Machine Learning with Word Embeddings applied to Biomedical Concept Disambiguation

Rui Antunes
ruiantunes@ua.pt

Sérgio Matos
aleixomatos@ua.pt

IEETA
University of Aveiro

Aveiro, Portugal
http://bioinformatics.ua.pt

Abstract

Artificial Intelligence (AI) has grown in the last years and it has many applications. Natural Language Processing is one of the AI tasks, which has the objective to endow the machines the capability of understanding human language. This is an important process due to the amount of information stored in textual form. There is a growing need for automatic extraction of knowledge, and NLP comes in this direction helping in tasks such as information extraction and information retrieval. Word sense disambiguation is an important NLP subtask, which is responsible for assigning the proper concept to an ambiguous word or term.

In this paper, we present results obtained from applying supervised machine learning algorithms with local features, and word embeddings as global features extracted from Wikipedia and PubMed knowledge sources. These results indicate that word embedding features are informative and may improve the biomedical word disambiguation accuracy.

1 Introduction

Large volumes of biomedical data are produced every day, and this is accompanied by an also increasing amount of textual data, mostly in the form of scientific publications. In order to efficiently treat and interpret these data it is necessary to create tools that automatically do this job, reducing the human efforts. This led to the application of text mining methods for extracting information from the literature and linking that to repositories of biomedical data. For instance, the work in [1] describes a framework for biomedical concept recognition, which is a relevant task for biomedical Information Extraction (IE).

Word Sense Disambiguation (WSD), an important subtask of Natural Language Processing (NLP) [2], is a challenging task that consists of finding the correct sense of an ambiguous term. Usually, this is achieved using the surrounding context of the term. Currently, there are mainly two distinct approaches for WSD, those based on Machine Learning (ML) algorithms and the ones based on knowledge sources. The ML approaches can follow supervised, semi-supervised or unsupervised algorithms, with supervised classification approaches currently offering the best results in terms of accuracy, achieving around 94% using a Support Vector Machine (SVM) classifier [3].

Knowledge-based approaches to WSD have also attracted large interest, as these approaches are usually less dependent on training data, which may lead to better generalization when compared to supervised learning algorithms. The use of multiple knowledge bases brings benefits to the problem of concept disambiguation [4]. WordNet [5] is a large knowledge database of the English language that has been extensively applied for word sense disambiguation [2]. In the case of biomedical texts, the largest and most relevant knowledge database is the Unified Medical Language System (UMLS) [6], which offers a rich integrated metathesaurus and semantic network for this domain.

Word embeddings is a recent technique, which can be applied in NLP. It converts words from a document collection, or corpus, into vectors of real numbers. These word embeddings can be used as global features in a ML classification problem. In our case, these features were used in the disambiguation task, which showed to be almost as effective as local features. In [3], the authors present a work on supervised biomedical word sense disambiguation applied to the MSH WSD data set [7], exploring the combination of unigrams as local features and word embeddings as global features. Other approaches using word embeddings for word sense disambiguation have also been proposed by Wu et al. [8], and Taghipour and Ng [9].

In this work, we applied several machine learning methods in the MSH WSD data set in order to measure the WSD accuracies. The ML classifiers used in this experiment were the decision tree classifier, the k-nearest neighbors vote, the passive aggressive linear model, the ridge regression classifier, the Support Vector Machine (SVM) classifier. Textual data from Wikipedia and PubMed corpus were used to generate the word embeddings features to be used in the classifiers.

2 Data Set

The MSH WSD data set was automatically generated using the UMLS Metathesaurus and MEDLINE citations [7]. The data consisting of scientific abstracts, each with one ambiguous term identified and mapped to the correct sense. It contains 203 ambiguous terms with a total of 423 distinct senses. Most terms (189) have only two different meanings, 12 terms have three different meanings, and the remaining 2 terms have four and five different meanings. There are a total of 37,888 examples of ambiguity. Each term has, on average, 187 citations, that is, ambiguity cases.

3 Methods

For each ambiguous term, we applied 5-fold cross-validation to subdivide the corresponding abstracts for training and testing the model. A bag-of-words (BOW) model was used to represent the texts, with local features acquired from the context, namely unigrams and bigrams, with tf-idf weighting. In order to evaluate the impact on WSD accuracy, we also added word-embedding vectors, calculated from Wikipedia and PubMed corpora, as global features. A list of 313 stopwords obtained from the Medline repository[1] was used to filter out very frequent words in the corpus. All these tasks were implemented using the framework Scikit-learn [10], a machine-learning library for the Python programming language. Word embedding models were obtained with the gensim framework [11].

3.1 Machine Learning Methods

In order to obtain the highest accuracy, several machine learning classifiers were compared: decision tree, k-nearest neighbor, passive aggressive linear model, ridge regression, SVM.

3.2 Feature Combination

The local features used were unigrams and bigrams, and the global features used were the word embeddings. We tested different feature combinations in order to understand which combination produced the best results. Local features were scaled using the term frequency – inverse document frequency (tf-idf) scheme.

Table 1: WSD accuracies with only global features (Wikipedia model vs PubMed model). Results shown are the average across five folds. DT: Decision Tree; kNN: k-Nearest Neighbor (k=5); PA: Passive Aggressive linear model; RR: Ridge Regression; SGD: linear Support Vector Machine with Stochastic Gradient Descent; SVC: Support Vector Classification.

	Model of word embeddings from	
	Wikipedia	PubMed
DT	0.817	0.849
kNN	0.896	0.918
PA	0.893	0.928
RR	0.905	0.910
SGD	0.874	0.916
SVC	0.912	0.924

[1] https://mbr.nlm.nih.gov/Download/2009/WordCounts/wrd_stop

Table 2: Accuracies using distinct features combinations. Results shown are the average across five folds. U: Unigrams; B: Bigrams; WE: Word Embeddings with PubMed model; DT: Decision Tree; kNN: k-Nearest Neighbor (k=5), PA: Passive Aggressive linear model; RR: Ridge Regression classifier; SGD: linear Support Vector Machine with Stochastic Gradient Descent; SVC: Support Vector Classification.

	Local features			Local and global features
	U	B	U+B	U+B+WE
DT	0.903	0.862	0.901	0.908
kNN	0.913	0.918	0.924	0.919
PA	0.950	0.938	0.949	0.934
RR	0.942	0.922	0.940	0.939
SGD	0.947	0.931	0.946	0.919
SVC	0.948	0.932	0.948	0.938

3.3 Word Embeddings

Two distinct models of word embeddings were calculated, from Wikipedia and PubMed articles respectively. Wikipedia is range-wide, having no specific domain. The full Wikipedia dump, obtained in September 2015, was used, amounting to approximately four million articles and containing about two million distinct words. PubMed, on the other hand, is specific to biomedical domain. Around six million abstracts corresponding to the years 2010 to 2015 were used, containing around 400 thousand distinct words. Both models were trained with a window of five words and for a feature vector of size 100. Each abstract (instance) was represented by the weighted average of the embedding vectors for all the words in the abstract, with the tf-idf value of each word used as weight.

4 Results

First, we compare the two distinct models of word embeddings as unique features of the classification problem in order to find the best model to use (Table 1). As expected, the model from the domain specific PubMed corpus outperformed the more general model created from Wikipedia articles. Nevertheless, the results obtained with the latter indicate that even features extracted from general corpora may contribute to these methods.

The results in Table 2 show that the state-of-the-art results for this problem can be reproduced using simple word-based features. It is also noticeable that bigram features contribute only slightly to the results, and unigram features alone achieve almost as good if not better results than the combination of unigram and bigrams. Also, comparing these results with Table 1, one can observe that word embedding features alone, which in this study represent vectors of 100 features, allow obtaining results that are very close to the best results obtained with unigram features.

On the other hand, the combination of word embedding features with unigrams and bigrams did not improve results. In our experiments the highest accuracy, 95.0%, was obtained with unigram features alone, using the passive-aggressive linear classifier.

5 Discussion

As expected, the word embeddings model from PubMed outperformed the Wikipedia model, since PubMed is specific to the biomedical domain. In our experiments, the best accuracy was attained with simple unigram features, and adding bigram features only improved the results in the case of the kNN classifier.

Disambiguation using word-embedding features alone proved very positive for this data set, even when using a small portion of the full MEDLINE database, which contains around 22 million abstracts. However, the combination of these and simple word-based features lowered the classification accuracy. Jimeno-Yepes [3] achieved an accuracy of 95.97%, in this same data set, with the combination of unigrams and word embeddings from the full MEDLINE. In future work, we will extend this analysis and investigate different strategies for integrating the word embedding features in this classification problem.

References

[1] D. Campos, S. Matos and J. L. Oliveira. A modular framework for biomedical concept recognition. *BMC Bioinformatics*, 14:281, 2013.

[2] R. Navigli. Word Sense Disambiguation: A Survey. *ACM Computing Surveys*, 41(2):10, 2009.

[3] A. J. Jimeno-Yepes. Higher order features and recurrent neural networks based on Long-Short Term Memory nodes in supervised biomedical word sense disambiguation. *arXiv: 1604.02506v1 [cs.CL]*, 2016.

[4] C. T. Tsai and D. Roth. Concept Grounding to Multiple Knowledge Bases via Indirect Supervision. *Transactions of the Association for Computational Linguistics*, 4:141–154, 2016.

[5] C. Fellbaum. WordNet: An electronic lexical database. *Cambridge: MIT Press*, 1998.

[6] O. Bodenreider. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, 32:267–270, 2004.

[7] A. J. Jimeno-Yepes, B. T. McInnes and A. R. Aronson. Exploiting MeSH indexing in MEDLINE to generate a data set for word sense disambiguation. *BMC Bioinformatics*, 12:223, 2011.

[8] Y. Wu, J. Xu, Y. Zhang and H. Xu. Clinical Abbreviation Disambiguation Using Neural Word Embeddings. *ACL-IJCNLP*, pages 171–176, 2015.

[9] K. Taghipour and H. T. Ng. Semi-Supervised Word Sense Disambiguation Using Word Embeddings in General and Specific Domains. *Proceedings of NAACL HLT*, pages 314–323, 2015.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[11] R. Rehurek and P. Sojka. Software Framework for Topic Modelling with Large Corpora. *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, 2010.