3.63%) and overall performance (+1.13%), with minimal cost in terms of recall (-1.42%). This is interesting and encouraging, because this gain in performance can be attributed to the attention layer. This means that adding features to the model (e.g., adding contextual information for disambiguating homonyms) will likely increase performance beyond the gains attributable to the attention layer.

However, our experiments also show that using an attention mechanism instead of the LSTM neural network resulted in poor performance. Therefore, while the attention mechanism is an important component of a neural network architecture, where the attention mechanism is used matters a great deal. In our model, the addition of an attention layer was on top of the LSTM layer was most beneficial.

Significance

Cost-effectiveness of the solution. While the addition of an attention layer only yielded modest gains in performance, these gains came at a very limited cost. Adding an attention layer to our original neural network architecture required minimal changes to the architecture. Moreover, adding an attention layer did not significantly increase processing time for training and testing. Finally, we were able to reuse the same datasets we created for the original work, both for training and testing.

Practical significance of decreasing the false positive rate. The UMLS Metathesaurus integrates some 10M English terms, which are amenable to synonymy prediction with our models. At this very large scale, a 1% gain in precision and the corresponding drop in false positive rate largely reduces the burden of manual curation. For example, in our test dataset with 173M pairs of terms with 5,581,209 positive pairs, increasing precision from 0.8875 to 0.9238 reduces the number of false positives from 654,699 to 419,487 – a 36% reduction. Even if each false positive only required ten seconds for a human Metathesaurus editor to adjudicate, this would represent a saving of 653 hours (at the limited scale of our test sample).

Limitations and future work

One limitation of this work is the small impact on recall (-1.42%) observed with the model after the addition of the attention layer. In the future we hope to compensate for this by adding features susceptible to increase recall, e.g., source synonymy. Another limitation is that this model remains a purely lexical model (i.e., only the terms themselves are fed to the model). Our future plans include adding contextual information to the model (e.g., hierarchical relations) to support the disambiguation of homonyms.

Conclusions

Adding an attention layer to our initial neural network architecture is a simple way of getting a moderate performance improvement, particularly for precision. Although limited, this increase in precision will reduce the false positive rate and minimize the need for manual curation.

Acknowledgements

This research was supported by the Intramural Research Program of the NIH, National Library of Medicine.

References

[1] D. Bahdanau, K. Cho, and Y. Bengio, Neural machine translation by jointly learning to align and translate, *arXiv preprint arXiv:1409.0473* (2014).

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

[3] B.L. Humphreys, G. Del Fiol, and H. Xu, The UMLS knowledge sources at 30: indispensable to current research and applications in biomedical informatics, *J Am Med Inform Assoc* **27** (2020), 1499-1501.

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

[5] Z. Lin, M. Feng, C.N.d. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, A structured self-attentive sentence embedding, *arXiv preprint arXiv:1703.03130* (2017).

[6] A.T. McCray, S. Srinivasan, and A.C. Browne, Lexical methods for managing variation in biomedical terminologies, *Proc Annu Symp Comput Appl Med Care* (1994), 235-239.

[7] V. Nguyen, H.Y. Yip, and O. Bodenreider, eds., *Biomedical Vocabulary Alignment at Scale in the UMLS Metathesaurus*, ACM, New York, NY, USA.

[8] B.A. Olshausen, C.H. Anderson, and D.C. Van Essen, A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information, *Journal of Neuroscience* **13** (1993), 4700--4719.

[9] A.P. Parikh, O. Täckström, D. Das, and J. Uszkoreit, A decomposable attention model for natural language inference, *arXiv preprint arXiv:1606.01933* (2016).

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, and I. Polosukhin, Attention is all you need, *arXiv preprint arXiv:1706.03762* (2017).

Address for correspondence

olivier.bodenreider@nih.gov