.

TüKaPo at SemEval-2020 Task 6: Def(n)tly not BERT: Definition Extraction using pre-BERT Methods in a post-BERT World

Madeeswaran Kannan
Department of Linguistics
University of Tübingen, Germany
mkannan@sfs.uni-tuebingen.de

Haemanth Shanthi Ponnusamy
Department of Linguistics
University of Tübingen, Germany
???@sfs.uni-tuebingen.de

Abstract

We describe our system (TüKaPo) submitted for Task 6: DeftEval, at SemEval 2020. We developed and evaluated multiple neural network models based on CNNs and LSTMs to perform binary classification of sentences containing definitions. Our final model achieved a F1 score of 0.6851 in subtask 1.

1 Introduction

Definition detection and extraction has been a well-researched topic in NLP research for over a decade.

2 Background

The DeftEval shared task is based around the English-language DEFT (Definition Extraction From Texts) corpus (Spala et al., 2019). It consists of annotated text extracted from the following semi-structured and free-text sources: 2017 SEC contract filings from the US Securities and Exchange Commission EDGAR database[1], and open-source textbooks from OpenStax CNX[2]. The latter encompasses topics from areas of biology, history, physics, psychology, economics, sociology, and government. Compared to similar existing definition corpora such as WCL (Navigli and Velardi, 2010) and W00 (Jin et al., 2013), the data offered by the DEFT corpus is larger in size (23,746 sentences; 11,004 positive annotations) while also providing finer-grained feature annotations (c.f figure 1[3]).

The shared tasks consists of three subtasks: 1) Sentence Classification (classify if a sentence contains a definition or not), 2) Sequence Labeling (label each token with BIO tags according to the corpus specification), and 3) Relation Classification (label the relations between each tag according to the corpus specification). We participated in the first subtask.

Training and development data is common for all three subtasks. It is presented in a tab-delimited CONLL-2003-like (Sang and De Meulder, 2003) format where each line represents a token and its features:

$$[TOKEN]\ \ [SOURCE]\ \ [START\_CHAR]\ \ [END\_CHAR]\ \ [TAG]\ \ [TAG\_ID]\ \ [ROOT\_ID]\ \ [RELATION]$$

SOURCE is the source text file, START_CHAR and END_CHAR are the character index boundaries of the token, TAG is the BIO label of the token, TAG_ID is the ID associated with the TAG, ROOT_ID is the ID associated with the root of this relation (if any), and RELATION is the relation tag of the token.

The test data for the first subtask is presented in the following CONLL-2003-like format: $[SENTENCE]$ $[BIN\_TAG]$. BIN_TAG is 1 if SENTENCE contains a definition, 0 otherwise. During the

[1] https://www.sec.gov/

[2] https://cnx.org/

[3] Additionally, relationships between terms and definitions are also annotated.

Tag	Description
Term	Primary term
Alias Term	Secondary, less common name for the primary term
Ordered Term	Multiple inseparable terms that have matching sets of definitions
Referential Term	NP reference to a previously mentioned term
Definition	Primary definition of a term
Secondary Definition	Supplemental information crossing a sentence boundary that could be part of the definition
Ordered Definition	Multiple inseparable definitions that have matching sets of terms
Referential Definition	NP reference to a previously mentioned definition
Qualifier	Specific date, location, or condition under which the definition holds

Table 1: DEFT Tag Schema

training for the first subtask, the training and development datasets were converted into the same format was the test dataset using a script provided with the corpus. A positive label was associated with every sentence that contained tokens with `B-Definition` or `I-Definition` tags; all other sentences were associated with a negative label.

2.1 Related Work

In the early days of definition extraction, rule-based approaches leveraging linguistic features showed promise. Westerhout (2009) used a combination of linguistic information (n-grams, syntactic features) and structural information (position in sentence, layout) to extract definitions from Dutch texts. Such approaches, however, were found to be dependent on language and domain and scale poorly. Later research incorporated machine learning methods to encode lexical and syntactic features as word vectors (Del Gaudio et al., 2014). Noraset et al. (2017) tackled the problem as a language modelling task over learned definition embeddings. Espinosa-Anke et al. (2015) derive feature vectors from entity-linking sources and sense-disambiguated word embeddings. More recently, Anke and Schockaert (2018) use convolutional and recurrent neural networks over syntactic dependencies to achieve state-of-the-art results on the WCL and W00 datasets (Navigli and Velardi, 2010; Jin et al., 2013).

3 System Overview

We developed and iterated over multiple LSTM-based recurrent neural network models (Hochreiter and Schmidhuber, 1997). As our baseline, we designated a model with a single bidirectional LSTM layer followed by two feed-forward layers and a final sigmoid read-out layer. Token sequences were extracted from the training data's source sentences and passed as inputs to an embedding lookup layer, whose embeddings were used as inputs to the LSTM layer.

To experiment with multiple features, we developed a non-sequential model that extracted and learned representations for each type of feature. The concatenated representation of the feature embeddings was then fed as input to a BiLSTM layer followed by a feed-forward network.

Finally, we also experimented with a hybrid-approach of using convolutional layers in addition to the recurrent layers. To this end, we augmented the previous architecture by interspersing single-dimensional convolution and max-pooling layers over the concatenated feature representation. The intuition behind this step was to leverage the implicit feature-extraction performed by the convolutional layers to refine the final representation passed to the recurrent layer.

3.1 Features

Beyond using word tokens, we experimented with different types of input primitives to bolster the informativity of the learned feature vectors. As evidenced by previous approaches, the task of definition extraction benefits from the incorporation of syntactic information. To facilitate this, we tested our models with part-of-speech tags, dependency labels, and head-modifiers as inputs (individually and combined).

We also tested the impact of punctuations and frequency constraints on all input primitives. Furthermore, to imbue the model with semantic and lexical information, pre-trained embeddings were used to initialise the embedding matrices of word tokens. For this, GloVe (Pennington et al., 2014) and word2vec (Mikolov et al., 2013) word embeddings were used.

3.2 Experiments

4 Evaluation & Results

5 Conclusion

References

Luis Espinosa Anke and Steven Schockaert. 2018. Syntactically aware neural architectures for definition extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 378–385.

Rosa Del Gaudio, Gustavo Batista, and António Branco. 2014. Coping with highly imbalanced datasets: A case study with definition extraction in a multilingual setting. *Natural Language Engineering*, 20(3):327–359.

Luis Espinosa-Anke, Horacio Saggion, and Claudio Delli Bovi. 2015. Definition extraction using sense-based embeddings. In *Gupta P, Banchs RE, Rosso P, editors. International Workshop on Embeddings and Semantics (IWES'15); 2015 Sept 15; Alicante, Spain.[Place unknown]:[CEUR]; 2015.[6 p.].* CEUR.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Yiping Jin, Min-Yen Kan, Jun Ping Ng, and Xiangnan He. 2013. Mining scientific terms and their definitions: A study of the acl anthology. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 780–790.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Roberto Navigli and Paola Velardi. 2010. Learning word-class lattices for definition and hypernym extraction. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 1318–1327. Association for Computational Linguistics.

Thanapon Noraset, Chen Liang, Larry Birnbaum, and Doug Downey. 2017. Definition modeling: Learning to define word embeddings in natural language. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

Sasha Spala, Nicholas A. Miller, Yiming Yang, Franck Dernoncourt, and Carl Dockhorn. 2019. DEFT: A corpus for definition extraction in free- and semi-structured text. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 124–131, Florence, Italy, August. Association for Computational Linguistics.

Eline Westerhout. 2009. Definition extraction using linguistic and structural features. In *Proceedings of the 1st Workshop on Definition Extraction*, pages 61–67. Association for Computational Linguistics.