The 3rd International Conference on Interdisciplinary research on Computer science, Psychology, and Education (ICICPE' 2019)
December 17-19, 2019. Phu Quoc, Vietnam

Achieving Super-Human Performance in Politeness Classification

Chanhee Lee[1], Chaeeun Lee[1], Minjeong Kim[1], Heuiseok Lim[1]

[1] Department of Computer Science and Engineering, College of Informatics,Korea University, Korea
{chanhee0222, mj169, limhseok}@korea.ac.kr

Abstract. Danescu-Niculescu-Mizil et al. (2013) proposed a computational framework for identifying and characterizing linguistic aspects of politeness. They construct a politeness classifier that achieves near human-level accuracy across domains. In this paper, we extend Danescu et al.'s study by adopting deep learning model in creating new politeness classifier. Our classifier is based on neural models such as Bi-LSTM and BERT in order to capture contextual information, which is a crucial determinant for the degree of politeness. The accuracy of our classifier is compared with the human reference. The results of this study indicate that our models perform better when they are evaluated on the Wikipedia domain, while they perform worse on the Stack Exchange domain. In particular, the BERT-based model was able to surpass human performance in the Wikipedia domain.

Keywords: Politeness, deep learning, BERT, text classification

1 Introduction

Politeness has been one of the topics extensively explored in linguistics literature, being one of the central forces shaping natural language. In their seminal work on politeness, Brown and Levinson (1978, 1987) mention that it is *the most remarkable phenomenon* in human language. What makes this linguistic phenomenon intriguing is both deviance and convergence it shows: While politeness seems arguably *deviant* from the usual economy-oriented language usage, a plethora of languages across the world still *converge* showing various means for encoding politeness.

Danescu et al. (2013) have recently shed a new light on politeness studies by adopting a novel computational framework. They have created the largest corpus of requests with politeness annotations based on the data from Wikipedia (4,353 requests) and Stack Exchange (6,604 requests) and constructed a politeness classifier using a machine learning model. They compare two classifiers - a bag of words classifier (BOW) and a linguistically informed classifier (Ling.) and use human performance on the same task as a reference point, further reporting their classifiers to be effective across domains.

In this paper, we aim to extend Danescu et al.'s (2013) study by leveraging a state-of-

the-art deep learning model in constructing a politeness classifier. While their classifiers are reported to achieve human level accuracy, we expect to see more improvement in classification performance with different types of neural models that are designed to cope with longer sequences. More specifically, politeness is known to be sensitive to morphosyntactic context in which it occurs. For instance, the politeness of *please* varies depending on its syntactic position and other politeness markers it co-occurs with (Danescu et al., 2013).

Danescu et al.'s (2013) classifiers, however, are both based on SVMs using either a unigram feature representation or certain politeness related features. Their models might not capture the aforementioned broader contexts that are necessary to understand the nature of politeness phenomena. This motivates us to modify their models in a way that better captures contextual information surrounding polite expressions. To address this, the current work has focused on introducing and comparing different types of politeness classifiers based on neural models such as Bi-LSTM (Schuster and Paliwal, 1997) and BERT (Devlin et al., 2018). Our results indicate that despite the excellent performance our models showed on the Wikipedia domain, there still exist areas for improvement.

2 Method

2.1 Sequence Encoding with Bidirectional RNN

In sequence classification tasks, such as sentiment analysis or intent classification, both future and past input tokens are available to the model. Bidirectional RNNs (Graves and Schmidhuber, 2005) can efficiently make use of future and past features over a certain time frame. We use Long Short-Term Memory (LSTM) for our RNN cell, which is better at capturing long-term dependencies than vanilla RNN. Output the forward and backward RNN layers are summed to form the feature vector of each time-step. Then, we perform a max-over-time pooling over the entire sequence to acquire the vector representation of this sequence. Each sentence is classified based on this feature vector, using a softmax layer.

To capture a more abstract and higher-level representation in different layers, a densely connected layer can be added before and after the Bi-LSTM layers. The input to this network at each time-step is the concatenation of the character-level feature vector and a pre-trained word vector.

2.2 Sentence Classification with BERT

BERT, which stands for Bidirectional Encoder Representations from Transformers, is a powerful pre-trained model capable of achieving state-of-the-art performance on multiple natural language processing tasks. The key benefit of this model is its ability

to utilize large amount of unlabeled text. It is shown that BERT can be very effective in resource-scarce settings.

Since the introduction of deep learning, the amount of data required to train a model has increased tremendously. This makes the available politeness-annotated data (Danescu et al., 2013) possibly insufficient to train a bidirectional LSTM model. Therefore, we evaluate how the BERT model performs under this setting.

3 Training Details

Pre-trained Word Embeddings Utilizing word embeddings pre-trained on large unlabeled text has shown to be one of the most effective ways to increase performance on various NLP tasks. Our model uses the GloVe (Pennington et al., 2014) 300-dimensional vectors trained on the Common Crawl corpus with 42B tokens as word level features, as this resulted in the best performance in preliminary experiments. Words that do not appear in the training data are replaced with a special Out-of-Vocabulary (OOV) token. To train the vector of this token, we randomly swap words with OOV tokens while training with a 0.01 probability, as in Lample et al. (2016). The word vector is then concatenated with the character-level feature vector and fed into the subsequent layer.

Dynamic Batch Size Keskar et al. (2016) showed that small batch sizes lead to more global and flat minimizers, while large batch sizes lead to more local and sharp minimizers. Therefore, starting from a small batch size and increasing it during training would result in a more global, but sharp minimizer. While having similar effect to learning rate decay, this strategy also has a benefit of accelerating the training as the batch size grows (Smith et al., 2017). Adopting this method, we start from a fixed initial batch size, and increase the batch size by a factor of two on each quarter of the course of training.

Parameter Optimization Our network is trained by minimizing the cross-entropy loss over the tags for the softmax model and maximizing the log-likelihood of the tag sequence for the CRF model. The objective function is optimized using the gradient-based optimization algorithm Adam (Kingma and Ba, 2014). For all experiments, we implement the model using the TensorFlow (Abadi et al., 2016) library.

4 Experimental Setup

We use the same task setup presented in the work of Danescu et al's (2013). The result is a binary classification task (polite or impolite), with 2,178 and 3,302 annotated sentences in Wikipedia and Stack Exchange domain, resprectively. The dataset is designed to have balanced classes.

Since the original work did not make use of any other data than these labeled sentences, we train a bi-LSTM model from scratch, for comparison's sake. To evaluate the effectiveness of pre-trained word vectors, we also train a variant of this model where the word embeddings are initialized from the GloVe vectors as

Table 1 Comparison with the work of Danescu et al., (2013).

		In-domain		Cross-domain	
	Train	Wiki	SE	Wiki	SE
	Test	Wiki	SE	SE	Wiki
Danescu-Niculescu-Mizil, Cristian, et al. (2013)	BOW	79.84	74.47	64.23	72.17
	Ling.	83.79	**78.19**	**67.53**	75.43
	Human	86.72	80.89	80.89	86.72
Ours	Char-Bi-LSTM w\o GloVe	79.26	59.70	60.51	64.19
	Char-Bi-LSTM w\ GloVe	85.71	64.24	63.29	69.10
	BERT	**91.71**	68.79	66.78	**80.49**

described in the previous section. Finally, we fine-tune BERT on this data to test how one of the most effective NLP models performs on this task.

5 Result and Discussion

The experimental results are summarized in Table 1. Overall, our models performed the best when they were evaluated on the Wikipedia domain. Notably, BERT was able to surpass human performance by almost 5% when it was both trained and evaluated in this domain. Even when the training domain was Stack Exchange, BERT outperformed the previous work's best classifier (Ling.) by over 5%.

When evaluated on the Stack Exchange domain, on the other hand, our model performed much worse than both Ling. and BOW. On a qualitative evaluation on the dataset, we found that the sentences from the Stack Exchange domain tend to be less formal than the ones from the Wikipedia domain. Since most unlabeled text used to pre-train models such as GloVe or BERT comes from formal media such as Wikipedia or news data, this discrepancy could be the cause of the inferior performance. Further pre-training GloVe or BERT in the target domain (i.e. Stack Exchange) could lead to better performance, but we leave this to a future work.

The LSTM model without GloVe, where no pre-trained knowledge is transferred to this task, has shown the worst performance. Since the amount of training data is too small, this model experienced heavy overfitting despite the various normalization techniques. When GloVe vectors are used, we achieve increased performance in all settings.

The findings of these experiments can be summarized as follows. First, it is always beneficial to utilize unsupervised pre-training methods when possible. Second, current pre-trained models are less effective when the pre-training domain is different from the target domain. And last but not least, classic machine learning algorithms such as SVMs can perform better than the most advanced deep models when there are not enough data in the target domain.

6 Conclusion

This paper sought to take a closer look at the performance of the classifier of Danescu et al.'s study and attempted to build an improved politeness classifier. Though Danescu et al. (2013) have reported that their classifier achieved near-human performance, there was potential for improvement on capturing cumulative effect of politeness features that is hard to be understood by machine learning models such as SVMs. To investigate this idea, we applied two advanced deep models, Bi-LSTM and BERT. These models can take into consideration the interaction between politeness marking and the morphosyntactic context.

As illustrated in Table 1, the overall results indicate that our models performed better when they were evaluated on the Wikipedia domain. Though the results showed the low performance on certain domain, our study implies the benefit of utilizing unsupervised pre-training methods, the effect of congruency between training domain and target domain, and the need for choosing appropriate models in relation to the amount of data set.

Acknowledgement

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIP) (No.NRF-2017M3C4A7068189).

References

1. Cristian Danescu-Niculescu-Mizil, Moriz Sudhof, Dan Jurafsky, Jure Leskovec, and Christopher Totts. 2013. A computational approach to politeness with application to social factors. *arXiv preprint arXiv:1306.6078.*
2. Penelope Brown and Stephen C. Levinson. 1978. Universals in language use: Politeness phenomena. In Esther N. Goody, editor, *Questions and Politeness: Strategies in Social Interaction*, pages 56–311, Cambridge. Cambridge University Press.
3. Penelope Brown and Stephen C Levinson. 1987. *Politeness: some universals in language usage*. Cambridge University Press.
4. Schuster, Mike, and Kuldip K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing 45.11, 2673-2681.*
5. Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805.*