DSC IIT-ISM at SemEval-2020 Task 6: Boosting BERT with Dependencies for Definition Extraction

Aadarsh Singh*, Priyanshu Kumar* and Aman Sinha

Indian Institute of Technology (Indian School of Mines) Dhanbad, India
{aadarshsingh191198, kpriyanshu256, amansinha091}@gmail.com

Abstract

We explore the performance of Bidirectional Encoder Representations from Transformers (BERT) at definition extraction. We further propose a joint model of BERT and Text Level Graph Convolutional Network so as to incorporate dependencies into the model. Our proposed model produces better results than BERT and achieves comparable results to BERT with fine tuned language model in DeftEval (Task 6 of SemEval 2020), a shared task of classifying whether a sentence contains a definition or not (Subtask 1).

1 Introduction

Definition Extraction from free text (DEFT) (Spala et al., 2020) involves finding term definition pairs from free and semi-structured texts, especially those whose term-definition span crosses a sentence boundary and those which do not have a definition phrase. An example of a cross boundary sentence from the DEFT corpus is given below:

> In doing so , <DEF> monomers release water molecules as byproducts <DEF> (1). This type of reaction is known as <TERM> dehydration synthesis <TERM> , which means " <QUALIFIER> to put together while losing water <QUALIFIER> . "(0)

It can be observed that the definition (enclosed by DEF tags) and the corresponding terms (enclosed by TERM tags) are present in different sentences, thus increasing the difficulty of definition extraction. The task is thus relatively different and complex from the conventional definition extraction (DE) task in which a definition could be broken into the following sub-parts:

1. The DEFINIENDUM field (DF) i.e., the word being defined.
2. The DEFINITOR field (VF) i.e., the verb phrase used to introduce the definition.
3. The DEFINIENS field (GF) i.e., genus phrase or the hypernym.
4. The REST field (RF) i.e., additional clauses that help to distinguish the definiendum from its genus.

and thus easily be captured by common verb phrases (DEFINITOR) like "means", "refers to", "is", etc. These kind of conventional definitions could easily be tagged as follows:

> <DF>Photosynthesis</DF> <VF>is</VF> <GF>the process </GF> <RF>by which green plants manufacture food. </RF>

In the example presented above, the definition (REST field) and the corresponding term (DEFINIENDUM field) are present in the same sentence. Moreover, the presence of DEFINITOR(s) in the text also eases the task of extracting such definitions.

* Equal contribution.

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 710–716
Barcelona, Spain (Online), December 12, 2020.

The order of the tags in the WCL Corpus (Navigli et al., 2010), one of the conventional corpus for Definition Extraction task, is predefined viz. DEFINIENDUM, DEFINITOR, DEFINIENS, REST. On the other hand, there is no such predefined order for the DEFT corpus regarding the occurrence of BIO Tags. This absence of order makes it difficult for the models to identify the definitional sentences in the DEFT corpus and then associate the words present in the definitions with the appropriate tags.

Another observation is the variation in the pattern of the definitional sentences in the DEFT corpus due to the presence of the heterogeneous distribution of tags. Some tags like 'O','I-Term','I-Definition', etc. are very frequent whereas tags such as 'Alias-Term', 'Secondary-Definition' are rarely seen. This imbalance in dataset causes difficulty in finding a structure for the definition and hence makes the DEFT corpus relatively more complex when compared to conventional corpora like WCL.

Thus, given the complexity of the definitions present in the DEFT corpus, the whole pipeline for the extraction of meaningful term-definition pairs from free text can be restructured as presented in Figure 1.

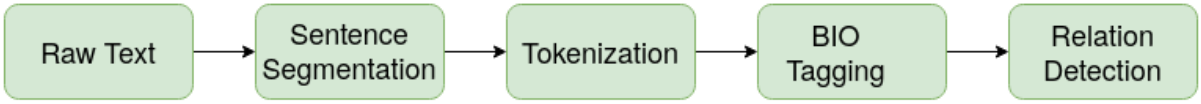

Figure 1: Pipeline for Definition Extraction

Since the corpus is already in the form of sentences, sentence segmentation is not required. For tagging tokens, authors of DEFT Corpus (Spala et al., 2019) have defined a new annotation scheme, a part of which has been touched upon in the introduction section. Moreover, it can be found in the corresponding paper.

In this paper, we present the approach used for the task "DeftEval: Extracting term-definition pairs in free text" of SemEval 2020. The subtasks are as follows:

1. Subtask 1 - Sentence Classification
2. Subtask 2 - Sequence Labeling
3. Subtask 3 - Relation Classification

Our presented methods for the sentence classification subtask (Subtask 1) revolve around the Transformer (Vaswani et al., 2017) based model, Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018). Experiments [1] have been done to improve the results and in the process we have come up with improvised architectures.

The rest of the paper is organized as follows: Related work with its limitations has been discussed in Section 2, followed by a description of the data used in Section 3. The proposed methods have been elaborated in Section 4 [2]. Section 5 and 6 contains the results and error analysis respectively. Section 7 concludes the paper and also includes the possible future work.

2 Related Work

The very first attempt at DE was by Kobyliński and Przepiórkowski (2008). They used a "Balanced Random Forest classifier" to classify definitions so as to handle the imbalance of the dataset. A major contribution to DE was by Navigli and Velardi (2010), by using a generalization of word lattices to model textual definitions in the form of Definiendum, Definitor, Definiens and Rest. However, this approach is unable to generalise definitions, especially the ones that defy the conventional semantics i.e. common patterns found in definitional sentences.

[1] Corresponding submissions have been made under the username of `mler` on Codalab

[2] Source code available at `https://github.com/dsciitism/SemEval-2020-Task-6`

Many existing works take advantage of linguistic features like syntactic dependencies. Jin et al. (2013) built DefMiner which is a supervised sequence labelling system using shallow parsing and dependency features. Training CRF (Conditional Random Fields) with lexical, terminological, and structural features extracted from data has also been tried by Anke (2013). The use of linguistic features lead to promising results and observations. Espinosa-Anke and Saggion (2014) took advantage of the subtrees in the dependency parsing of a sentence. Feature representation of sentences were created using these subtrees. Espinosa-Anke et al. (2015) incorporated both linguistic and semantic features of sentence for training classifiers. SensEmbed is used to reveal the semantic compactness of definition containing sentences.

With the advent of the deep learning era, the first attempt at DE using deep learning was by Li et al. (2016). Feature representation of sentences were learned using Long Short Term Memory (LSTM) cells. Text preprocessing such as replacing some selected words with POS tags, were carried out thus obtaining brilliant results. Anke and Schockaert (2018) developed Syntactically Aware Neural Networks by incorporating syntactic information (syntactic dependencies and dependency labels) along with the text sentences as input. With the help of pre-trained word embeddings, convolutional filters and Bi-LSTM cells, the model was capable of extracting both short range and long range dependencies from the text data.

In the most recent attempt at Definition Extraction, Veyseh et al. (2019) proposed a multi-task model to perform sentence classification and sequence labelling simultaneously. The model took advantage of the entire syntactic dependency tree rather than just dependencies, thus yielding state of the art results on the WCL dataset.

3 Dataset

For the purpose of training, evaluation and testing, the DEFT corpus has been used. Not only is the corpus significantly larger than the previously available corpora in the field of Definition Extraction, but the dataset also contains definitions from complex, human-annotated data across a variety of topics and from both free (textbook) and semi-structured (legal document) language. Table 1 shows the statistics of the major datasets available in the field of DE namely, WCL, W00 (Jin et al., 2013) and DEFT.

Dataset	No. of positive annotations	Size(in sentences)
WCL	1,871	4,718
W00	731	2,185
DEFT	11,004	23,746

Table 1: Dataset statistics

By complicated structure, we mean that the corpus does not contain simple sentences of the form "X is a Y" as is the case in most of the definitions present in the WCL dataset. Instead, roughly 50% of term-definition pairs in the dataset appear across sentence boundaries or with an otherwise complex structure (e.g., containing secondary information, containing ambiguous references to previously stated terms or definitions) whereby the relationship between a term and definition requires more deduction than finding a definition verb phrase.

4 Methods

4.1 Data Preprocessing

Unlike conventional approaches for text classification (Tf-Idf, Bag of Words Model, etc.), deep learning approaches require minimal preprocessing. It is often believed that preprocessing leads to the loss of information. So, for preprocessing the we have only performed the following two steps:

1. Removal of leading line numbers: Some of the sentences in the data have a leading line number. It has been removed since it is irrelevant to the content of the sentence and acts as noise.

2. Addition of the subject token: It has been found in some cases of our experiments that adding the subject token i.e. the subject of the textbook from which the sentence to be classified is picked, helps improve the results.

4.2 Model Architectures

Initially, we started off by using the approach of Syntactically Aware Neural Networks. However, owing to the complexity of the DEFT corpus, the model was unable to perform well. Hence, we shifted towards using larger pretrained models such as Transformers. We concentrate on the performance of BERT since it has achieved the State of The Art results in many NLP tasks with limited fine-tuning on task-specific training data. Moreover, Kumar et al. (2020) have shown that BERT implicitly captures syntactic dependencies, which play an essential role in Definition Extraction.

We now describe the three approaches that we have implemented and submitted to the shared task.

1. **BERT** : BERT, based on the Transformer architecture, consists of multi-attention heads which apply sequence-to-sequence transformation on the input text sequence. BERT incorporates the following practices for training (a) learn to predict a masked token using the left and right context of the text sequence (Masked Language Model) (b) learn to predict whether two sentences occur in continuation or not (Next Sentence Prediction)

   For our experiments, we use the BERT (base-cased) made publicly available by Huggingface (Wolf et al., 2019). It consists of 12 hidden layers in the encoder of the Transformer. The encoder outputs a feature vector of 768 dimensions. We select the cased model (pre-trained on cased English text) because lower casing the data leads to loss of information. The representation corresponding to the CLS token is fed through two feed-forward linear layers, thus giving two values corresponding to the logits of the two classes. We train the model for 5 epochs using AdamW (Loshchilov and Hutter, 2017) optimizer with a learning rate of 2e-5.

2. **BERT with fine tuned language model** : This model is the same as the first one, except we fine tune the Masked Language Model of BERT using the training data. This helps BERT to understand the context of the corpus in a better manner. Results show that fine tuning the language model of neural networks gives improved results (Howard and Ruder, 2018).

   To fine tune the LM, we take help of Huggingface's *run_language_modeling.py* utility[3]. The other hyperparameter settings remain the same as the first model.

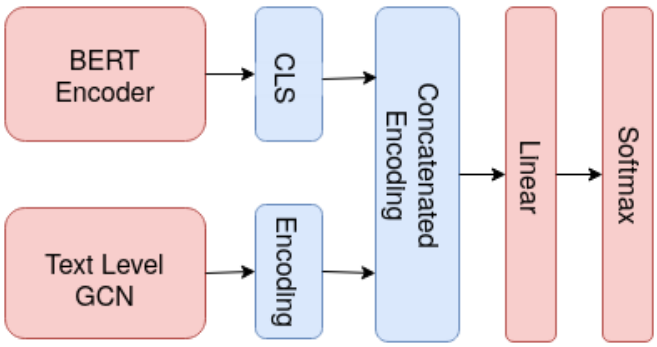

Figure 2: Joint model of BERT and Text Level GCN

3. **Joint model of BERT and Text Level GCN** : The above models do not take into account linguistic features along with the text data. Researchers have obtained superior results by incorporating linguistic information into the models. However, creating such features requires extra effort and involves the use of dependency parsers, which may lead to error propagation (error in extracting linguistic

[3] https://github.com/huggingface/transformers/tree/master/examples/language-modeling

information will lead to feeding of erroneous information to the model). Graph Convolutional Networks (GCN) (Kipf and Welling, 2016) have yielded superior performance on graph-structured data. Since text can also be represented as graph data (for example syntactic dependency trees), we use the feature representation obtained from a Text Level Graph Convolutional Network (Huang et al., 2019) along with the feature representation obtained from BERT, to solve our task. As shown in Figure 2, both the representations are concatenated and a linear layer is used to output the logits of the two classes.

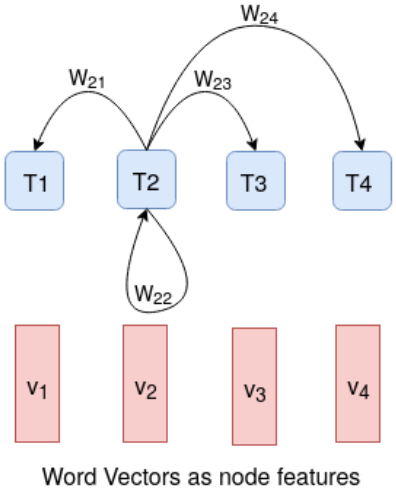

Figure 3: Inner Working of Text Level Graph Convolutional Networks

In a GCN, data is represented as graph(s) and each node of the graph is represented by a set of attributes. A Text Level GCN represents a text sequence as a graph. Each token of the sentence represents a node and has its corresponding word vector as attributes. A token has an edge between its n-grams to the left and right of it (n is also referred as window size). The greater the value of n, greater is the range of dependencies that are captured. Figure 3 shows the corresponding edges for a token (T2) of a text sequence with 4 tokens, considering a window size of 2.

The weights of the edges are learned through training. The weighted representations (product of edge weight and word vector) is sent to its neighbours by each node. Each node gathers information from its neighbouring nodes, and uses it to update its own representation (i.e. attributes) and edge weights. The values of the edge weights represent the significance of the dependency between a token and one of its n-grams. Thus, the model understands the difference between strong and weak dependencies without any human intervention.

The language model of the BERT component of the model is not fine tuned. We use the publicly available pre-trained weights. The Text Level GCN component uses pre-trained GloVe word embeddings (Pennington et al., 2014) as node attributes. We use a window size of 5 in our experiments. The model is trained for 5 epochs using AdamW with learning rate 2e-5.

5 Results

We evaluate the performance of the three models on the validation set and the test set. The evaluation metric is F1 score on the positive class. The submissions of the test set were done using a 10-fold cross validation so as to increase the robustness of the results. The results are presented in Table 2.

The results show that fine tuning the language model of BERT leads to improved results. The fine tuned BERT achieves the greatest F1 score on the test set among the 3 models. The reason for such a performance boost can be attributed to the structure of the DEFT corpus. The corpus consists of long continuous segments from textbooks and legal documents. Since there is a lot of text in the same context, fine-tuning the language model helps BERT to get a better understanding of the domain of the corpus.

Model	Validation Set	Test Set
BERT	0.74	0.731
BERT with fine tuned language model	0.768	**0.775**
Joint model of BERT & GCN	**0.781**	0.758

Table 2: Comparison of results of the mentioned models

It is also evident from Table 2 that the inclusion of Text Level GCN along with BERT is also beneficial. The Text Level GCN component of the joint model is capturing some extra information from the text data which is boosting the performance.

6 Error Analysis

We analyze the predictions on the validation set. For the predictions of all the models, we calculate the binary cross entropy loss for each example and then sort the examples as per descending loss values. We examine the top commonly mis-classified examples by each of the 3 models (in Table 3).

Sentence	True Label
”United States v. Miller , 307 U.S. 174 (1939) .	1
Pathogens include bacteria , protists , fungi and other infectious organisms .	1
Toll goods are available to many people , and many people can make use of them , but only if they can pay the price .	1

Table 3: Common highly mis-classified sentences

We observe that majority of these examples belong to the cross sentence definition scenario as mentioned in Section 1. This implies that models are struggling to extract cross-sentence definitions. The models lack a context to the sentence to be classified, because of which it is unable to classify definitions that are covered in more than one sentence. To the human reader, the predictions of most of mis-classified sentences would appear correct. However, they are incorrect according to DEFT corpus.

An additional reason for the errors is the incorrect/ambiguous labelling of the dataset. For example, consider the sentence ‘”United States v. Miller , 307 U.S. 174 (1939) .’ in table 3. The sentence is not a definition but due to the inverted comma (”) of the trailing sentence (which was a definition), this sentence was also labelled a definition. Similarly, the label for the sentence “Pathogens include bacteria , protists , fungi and other infectious organisms .” is debatable as it seems more of a description than a definition. Last but not the least, “Toll goods are available to many people , and many people can make use of them , but only if they can pay the price .” is clearly not a definition.

7 Conclusion and Future Work

We study the performance of BERT on the DEFT corpus and tried to boost BERT with explicit linguistic information (in the form of dependencies) using a Text Level GCN model. For further improvements, we can use BERT with a fine-tuned language model as the component of the joint model. The models can also make use of a context in the form of the sentence preceding the sentence to be classified. The incorporation of a context can help models overcome the problem of misclassifying cross-sentence definitions.

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.

Luis Espinosa Anke. 2013. Towards definition extraction using conditional random fields. In *Proceedings of the Student Research Workshop associated with RANLP 2013*, pages 63–70.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Luis Espinosa-Anke and Horacio Saggion. 2014. Applying dependency relations to definition extraction. In *International Conference on Applications of Natural Language to Data Bases/Information Systems*, pages 63–74. Springer.

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.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Lianzhe Huang, Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng WANG. 2019. Text level graph neural network for text classification. *arXiv preprint arXiv:1910.02356*.

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.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Łukasz Kobyliński and Adam Przepiórkowski. 2008. Definition extraction with balanced random forests. In *International Conference on Natural Language Processing*, pages 237–247. Springer.

Priyanshu Kumar, Aadarsh Singh, Pramod Kumar, and Chiranjeev Kumar. 2020. An explainable machine learning approach for definition extraction. In Arup Bhattacharjee, Samir Kr. Borgohain, Badal Soni, Gyanendra Verma, and Xiao-Zhi Gao, editors, *Machine Learning, Image Processing, Network Security and Data Sciences*, pages 145–155, Singapore. Springer Singapore.

SiLiang Li, Bin Xu, and Tong Lee Chung. 2016. Definition extraction with lstm recurrent neural networks. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, pages 177–189. Springer.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

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.

Roberto Navigli, Paola Velardi, Juana María Ruiz-Martínez, et al. 2010. An annotated dataset for extracting definitions and hypernyms from the web. In *LREC*.

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.

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.

Sasha Spala, Nicholas Miller, Franck Dernoncourt, and Carl Dockhorn. 2020. Semeval-2020 task 6: Definition extraction from free text with the deft corpus. In *Proceedings of the 14th International Workshop on Semantic Evaluation*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Amir Pouran Ben Veyseh, Franck Dernoncourt, Dejing Dou, and Thien Huu Nguyen. 2019. A joint model for definition extraction with syntactic connection and semantic consistency. *arXiv preprint arXiv:1911.01678*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.