TMLab SRPOL at SemEval-2019 Task 8: Fact Checking in Community Question Answering Forums

Piotr Niewiński, Aleksander Wawer, Maria Pszona, Maria Janicka
Samsung R&D Institute Poland
pl. Europejski 1
00-844 Warsaw, Poland
{p.niewinski, a.wawer, m.pszona, m.janicka}@samsung.com

Abstract

The article describes our submission to SemEval 2019 Task 8 on Fact-Checking in Community Forums. The systems under discussion participated in Subtask A: decide whether a question asks for factual information, opinion/advice or is just socializing. Our primary submission was ranked as the second one among all participants in the official evaluation phase. The article presents our primary solution: Deeply Regularized Residual Neural Network (DRR NN) with Universal Sentence Encoder embeddings. This is followed by a description of two contrastive solutions based on ensemble methods.

1 Introduction

Community question answering forums are good platforms for knowledge sharing; hence, they are widely used sources of information. The growing popularity of such knowledge exchange leads to a growing need to automate the process of verifying the post quality. The first step, often overlooked, is to categorize each question and establish what kind of information the user seeks.

Question classification has been mainly used to support question answering systems. Two main method types have been proposed in the literature: (1) rule-based approaches with linguistic features (Tomuro, 2004; Huang et al., 2008; Silva et al., 2011), and (2) machine learning approaches (Zhang and Lee, 2003; Metzler and Croft, 2005). These methods are rather simple, due to the fact that question classification is often just a preprocessing step in a larger task. However, we can observe some recent advances in this area, such as ULMFiT (Howard and Ruder, 2018), which achieves state-of-the-art performance on the TREC dataset (Voorhees and Tice, 1999).

The present article describes our systems submitted to the SemEval 2019 competition Task 8 subtask A on question classification. The competition data set consisted of QatarLiving forum questions classified as FACTUAL, OPINION or SOCIALIZING. The training data contained only 1,118 questions. Moreover, according to our evaluation, human-level accuracy on this data set was about 0.75, which was relatively low. Therefore, we approached the task as a challenging classification problem.

The article is structured as follows. Section 2 presents our experiments with preprocessing methods. Section 3 describes our official submission, where we propose an architecture utilizing several regularization methods to address the problem of the small data set. For comparative purposes, section 4 presents two ensemble models as contrastive examples. Section 5 provides the results achieved by the models. Lastly, section 6 concludes the discussion.

2 Data Preprocessing

We tested a few simple text preprocessing setups. Unfortunately, none of them helped the models achieve improved results. Hence, they are here presented as negative results, and for reference.

First, all emojis were removed from the text, and all URLs were replaced with the string 'url link'. Next, all dates and hours were replaced with 'date' and 'hour', respectively. Ordinal numbers – i.e. 1^{st}, 2^{nd}, 5^{th} etc. – were replaced with 'nth', while the remaining numbers were substituted wtih 'num'. All of these sequences were found using regular expressions. Furthermore, if most of the letters were uppercase, the whole text was lowercased.

Second, some of the forum-specific jargon was replaced with more generally used terms. This was achieved by an internally prepared dictionary that translated 'qar' into 'Qatar currency', 'qling' into

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 1172–1175
Minneapolis, Minnesota, USA, June 6–7, 2019.

'browsing Qatar forum', 'ql' into 'Qatar forum', 'villagio' to 'Qatar shopping center', etc. Additionally, it helped us to correct common spelling errors, such as 'doha' for 'Doha' and 'qatar' for 'Qatar'. Finally, spelling correction was performed by a custom character-based CNN language model. This way, we hoped to obtain a better representation of texts when embedded into vectors.

However, the experiments showed that none of these methods brought significant improvement in classification accuracy. It seemed that noise removal, combined with text normalization, deprived the data of significant features and information which carried crucial meaning for preparing text embeddings. Therefore, we finally did not perform any preprocessing and worked on raw question subjects and body text.

3 Primary Submission

3.1 Features

The feature space for the models was created by combining three different sources of information:

1. *Universal Sentence Encoder* – The concatenated post subject and body text were embedded with the Universal Sentence Encoder (USE) (Cer et al., 2018) to create a 512-dimensional vector representation.

2. *fastText embeddings* – The concatenated post subject and body text were tokenized with the Spacy library and embedded on the word level with 300-dimensional fastText vectors. Then, the vectors were averaged on the sequence dimension.

3. *Category statistics* – For each QL post category, the ratio of the FACTUAL, OPINION and SOCIALIZING labels was calculated. The three numbers were normalized, forming a 3-dimensional vector.

The three subfeature vectors were concatenated to produce an 815-dimensional vector for each question.

3.2 Model Architecture

We proposed the Deeply Regularized Residual Neutral Network architecture, shown in Figure 1.

The model took as its input the 815-dimensional vector of floats (concatenated USE embeddings, fastText embeddings averaged, and category statistics). During the training, a large dropout of 0.73 was applied to the input vector.

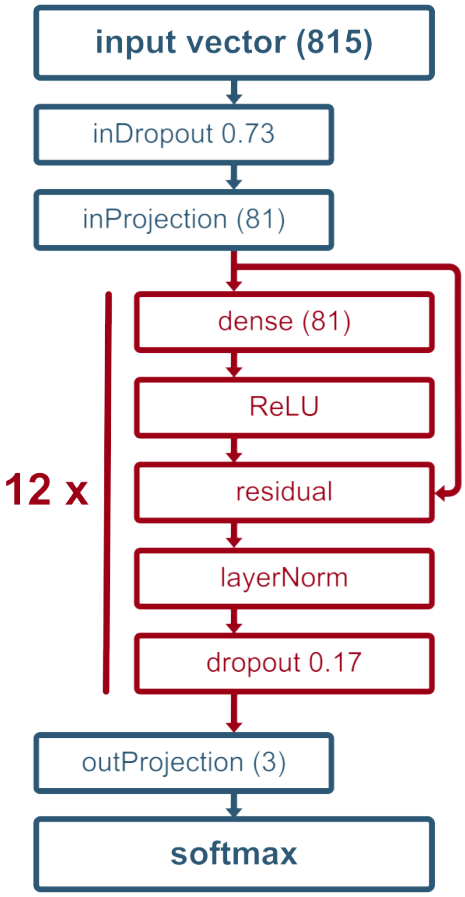

Figure 1: The architecture of DRR NN (primary submission).

The core of the model was a deep subnetwork built of 12 stacked blocks. Each block contained an 81-dimensional dense layer followed by ReLU activation, residual connection, layer normalization and 0.17 dropout. Finally the output of the last block was projected with a dense layer into a 3-dim logits vector.

The model was trained with the Adam optimizer, at a 6e-3 learning rate, and with 500-epoch linear warmup. We used softmax cross entropy loss with 0.14 of L2 penalty regularization.

All model hyperparameters were optimized with a randomized search algorithm and 5-fold cross-validation over the training data set. The final model size was 148K learnable variables.

3.3 Model Training

The main idea behind the advanced training procedure was to split the training data into a bigger learning part and a smaller validation part. The loss was minimized on the learning part until the accuracy on the validation part began to increase.

Generally, model performance depends on many factors, such as training efficiency, model architecture, optimization algorithms, etc. At the same time, it is affected by sample distribution between learning and validation parts.

In order to aggregate more knowledge from the training data, we used 5-fold cross-validation splits. We prepared 4 such splits using different random seeds. This procedure gave us a total of 20 different pairs of learning/validation sets.

We set the maximum number of epochs to 700. The model was validated after each training epoch and saved until its classification accuracy improved. Usually, the accuracy was improving for about 300-600 epochs. For the final prediction, we used the argmax of the summarized softmax of 20 models:

$$\arg\max \sum_{k=1}^{20} \mathrm{softmax}(logits_k).$$

4 Contrastive Submissions

For the contrastive submissions, our overall idea was to utilize multiple models that were as varied as possible, and combine their outputs.

In the first step, we used the following systems to obtain label probabilities for each question:

- *ELMO* (Peters et al., 2018) – a deep, contextualized word representation to obtain sentence representation, followed by a neural network of two dense layers. We arrived at the following architecture and hyper-parameters during the optimization: a dense layer of 48 neurons (dropout 0.5), followed by a second dense layer of 10 neurons (dropout 0.5). When tested on the training data in cross-validation, this solution alone achieved a micro-accuracy of 0.72.

- *BERT* (Devlin et al., 2018) – a deep, bidirectional transformer model with sequence classification layers on the top. The BERT language model was pre-trained, so only the sequence classifier was initialized and trained on the SemEval data. We used the PyTorch implementation of the case-insensitive 'base' version[1] with the optimal number of epochs (10) determined on the development set. When tested on the training data in cross-validation, this solution alone achieved a micro-accuracy of 0.717.

- *Bag-of-words* – a machine learning solution based on character n-gram vectorization with TF-IDF weighting and a linear kernel SVM classifier. We used the implementation from the scikit-learn package (Pedregosa et al., 2011). When tested on the training data in cross-validation, this solution alone achieved a micro-accuracy of 0.699.

In the second step, we prepared two different ensemble models combining the probability outputs from *ELMO*, *BERT*, *Machine Learning* and *DRR NN*.

The first contrastive submission (**Contrastive-1**) used the SVM classifier with linear kernel. The second contrastive submission (**Contrastive-2**) was designed as a bagging classifier of 10 estimators, each a voting ensemble of logistic regression, random forest and SVM with linear kernel.

5 Results

Table 1 contains the results of the evaluation on the official test set. The primary submission and both contrastive submissions were presented during the official phase of the contest. After the official competition, we tested additional solutions. Surprisingly, we achieved the best results with the SVM classifier (RBF kernel) on the USE embedding (**Post-evaluation**).

Model	Accuracy	F1	AvgRec
DRR NN (Primary)	0.83	0.72	0.76
Contrastive-1	0.83	0.72	0.76
Contrastive-2	0.81	0.69	0.73
Post-evaluation	0.87	0.77	0.78

Table 1: Official results of our submissions on the test set.

6 Conclusions

According to the experiments, and as reflected in the results on the test set, the best performing system was DRR NN based on the Universal Sentence Encoder. We attributed its good performance

[1] https://github.com/huggingface/pytorch-pretrained-BERT

on the small data set to the deep regularization and the advanced training procedure. However, the SVM classifier performed even better, probably thanks to its overfitting resistance (Xu et al., 2009).

Additionally, we tested several approaches, including the usual high performers, such as BERT or ELMO, and the ensemble systems. None of them was able to outperform our primary submission. We attribute such behaviour to data over-fitting and lack of ability to extract higher-level dependencies from the provided samples.

Some influence on the results could have been exerted by the significantly differing distributions of post categories among the train, dev and test sets. For example, while more than 30% of all questions from the test set belonged to the 'Visas and permits' category, only 8% from the train set and 5% from the dev set fall into the same category.

Linear SVM with the USE embeddings reached an accuracy of 0.84 on the dev set and 0.86 on the test set. Surprisingly, with a different set of parameters, we achieved 0.87 accuracy on the test set, and only 0.81 accuracy on the dev set.

References

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

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

Zhiheng Huang, Marcus Thint, and Zengchang Qin. 2008. Question classification using head words and their hypernyms. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 927–936. Association for Computational Linguistics.

Donald Metzler and W Bruce Croft. 2005. Analysis of statistical question classification for fact-based questions. *Information Retrieval*, 8(3):481–504.

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, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics.

Joao Silva, Luísa Coheur, Ana Cristina Mendes, and Andreas Wichert. 2011. From symbolic to subsymbolic information in question classification. *Artificial Intelligence Review*, 35(2):137–154.

Noriko Tomuro. 2004. Question terminology and representation for question type classification. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, 10(1):153–168.

Ellen M Voorhees and Dawn M Tice. 1999. The trec-8 question answering track evaluation. In *TREC*, volume 1999, page 82. Citeseer.

Huan Xu, Constantine Caramanis, and Shie Mannor. 2009. Robustness and regularization of support vector machines. *Journal of Machine Learning Research*, 10(Jul):1485–1510.

Dell Zhang and Wee Sun Lee. 2003. Question classification using support vector machines. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 26–32. ACM.