.

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

Madeeswaran Kannan* **Haemanth Shanthi Ponnusamy†**
Department of Linguistics
University of Tübingen, Germany

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

By most accounts, the first reliable English dictionary was written by Samuel Johnson[1] and published on 15 April, 1755. It was 18 inches tall, 20 inches wide when opened, and contained 42,773 entries. It took Dr. Johnson 7 years to complete, and yet it was missing the word *contrafibularity*, amongst others[2]. In the 265 years that have passed since, the world has evolved at a blisteringly fast pace with technology integrating itself ever more closely with our lives. However, the compilation and maintenance of dictionaries and lexicons - one of the most important and authoritative sources of meaning - continues to be the exclusive field of domain experts and lexicographers. Nevertheless, with the recent advances in natural language processing, this area - like many others that deal with human language - has seen a growing interest in automating the development of such resources.

Definition extraction is defined as the automatic identification of definitional knowledge in text, modeled as a binary classification problem between definitional and non-definitional text. 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 scaled 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 very good results on the WCL and W00 datasets (Navigli and Velardi, 2010; Jin et al., 2013).

This paper describes our approach of combining existing methods over state-of-the-art techniques in an attempt to determine if the former still offer avenues of optimization that can help them perform competitively with the latter.

* mkannan@sfs.uni-tuebingen.de
† haemanth.santhi-ponnusamy@student.uni-tuebingen.de

[1] `https://en.wikipedia.org/wiki/A_Dictionary_of_the_English_Language`

[2] *Aardvark* was another.

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

1.1 Task 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[3], and open-source textbooks from OpenStax CNX[4]. 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 table 1).

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 training for the first subtask, the training and development datasets were converted into the same format as 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 System Overview

2.1 Baseline

We developed and iterated on both LSTM-based (Hochreiter and Schmidhuber, 1997) recurrent and convolutional (O'Shea and Nash, 2015) neural network models. Our baseline RNN architecture is a network of a single bidirectional LSTM layer followed by two feed-forward layers and a final

[3]https://www.sec.gov/
[4]https://cnx.org/

Experiment	Model	Word Embeddings	Features	Precision	Recall	F1-Score
1. W/o Semantic Information	BL-RNN	-	POS + Deps	0.56	0.75	0.64
2. W/t Semantic Information	BL-RNN	Glove	Tokens + Deps	0.74	0.63	**0.68**
		w2v		0.71	0.60	0.65
	BL-CNN	Glove		0.72	0.62	**0.67**
		w2v		0.72	0.58	0.64
3. Effect of punctuation & dependency relations	BL-RNN	Glove	Tokens + POS	0.75	0.62	0.68
			Tokens + Deps + POS	0.76	0.58	0.66
			Tokens + POS + Punct	0.76	0.64	**0.69**
			Tokens + Deps + POS + Punct	0.76	0.62	0.68
	BL-CNN		Tokens + POS	0.74	0.65	0.69
			Tokens + Deps + POS	0.77	0.67	**0.71**
			Tokens + POS + Punct	0.77	0.64	0.70
			Tokens + Deps + POS + Punct	0.79	0.62	0.70
4. Final Model	FINAL-HYBRID	Glove	Tokens + POS + Punct	0.77	0.64	0.70
			Tokens + Deps + POS + Punct	0.75	0.70	**0.73**

Table 2: Results of experiments performed on the baseline & final models

sigmoid-activated read-out layer. This architecture is implemented by model *BL-RNN* whose input layer accepts sequences of features vectors. Our baseline hybrid-CNN architecture is implemented by model *BL-CNN*, which is based on the work by Anke and Schockaert (2018). It accepts feature vector sequences that are passed through a one-dimensional convolutional filter and a max-pooling layer, followed by a single BiLSTM and read-out layers. The intuition behind combining convolutional and recurrent layers is to leverage the implicit local feature-extraction performed by the convolutional layers to refine the final representation passed to the recurrent layer, which accounts for global features. The input sequences are composed as concatenations of vectors of individual features at the token level, resulting in a homogenous representation, e.g. each token is encoded as the concatenation of a n-dimensional word vector, a m-dimensional one-hot encoded POS tag vector, etc.

We conducted several experiments with the above two architectures and iterated on successful models. The provided corpus was pre-split into *train* and *dev* splits. A 90-10 split was perfomed on the *train* split to generate the validation set; the *dev* split was used as the test data as-is. All models were trained for 100 epochs with an early-stopping mechanism that monitored the validation loss over the last 10 epochs. Batch size was set to 128, and ADAM (Kingma and Ba, 2014) was used as the binary cross-entropy optimizer. URLs were stripped from token sequences as a preprocessing step. The results of our experiments are listed in table 2. The reported figures were averaged over three iterations of each experiment.

2.2 Influence of Semantic Information

Our initial experiments were premised on the hypothesis that neural definition extraction can be primarily modelled on morphosyntactic features while excluding or restricting the use of semantic and lexical information. By limiting the influence of semantics, we expected to train a model that generalized well over multiple domains by virtue of being less susceptible to lexical cues that could potentially act as distractors. To test this, we trained the RNN model on a concatenation of part-of-speech tag and dependency relation sequences. Similarly, two more RNN models were trained on word embedding and dependency relation sequences, their word embedding matrix initialized with 300-dimensional pre-trained GloVe (Pennington et al., 2014) and word2vec (Mikolov et al., 2013) embeddings respectively[5]. The results proved our hypothesis to be flawed, as the models enriched with semantic information provided by the word embeddings consistently out-performed our syntax-only model. This result also carried over to the hybrid-CNN models. Overall, we found that models trained with GloVe embeddings to be more performant than those trained with w2v.

[5]The GloVe and w2v embeddings were trained on the Common Crawl and Google News corpora respectively.

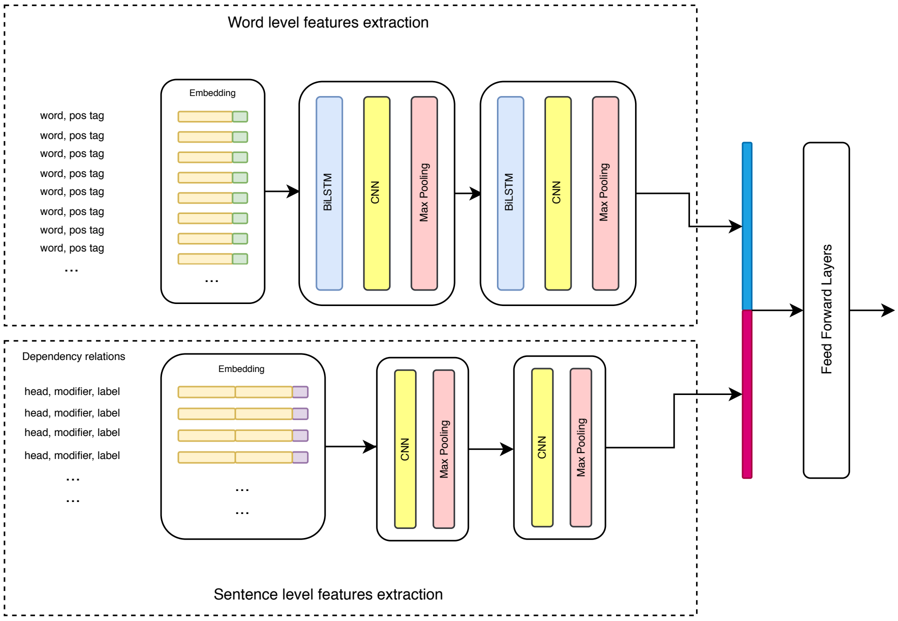

Figure 1: Final architecture

2.3 Feature Modelling

Building upon the findings of the previous experiments, we tested the effect of combining punctuation and part-of-speech tags. It was immediately evident that replacing the *PUNCT* POS tag with the punctuation character occurring at that position had a positive effect on the model's performance. Beyond the implicit increase in information offered by the actual character, it also reaffirms the importance of syntactic features in this task. The addition of dependency relation features, however, had a less immediately-obvious impact. The RNN model saw a reduction in performance while the hybrid-CNN model fared better. Upon further investigation, we determined that the input encoding scheme's attempt to homogenize feature vectors across disparate features, viz., combining sequential (token-level) features (token, POS) with non-sequential (sentence-level) features (dependencies), actually hindered the recurrent model from optimally exploiting the former. With this key insight, we were able to rearchitect our model to learn a representation that composes both token and sentence-level features in an separate but efficient way.

2.4 Final Architecture

Our final architecture is informed by the findings of our previous experiments. It accepts five inputs: At the token-level, both token and part-of-speech tags (w/t punctuation) are used. Pre-trained GloVe embeddings are used for tokens, while embeddings for POS tags are learned on-the-fly. The concatenation of both embeddings is passed through two "feature extraction units" that consist of a BiLSTM (to target sequence/global information) and a 1D-Conv + MaxPool layer (to target local information). At sentence-level, dependency information is encoded as the concatenation of the embeddings of the head word, modifier word and dependency label of each relation. This is connected to two stripped-down "feature extraction units" without the BiLSTM layer, since dependency relations are sequentially independent. Finally, the extracted representations of both token and sentence-level features are concatenated and connected to a feed-forward layer and then a read-out layer.

The above architecture's separation of feature-extraction at token and sentential levels allows their information to be combined at a higher level in the network. And we indeed see a marked improvement when this model is trained with dependency information. The model achieved a best F1-score of 0.76 during development.

Hyperparameter		Value	
		Token-Level	Sentence-Level
Embeddings Dim	Word	300	
	POS	32	
	Dep. Label	32	
Feature Extractor Units (Unit 1, Unit 2)	LSTM Units	128, 64	-
	Conv. Filters	128, 64	64, 32
	Conv. Kernel Size	3, 3	
	MaxPooling Pool Size	2, 2	
L2 Regularization β		0.001	
Feed-forward Units		24	

Table 3: Hyperparameters for the final model

3 Results & Discussion

The final model achieved a positive-class F1-score of 0.6851, ranking 47th out of 56 submissions for the first subtask. While the model under-performed in a substantial departure from our expectations, we identified multiple factors that may have contributed to it[6]. During training, we noticed that the number of unique terms in the preprocessed corpus out-stripped the number of training samples (over 21K unique tokens in approx. 17K sentences), over 75% of which occurred only once. This inevitably results in a large pool over out-of-vocabulary words. Pre-trained word embeddings trained on a relatively small corpus would be unable to model the vocabulary of the DEFT corpus completely, particularly as the latter mostly comprises of domain-specific text. This could potentially be mitigated by restricting the vocabulary based on term frequency count, but care must be taken not to restrict it too much as definitions, by definition, are dependent on uniquely identifiable terms.

We also found several incongruities in the corpus where contradictions in annotations led to an ambiguous ground-truth. Consider the following sentences from the training corpus: *"Organisms are individual living entities."* and *"Organelles are small structures that exist within cells."* The first sentence was annotated with the positive class (contains a definition) even though the latter was not. Another similar albeit more ambiguous example: *"Recall from The Macroeconomic Perspective that if exports exceed imports, the economy is said to have a trade surplus."* and *"If imports exceed exports, the economy is said to have a trade deficit."* Here, the second sentence is tagged as containing a definition even though the first isn't. While some of these ambiguities can be attributed to how the training data for the binary classification task is generated from the larger sequence-annotated corpus, there are many other counter-examples where the rationale behind the annotation is unclear. Such incongruities ultimately make it more challenging for the model to attain a clear and optimal generalization.

And finally, on perhaps a more mundane note, real-world issues such as confusing instructions from the task organizers and unreliable testing/submission infrastructure unfortunately also contributed to a less-than-optimal evaluation phase for our team.

4 Conclusion

We presented our system for definition extraction whose pre-BERT methods achieved an admittedly pre-historic F1-score of 0.6851 in Task 6: DeftEval, subtask 1. Future work could potentially include the customization of the architecture to incorporate ensemble training, exploring the usage of more task-specific cues such as topical information, and perhaps even becoming one with the BERT side and using contextualized word embeddings.

[6] Since the gold-standard data for the test set was not available at the time of publication, we based our statements on our analysis of the training corpus and the prediction results of the test data.

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.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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*.

Keiron O'Shea and Ryan Nash. 2015. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.

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.