NL-FIIT at SemEval-2019 Task 9: Neural Model Ensemble for Suggestion Mining

Samuel Pecar, Marian Simko, Maria Bielikova
Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovicova 2, 842 16 Bratislava, Slovakia
{samuel.pecar, marian.simko, maria.bielikova}@stuba.sk

Abstract

In this paper, we present neural model architecture submitted to the SemEval-2019 Task 9 competition: "Suggestion Mining from Online Reviews and Forums". We participated in both subtasks for domain specific and also cross-domain suggestion mining. We proposed a recurrent neural network architecture that employs Bi-LSTM layers and also self-attention mechanism. Our architecture tries to encode words via word representations using ELMo and ensembles multiple models to achieve better results. We performed experiments with different setups of our proposed model involving weighting of prediction classes for loss function. Our best model achieved in official test evaluation score of 0.6816 for subtask A and 0.6850 for subtask B. In official results, we achieved 12th and 10th place in subtasks A and B, respectively.

1 Introduction

Review-based portals and online forums contain plethora of user-generated text. We can consider customer reviews and inputs from online forums as an important source of novel information. These texts often contain many different opinions, which are the subject of research in area of opinion mining.

On the other hand, there can be also different types of information within these texts, such as suggestions. Unlike opinions, suggestions can appear in different parts of text and also appear more sparsely. Suggestion mining, as defined in this task, can be realized as standard text classification. We perform classification to two classes, which are suggestion and non-suggestion.

As presented by organizers, suggestion mining has different challenges (Negi et al., 2019):

- Class imbalance - suggestions appear very sparsely in reviews and forums and most of the samples are negatively sampled,
- Figurative expressions - expression can be often found in social networks but it is not always in form of suggestion,
- Context dependency - some sentences can be viewed as a suggestion, if it appears in specific domain or surrounded by specific sentences,
- Long and complex sentences - suggestions can be expressed as only small part of original sentence, which can be much longer.

Unlike opinions, suggestions can be more likely extracted also by pattern matching. We can extract suggestions by different heuristic features and keywords, such as *suggest, recommend, advise* (Negi and Buitelaar, 2015). Some works deal with domain terminology, thesaurus, linguistic parser and extraction rules (Brun and Hagege, 2013). Linguistic rules were also used for identification and extraction in sentiment expression (Viswanathan et al., 2011).

We believe that different extracted information from customer reviews and online forums can offer a valuable input for both customers and owners of products or forums and this information can be also a subject for automatic opinion summarization (Pecar, 2018).

In this paper, we present a neural network architecture consisting of different types of layers, such as embedding, recurrent, transformer or self-attention layer. We continued in our previous work on multi-level pre-processing (Pecar et al., 2018). We performed experiments with different word representations, such as ELMo, BERT or GloVe. We report results of our experiments along with error analysis of our models.

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 1218–1223
Minneapolis, Minnesota, USA, June 6–7, 2019.

2 Model

In this SemEval task, we experimented with multiple setups based on different types of neural layers on the top of an embedding layer. In Figure 1, we show general architecture of our proposed model. We also experimented with a transformer encoder, which is described in the paragraph on encoder layer below.

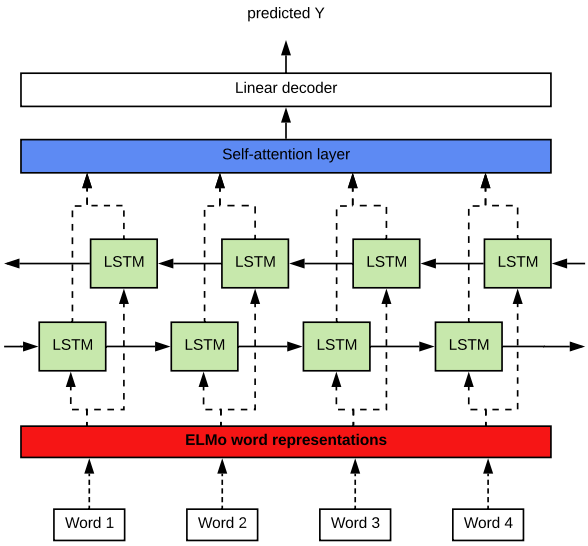

Figure 1: Proposed neural model architecture

Preprocessing We consider preprocessing of input samples as one of the most important phases in natural language processing. For user generated content, preprocessing is even more important due to noisy and ungrammatical text. We performed a study on the impact of preprocessing in our previous work (Pecar et al., 2018).

For this suggestion mining task, we used preprocessing in several stages, which were performed in order as follows:

1. text cleaning – removing all characters from not Latin alphabet, such as Cyrillic, Greek or Chinese characters,

2. character and word normalization – normalization of different use of characters and words, such as apostrophe, punctuation, date and time (e.g. ‘’ for apostrophe and ”“„ for quotation),

3. shorten phrase expanding – expanding all shorten phrases to their appropriate long form (e.g. *I'll* to *I will*),

4. expanding negations – expanding all negation forms, which appeared in short form to their appropriate long form (e.g. *aren't, won't* to *are not, will not*),

5. punctuation escaping – escaping all punctuation with spaces do separate those characters from words.

Word Representations To represent words from samples, we used deep contextualized word representations (Peters et al., 2018) also known as ELMo along with its available pre-trained model[1]. We also experimented with transformers model word representations known as BERT (Devlin et al., 2018) and its pre-trained model[2]. For language modeling in subtask B, we also experimented with GloVe embeddings (Pennington et al., 2014).

Encoder layer Word representations are fed into different encoder layers. Mostly, we used different setups of Bi-LSTM. We experimented with multiple stacked recurrent layers with different number of units within layers. In both cases (only one layer, multiple stacked layers) we used also self-attention mechanism to improve results and reduce over-fitting to the train dataset. We also tried to experiment with different attention layers and used transformer encoder (Vaswani et al., 2017) but due to very high requirements for memory, we were not able to run model with full size of this network and smaller networks produced significantly worse results.

Decoder Layer We used standard linear layer to decode output representation of recurrent layers with self-attention mechanism to class probabilities. In case of model ensemble, we needed to employ also another logarithmic softmax function for better interpretations of probabilities of samples for both classes.

Loss Function For a loss function, we experimented with the standard cross-entropy loss and the negative log likelihood loss in case a logarithmic softmax were used. We also experimented with weight setup for classes for loss contribution.

Model Ensemble Model ensemble can be considered as a useful technique to obtain better results than using only single model for predictions.

[1] https://allennlp.org/elmo
[2] https://github.com/google-research/bert

We experimented with different size of model ensemble and also two different types. In one model, we tried averaging model prediction probabilities and in second one, we used voting mechanism and predicted class with more votes.

Regularization To reduce possibilities of overfitting to train dataset, we used also dropout as a regularization technique. We used dropout on embedding layer output, on encoder output along with dropout between stacked RNN layers and also at the output of attention layer. We used different dropout probabilities in range from 0.2 to 0.6.

3 Evaluation

In this section, we briefly summarize basic information about used dataset. Later, we describe different setups of our model. Each team could submit in total 4 submission as an official results. For evaluation, binary F1 measure (F1 score over positive labels) was taken as an official results of submission.

3.1 Dataset

The dataset for suggestion mining task consists of feedback posts on Universal Windows Platform available on *uservoice.com*. The dataset contains only labels for two categories: the text is suggestion or it is not. The train dataset contains approximately 9 thousands of text samples. For validation, there were available approximately 600 samples for subtask A and 800 samples for subtask B. The size of test datasets were approximately 800 and 1000 samples for subtask A and B, respectively. More detailed information can be found in the main paper of the task (Negi et al., 2019).

3.2 Results

In Table 1, we provide basic information about setups of performed submission. For every setup, we used ELMo word representations as an embedding layer, different setups of dropout in each layer of neural network in the range from 0.3 to 0.6. Each LSTM layer has its hidden size set to 1024 units per layer. For some submissions, we also experimented with model ensemble. We took different number of models, which had the best performance on development (trial) set and used averaging predicted probabilities to get final prediction or voting mechanism and get label with more votes. In subtask A, we used 5 best trained models for voting model ensemble and 3 models for mean model ensemble. In subtask B, we used 3 best trained models for model ensemble.

In Table 1, we show also results of submitted all models in 3 measures, micro F1, macro F1 and binary F1 (F1 score over positive samples). As an official results, binary F1 measure was taken. From these results, we can observe that model ensemble can significantly help obtain better results for both subtasks.

3.3 Model ensemble results

In this section, we discuss results of each model from model ensemble in detail for both subtasks.

Table 2 shows results of each model used for model ensemble for subtask A. We can observe that the best model obtained binary F1 score 0.6609 and both types of model ensembles get better results up to 2 percents than each model separately. For mean model ensemble first 3 models were used and for voting ensemble all 5 models were used.

Table 3 shows results of each model used for model ensemble for subtask B. We can see that the best model obtained binary F1 score 0.6770 and both types of model obtained better results than each model separately. For both types of model ensemble all 3 models were used. Results for voting ensemble were not part of the official submissions.

3.4 Error analysis

We provide also error analysis of proposed model for both subtasks. We made 3 official submissions for subtask A and 2 for subtask B.

In Table 4, we show simple results from confusion matrix for subtask A and also for subtask B. For subtask A, we can observe that the main problem of our proposed models was high number of false positive labels and our models predicted presence of suggestion too often. In subtask B, there is more problematic prediction of non-suggestion labels, where number of false negative samples is much bigger. This problem can be caused also due to different distributions in training and test datasets. We also used for training dataset from a different domain, which even highlighted this problem. We used the same class weight modification for loss in subtask B as was used in subtask A.

task	submission	layers	model ensemble	micro F1	macro F1	binary F1
A	1	2 Bi-LSTM	Mean	0.9147	0.8162	0.6816
	2	2 Bi-LSTM	Voting	0.9116	0.8091	0.6696
	3	2 Bi-LSTM	None	0.9051	0.8029	0.6609
B	1	1 LSTM	Mean	0.7779	0.7567	0.6850
	2	1 LSTM	None	0.7463	0.7306	0.6656

Table 1: Official submission results in different measures

model	micro F1	macro F1	binary F1
1	0.9051	0.8029	0.6609
2	0.9050	0.8000	0.6550
3	0.9075	0.8021	0.6577
4	0.9099	0.8013	0.6543
5	0.9159	0.8061	0.6601
mean	0.9147	0.8162	0.6816
voting	0.9116	0.8091	0.6696

Table 2: Results of unsubmitted models in different measures for subtask A

model	micro F1	macro F1	binary F1
1	0.7742	0.7517	0.6770
2	0.7730	0.7446	0.6593
3	0.7463	0.7306	0.6656
mean	0.7779	0.7567	0.6850
voting	0.7574	0.7803	0.6830

Table 3: Results of unsubmitted models in different measures for subtask B

3.5 Unsubmitted models

In this section, we present results of models, which were not used to make an official submission. We experimented with these models for subtask A and also subtask B. Results can be found in Table 5 and 6. Each model in this section is used without model ensemble and we can compare results with the best models themselves. In each table, best model indicates best submitted model without any model ensemble.

The only modification used for model 1 is replacing ELMo word representation with BERT. Obtained word representations from pre-trained BERT performed much worse than ELMo representation. This fact was also observed while evaluating on development (trial) dataset.

Model 2 had a more significant modification, where LSTM encoder was replaced with transformer network (Vaswani et al., 2017). Due to high memory requirements of this model we were not able to run full encoder of the original network and used only smaller part with 6 layers and 4 head attention layers.

task	submission	TP	FP	FN	TN
A	1	76	60	11	686
	2	75	62	12	684
	3	77	69	10	677
B	1	199	34	149	442
	2	208	69	140	407

Table 4: Error analysis for submissions

As we observed in error analysis of submitted results (see Table 4), one of the significant problems was predicting too many positive labels. For all submissions, we used re-balancing class weights for loss function based on distribution in train dataset (0.6625, 2.0384). In model 3, we changed class weights to be more balanced (1.0 and 2.0). In model 4, we used completely balanced weights (1.0 and 1.0) and in model 5, we tried to change class weights to prefer negative labels (2.0 and 1.0).

model	micro F1	macro F1	binary F1
best	0.9051	0.8029	0.6609
1	0.8415	0.7214	0.5384
2	0.9123	0.7952	0.6404
3	0.9314	0.8440	0.7272
4	0.9459	0.8577	0.7457
5	0.9495	0.8479	0.7236

Table 5: Results of unsubmitted models in different measures for subtask A

For subtask B, we also experimented with use of pre-trained weights from language model. We trained language model on dataset of hotel reviews – arguana (Wachsmuth et al., 2014). Unfortunately, without fine-tuning on in-domain dataset

used for classification, this model did not obtained better results. We show results of this experiment as model 1 in Table 6. We suppose that further experiment would be needed with combination of class re-balancing for loss and also fixing pre-trained weights. Since our language model was trained with GloVe embeddings, we had to use GloVe also in training for this task. Models 2 and 3 show results with change of class weight for loss function to prefer positive labels (1.0, 4.0 and 1.0, 5.0).

model	micro F1	macro F1	binary F1
1	0.7208	0.6698	0.5401
2	0.7961	0.7844	0.7341
3	0.8264	0.8186	0.7810

Table 6: Results of unsubmitted models in different measures for subtask B

As we showed in this section, our further experiments along with error analysis showed also significant improvement in comparison to performed official submissions. We believe further work can provide even better results, especially in combination with model ensemble.

4 Conclusions

We proposed a neural model architecture for suggestion mining. For subtask A, we employed bidirectional LSTM encoder, which consisted from 2 stacked layers followed by self attention. For subtask B, better performance proved only one layer in one direction to reduce learning process and over-fitting to train domain. Our experiments showed that pre-trained ELMo word representations performed much better than pre-trained BERT. We also performed other experiments with different setups of our architecture, which were not submitted as official results. As we showed, model ensemble can significantly improve results compared when using only single models.

To obtain better results, we would need to employ transfer learning to a much bigger extent, especially for subtask B. We could also consider further experiments with re-balancing of class weights for loss function, as we predicted too many suggestions, especially for subtask A. Another possible experiments would employ transformer network, which we were not able to fully employ due to high resource requirements. Interesting would be also employing some pattern approaches, which proved as very successful for subtask B in baseline provided by organizers. Code for our submission can be found in GitHub repository[3].

Acknowledgments

This work was partially supported by the Slovak Research and Development Agency under the contracts No. APVV-17-0267 and No. APVV SK-IL-RD-18-0004, and by the Scientific Grant Agency of the Slovak Republic grants No. VG 1/0725/19 and No. VG 1/0667/18. The authors would like to thank for financial contribution from the STU Grant scheme for Support of Young Researchers.

References

Caroline Brun and Caroline Hagege. 2013. Suggestion mining: Detecting suggestions for improvement in users' comments. *Research in Computing Science*, 70(79.7179):5379–62.

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*.

Sapna Negi and Paul Buitelaar. 2015. Towards the extraction of customer-to-customer suggestions from reviews. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2159–2167. Association for Computational Linguistics.

Sapna Negi, Tobias Daudert, and Paul Buitelaar. 2019. Semeval-2019 task 9: Suggestion mining from online reviews and forums. In *Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019)*.

Samuel Pecar. 2018. Towards opinion summarization of customer reviews. In *Proceedings of ACL 2018, Student Research Workshop*, pages 1–8. Association for Computational Linguistics.

Samuel Pecar, Michal Farkaš, Marian Simko, Peter Lacko, and Maria Bielikova. 2018. NL-FIIT at IEST-2018: Emotion recognition utilizing neural networks and multi-level preprocessing. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 217–223. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language*

[3]https://github.com/SamuelPecar/NL-FIIT-SemEval19-Task9

Processing (EMNLP), pages 1532–1543. Association for Computational Linguistics.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Amar Viswanathan, Prasanna Venkatesh, Bintu G Vasudevan, Rajesh Balakrishnan, and Lokendra Shastri. 2011. Suggestion mining from customer reviews. In *AMCIS*.

Henning Wachsmuth, Martin Trenkmann, Benno Stein, Gregor Engels, and Tsvetomira Palakarska. 2014. A review corpus for argumentation analysis. In *Proceedings of the 15th International Conference on Intelligent Text Processing and Computational Linguistics*, pages 115–127, Berlin Heidelberg New York. Springer.