DeftPunk at SemEval-2020 Task 6: using RNN-ensemble for the sentence classification

Jekaterina Kaparina
jekaterina.kaparina@student.uni-tuebingen.de

Anna Soboleva
anna.soboleva@student.uni-tuebingen.de

Abstract

This paper describes participation in DeftEval 2020 (part of SemEval sharing task competition), and is focused on the sentence classification. Our approach to the task was to create an ensemble of several RNNs combined with fasttext and ELMo embeddings. Results show that various types of models in an ensemble give a performance boost in comparison to standard models. Our model achieved F1-score of 78% for a positive class on the DeftEval dataset.

1 Introduction

Definition extraction has been a popular topic in NLP for a long time. Early research in this field dates back to the end of the 20th century (Nakamura and Nagao, 1988). Throughout the last two decades numerous methods to solve the problem were proposed: rule-based approaches (Storrer and Wellinghoff, 2006; Pollak et al., 2012), machine learning approaches (for example, genetic algorithms (Borg et al., 2009), probabilistic topic models (Faralli and Navigli, 2013)) etc. Interest in the definition extraction can be explained by the fact that it is applicable in a large number of areas, including e-learning (Monachesi and Westerhout, 2008), glossary creation (Park et al., 2002), Word Sense Disambiguation (Camacho-Collados et al., 2015) and Information Extraction (Delli Bovi et al., 2015).

The goal of the task was to correctly label sentences as definitions or non-definitions. According to the organizers, most of the research in the definition extraction was based on the assumption that definitions can be retrieved by detecting keywords or syntactic structure, following the so-called Aristotelian idea of the definition: "X is Y, which is..." (Spala et al., 2019). The corpus, used for this task, is focused on other types of definitions which are harder to capture with simple pattern detection.

Spala et al. (2019) summed up two main approaches to classification of definitions: usually the main method was either pattern matching or hypernym detection. They also noted that previous research was aimed primarily at English sources and that the number of available definition extraction corpora is quite limited. Perhaps because the first research in the field was conducted relatively long ago and the open source culture was not as strong, a large number of corpora remained private.

This is especially visible in Espinosa-Anke et al. (2015), where seven Definition Extraction corpora are listed, but only two of them are easily accessible.

Apart from the DEFT corpus, used in this shared task, to our knowledge, only two other definition extraction corpora are publicly available: so-called W00 corpus (Jin et al., 2013) and WCL corpus (Navigli et al., 2010). Unfortunately, their material mostly follows the Aristotelian model of definition, hence, they were ineffectual as additional training data.

It is noticeable that the problem of definition extraction has become less fundamental in recent years. For example, the state-of-the-art model, described in Espinosa-Anke and Schockaert (2018) is one of the few examples of a neural network approach to this problem, namely, a Bi-LSTM, where features are learned via convolved filters (LeCun et al., 1998). We used this approach as one of our baseline models.

Our main model uses an ensemble method, combining several RNNs (described in details in the Section 3). Some classical approaches were also applied to the task (e.g., pattern matching) and experiments with data augmentation techniques were conducted. Some of the resulted models were used as baselines.

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 697–703
Barcelona, Spain (Online), December 12, 2020.

2 Data

The corpus, used for the task, consists of 21,303 sentences, taken from freely available school textbooks. The bold typed words inside textbooks were considered as terms and it was assumed that the sentence around them is a definition. Personalities and places were excluded from the corpus. The topics of the textbooks belonged to one of these categories: biology, history, physics, psychology, economics, sociology, government.

For the Subtask 1, we were provided with the train and dev sets, where the single sentence was labeled in the binary manner (either "0", or "1").

Our data preprocessing was limited to the text tokenization and removal of tags and numbers, which did not carry any semantic meaning, such as numbers of sections or links.

3 Model

An ensemble of models can boost the performance of single classifiers by joining their best qualities (Krogh and Vedelsby, 1995; Hansen and Salamon, 1990). Our ensemble consists of three networks outlined below. The visualization of the full system is shown in Figure 1.

Two models (3.1 and 3.2) use a bidirectional LSTM network (Graves and Schmidhuber, 2005; Hochreiter and Schmidhuber, 1997), which proved to be an effective NLP tool for handling text sequences (Lipton et al., 2015), preserving long-term dependencies, as well as context. The third model (3.3) uses a bidirectional GRU (Cho et al., 2014) instead of LSTM due to its simple architecture. Hyperparameters were chosen empirically, resulting in a batch size of 64, hidden neurons of 256 and the dropout of 0.3 on each LSTM layer. We evaluate all models every 300 steps on the development set and after two evaluation periods without an improvement we scale the learning rate by 0.3. We also apply batch normalization before the output layer (Ioffe and Szegedy, 2015). To counter the class imbalance in the training data, we weight the loss of positive examples higher than those of negative ones.

The final prediction of the ensemble is the average of the probabilities obtained from each classifier. Every example that received an averaged probability over the empirically chosen threshold of 0.53 receives the label *definition*. All models were trained on single customer grade GPU.

3.1 Joint-topic Bi-LSTM over fasttext embeddings

This model uses pretrained[1] fasttext embeddings as word representations (Bojanowski et al., 2017), extracted with `finalfusion`[2] library.

The network is essentially a two-layered Bi-LSTM over fasttext embeddings. To accomodate for the different domains covered by the various topics of the training data, we include the topic of each example as an additional training target. We hypothesize that making the classifier topic aware helps identifying domain-specific language used for definitions. Topic labels are extracted from the training data file names. Following the low-supervision paradigm (Søgaard and Goldberg, 2016), we place the topic classifier on the first layer and the definition classifier on the output of the last layer. We sum the unweighted loss of both tasks. The probability predicted by this model is obtained by passing it through the logistic function.

3.2 Joint-definition-word Bi-LSTM over fasttext embeddings

Similar to the previous model, this network is a two-layer Bi-LSTM over fasttext embeddings. We set an additional training target to detect the definition word itself, after predicting whether a sentence is a definition. This strategy is motivated by how textbooks, where the training data comes from, emphasize the definition words in bold. We hypothesize that it will force the network to pay more attention to specific words when making the decision on the sentence level. We make use of this assumption through a binary definition classifier that predicts whether each word in a sentence is a definition word. It uses IOB-tags labels extracted from the original corpus. We place the definition classifier on the first layer and its output is the final output of the model. The aforementioned definition word classifier is placed on the second

[1] https://fasttext.cc/docs/en/crawl-vectors.html

[2] https://github.com/finalfusion/finalfusion-python

layer. As in the previous model, we sum the unweighted loss of both tasks and obtain the probability of the whole model.

3.3 Bi-GRU with ELMo

In contrast to our other models, ELMo contextualized word-representations (Peters et al., 2018) start with word representations build from convolutions over characters. Compared to other contextualized word-representations, such as BERT, ELMo, fits our computational budget as it is much more lightweight in terms of parameter count. We use the output of all ELMo layers and apply the standard scalar-weighting approach to combine them. The resulting vectors are fed as input to a Bi-GRU. The final-states of the last layer of the Bi-GRU are used to perform the binary classification.
We applied dropout of 0.4 to each direction of a GRU layer and a mixed precision to make the model more efficient. ELMo was applied using Tensorflow implementation.[3]

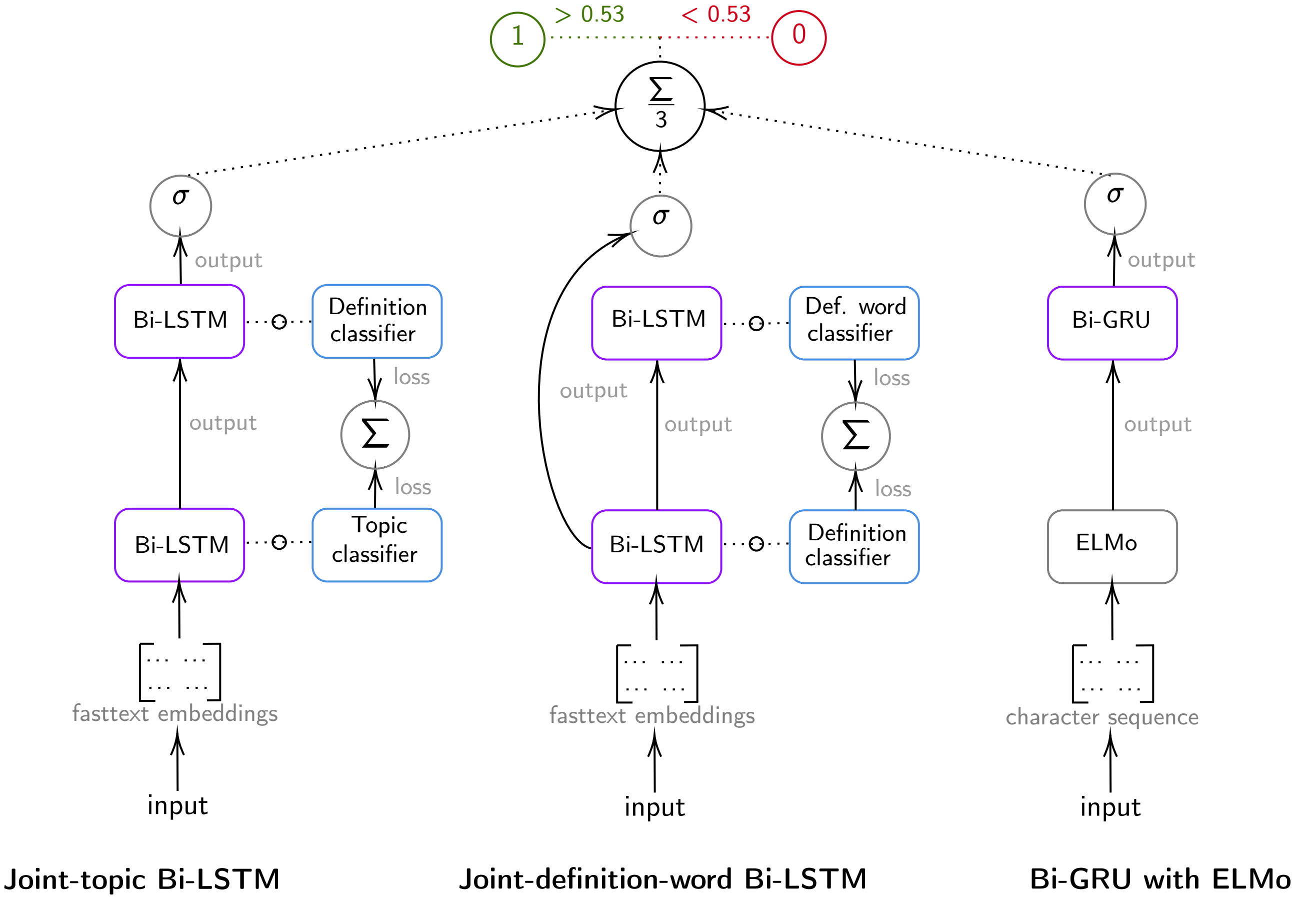

Figure 1: This figure depicts the structure of the ensemble network.

4 Evaluation

This section describes evaluation baselines. We used various types of models in attempt to show possible diversity in the performance of this task.
Neural_de: "Neural_de" (Espinosa-Anke and Schockaert, 2018) is the name of the already mentioned state-of-the-art model for this task. Its code is freely available. We adapted it slightly for our needs, leaving the initial architecture and embedding the same, and used it on our data.
SVM: the SVM (Cortes and Vapnik, 1995), often used in classification problems, is one of the older classical approaches. In comparison to modern Deep Learning methods, it does not require a lot of data, which makes it quite competitive when dealing with low resource data. Under specific circumstances it can

[3] https://tfhub.dev/google/elmo/2

perform even better than traditional Deep Learning approaches (Liu et al., 2017). We used Bag-of-ngrams (uni-, bi- and trigrams) for text representation, resulting in the vocabulary of size 457015 items. Additional ngrams of POS and dependency tags were used as features, which we extracted with `Spacy`.[4]

Rule based approach: In an attempt to try a more classical approach to the definition extraction and used simple rule-based model, created with the `Spacy` library. We had three types of patterns we were looking for: simple word constructions (as "that occurs", "refer to" or "be known", where all the verbs were matched on the lemma level), syntactic patterns, which were heavily relying on POS tags (for example, "NOUN" + "be" + "DET" + "NOUN) and universal dependency based patterns.

Logistic regression: Another baseline model was a Logistic Regression with a simple tf-idf embeddings. It is commonly used as a baseline for various machine learning problems.

Bi-LSTM over character Bi-LSTM: A model more related to our approach is a Bi-LSTM network that contains one layer with a definition sentence classifier and uses character based word representations. These are created by running another Bi-LSTM network with 64 hidden neurons over character embeddings. The final state of the forward and backward direction are concatenated and used as word representations. Due to the memory limitations, we truncate words at 50 characters.

Simple Bi-LSTM with 1 layer: A simple Bi-LSTM network with fasttext embeddings was used to evaluate our models against similar system for a fair comparison. Therefore all parameters of this network are exactly the same as defined for our target LSTM networks.

Simple Bi-LSTM with 2 layers: This model is the same as previously described network, except that it has 2 layers.

5 Results

Results are shown in the results-table. Due to several bugs, caused by differences in preprocessing of test and train sets, the submitted results differ from those reported in this paper. These have been fixed and the table represents the true scores.

Model	F1 (1)	F1 (0)	F1 avg	Precision	Recall	Accuracy
Logistic regression	0.56	0.85	0.70	0.77	0.69	0.77
Rule based system	0.41	0.81	0.61	0.67	0.61	0.71
SVM	0.42	0.73	0.62	0.61	0.63	0.63
Neural_de	0.66	0.84	0.60	0.76	0.75	0.78
Bi-LSTM over char. Bi-LSTM	0.50	0.58	0.53	0.51	0.54	0.54
Simple Bi-LSTM with 1 layer	**0.70**	0.86	**0.78**	0.79	**0.77**	0.81
Simple Bi-LSTM with 2 layers	0.69	**0.87**	0.78	**0.79**	0.77	**0.81**
Joint-definition-word Bi-LSTM	0.72	0.86	0.79	0.79	0.80	0.81
Joint-topic Bi-LSTM	0.74	0.88	0.81	0.81	0.81	0.83
Bi-GRU with ELMo	**0.76**	**0.89**	**0.83**	**0.83**	**0.82**	**0.85**
Ensemble	**0.78**	**0.90**	**0.84**	**0.85**	**0.83**	**0.86**

Table 1: This table demonstrates the evaluation results. First section of the table contains baseline models, second section consists of implemented single models and final section shows the results of the ensemble of single models. F1 (1) column represents F1-scores for a positive class (*definition*) and F1 (0) for negative class (*not a definition*). Best scores across networks in each sections are marked with **bold**.

Baseline: The results show that among baseline models, almost all deep learning networks outperform classical approaches. Logistic Regression gave better than expected results, achieving 77% of accuracy on the test set, but ultimately reaching unsatisfactory results on the F1 score for positive label. Worse performance of the rule based system could be due to instability of the structure of the definition sentences,

[4] `https://spacy.io/`

as well as lack of rules and linguistic features. The low performance of SVM can be explained by a big difference in the vocabulary of train and test sets, resulting in poor word representation of test samples. We can also observe no significant difference between the Bi-LSTM network with 1 and 2 layers.
It is quite difficult to reason the bad performance of the Bi-LSTM over character embeddings, but perhaps the character network is too small and having more computational power to handle a bigger architecture could improve it. Low performance of neural_de model could be explained by the fact, that the model was created with the completely different format of the definition in mind. Overall, all these models could achieve much higher results with better tuning.
Experiment: All implemented single models outperform baseline networks that we have presented. The results demonstrate, that the presence of 2 Bi-LSTM layers does not guarantee the best results itself, but that additional classifiers are beneficial. Out of implemented models, even though LSTM is the state of the art technique, GRU can compete well, if used with other well performing networks (such as ELMo). Bi-GRU with ELMo demonstrates the best results across single models, reaching 76% F1 for a positive class.
Finally, the ensemble of all three models reaches the highest performance across all systems, improving over Bi-GRU with ELMo by a margin of 2% on F1 for a positive score, showing that the combination of strong models and the optimal threshold probability, can boost the overall results by joining each models' certainty about a specific decision.

6 Discussion

Considering that ensemble of several models proved to be a competitive approach to the definition extraction task, it seems logical to assume that adding more models can be beneficial too. It is frequently noted that the efficiency of the ensemble also raises via diversity of the included in it models (Tabik et al., 2020).
Another seemingly promising approach is data augmentation. It is evident, that there is a lack of annotated data and some attempts in improving it could be made. We attempted some simple data augmentation for this task, but results were not satisfying. We tried the approach described by Wei and Zou (2019), which we modified accordingly to our task logic, substituting only adjectives, adverbs and nouns with synonyms or adding a few additional ones, trying to preserve original syntactical structure, but the resulted data was unusable. There certainly could be interesting ways to create more data in the future: for example, by incorporating knowledge graphs, following the steps of Sharifirad et al. (2018).
Another potential approach to the task would be usage of other deep learning models such as BERT that currently achieves state-of-the-art performance for numerous other NLP tasks.

7 Conclusion

This work describes the results of participation in a competition DeftEval 2020 (SemEval 2020) Task 6 on Definition Extraction. We present an ensemble of two Bi-LSTM models over fasttext embeddings, using additional topic and a definition word binary classifiers, with a Bi-GRU network over ELMo. The results demonstrate that Bi-LSTM and Bi-GRU systems significantly outperform non deep learning models. We also show the benefit of using joint classifiers in comparison to having simple layers. Finally, we demonstrate that an ensemble of strong single systems can improve the performance, reaching 78% F1 score for a positive class.

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Claudia Borg, Mike Rosner, and Gordon Pace. 2009. Evolutionary algorithms for definition extraction. In *Proceedings of the 1st Workshop on Definition Extraction*, WDE '09, page 26–32, USA. Association for Computational Linguistics.

José Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. 2015. A unified multilingual semantic representation of concepts. In *ACL*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

Claudio Delli Bovi, Luca Telesca, and Roberto Navigli. 2015. Large-scale information extraction from textual definitions through deep syntactic and semantic analysis. *Transactions of the Association for Computational Linguistics*, 3:529–543.

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, New Orleans, Louisiana, June. Association for Computational Linguistics.

Luis Espinosa-Anke, Horacio Saggion, and Francesco Ronzano. 2015. Weakly supervised definition extraction. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 176–185, Hissar, Bulgaria, September. INCOMA Ltd. Shoumen, BULGARIA.

Stefano Faralli and Roberto Navigli. 2013. Growing multi-domain glossaries from a few seeds using probabilistic topic models. In *EMNLP*, pages 170–181.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural networks*, 18(5-6):602–610.

Lars Kai Hansen and Peter Salamon. 1990. Neural network ensembles. *IEEE transactions on pattern analysis and machine intelligence*, 12(10):993–1001.

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

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

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, Seattle, Washington, USA, October. Association for Computational Linguistics.

Anders Krogh and Jesper Vedelsby. 1995. Neural network ensembles, cross validation, and active learning. In *Advances in neural information processing systems*, pages 231–238.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition.

Zachary C Lipton, John Berkowitz, and Charles Elkan. 2015. A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*.

Peng Liu, Kim-Kwang Raymond Choo, Lizhe Wang, and Fang Huang. 2017. Svm or deep learning? a comparative study on remote sensing image classification. *Soft Computing*, 21(23):7053–7065.

Paola Monachesi and Eline Westerhout. 2008. What can nlp techniques do for elearning.

Jun-ichi Nakamura and Makoto Nagao. 1988. Extraction of semantic information from an ordinary English dictionary and its evaluation. In *Coling Budapest 1988 Volume 2: International Conference on Computational Linguistics*.

Roberto Navigli, Paola Velardi, and Juana Ruiz-Martínez. 2010. An annotated dataset for extracting definitions and hypernyms from the web. 01.

Youngja Park, Roy J. Byrd, and Branimir Boguraev. 2002. Automatic glossary extraction: Beyond terminology identification. In *COLING*.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Senja Pollak, Anze Vavpetic, Janez Kranjc, Nada Lavrac, and Spela Vintar. 2012. Nlp workflow for on-line definition extraction from english and slovene text corpora. In *KONVENS*.

Sima Sharifirad, Borna Jafarpour, and Stan Matwin. 2018. Boosting text classification performance on sexist tweets by text augmentation and text generation using a combination of knowledge graphs. In *Proceedings of the 2nd workshop on abusive language online (ALW2)*, pages 107–114.

Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 231–235.

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.

Angelika Storrer and Sandra Wellinghoff. 2006. Automated detection and annotation of term definitions in german text corpora. In *LREC*.

Siham Tabik, Ricardo F. Alvear-Sandoval, María M. Ruiz, José-Luis Sancho-Gómez, Aníbal R. Figueiras-Vidal, and Francisco Herrera. 2020. Mnist-net10: A heterogeneous deep networks fusion based on the degree of certainty to reach 0.1% error rate. ensembles overview and proposal. *Information Fusion*, 62:73 – 80.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6383–6389, Hong Kong, China, November. Association for Computational Linguistics.