RoBERTa Is All You Need

Ziyin Zhang[*] and **Sizhe Zhou**[†] and **Xin Xin**[‡] and **Yuqian Li**[§]
519021910348 519021910587 520202910010 520021910501
{daenerystargaryen, sizhezhou, ceciliaxin, cloude_sean}@sjtu.edu.cn

Abstract

In this work, we conduct experiments of sentiment analysis on 1.6 million training samples using classifiers both statistical and deep. Our best model, fine-tuned RoBERTa-large, achieves an accuracy score of 88.58, followed closely by its multilingual counterpart XLM-R with 88.30. Prompt-tuning RoBERTa on only a fraction of the training data yields a surprising result of 86.35, and support vector machine also demonstrates a decent performance when coupled with features extracted by RoBERTa's tokenizer and embedding layer. Our code is published at `https://github.com/Geralt-Targaryen/CS247-sentiment-analysis`.

1 Introduction

Sentiment analysis is a classical task in artificial intelligence, and is generally considered to be an important benchmark in natural language understanding (Wang et al., 2018). A prevailing formulation of sentiment analysis is the quintuple definition, where an opinion is defined as a quintuple $(e_i, a_{ij}, oo_{ijkl}, h_k, t_l)$ with entity e, its aspect a, opinion orientation oo, opinion holder h, and time of opinion expression t. The opinion orientation can be either positive/negative/neutral, or expressed with different intensity levels. In this work, we make simplifications to the quintuple definition, and only consider the two-tuple (a_{ij}, oo_{ij}).

Historically, many popular supervised classification models, including Naive Bayes, SVM, logistic regression, have been adopted for sentiment analysis. Most of these statistical classifiers assume bag-of-word model, and represent each sentence with an unordered combination of each token's feature. Tan et al. (2011) also took social relationships behind user-level sentiments into consideration.

More recently, deep neural networks have been applied to sentiment analysis and achieved superior performance. Socher et al. (2013) introduced a semantic treebank with fine-grained sentiment labels for every phrase in a sentence, while Kim (2014) groundbreakingly employed convolutional neural network in text classification. And since the advent of BERT (Devlin et al., 2019), pre-trained language models based on Transformer (Vaswani et al., 2017) have established dominance in sentiment analysis, along with other natural language understanding benchmarks (Liu et al., 2019; Conneau et al., 2020).

2 Traditional Classifiers

Since the rise of deep learning, applying statistical models to features extracted by neural networks has been a popular approach, especially in the field of computer vision after the advent of ResNet (Chopra et al., 2013; Hoffman et al., 2014). Similarly, in this work we explore the capabilities of classical models such as SVM with features extracted by various networks.

2.1 SVM

Support Vector Machine was arguably the most popular and powerful model in machine learning before the rise of deep neural networks. Even today, researchers still resort to SVM on tasks where data is really scarce. However, as SVM performs classification by maximizing the margins between different data classes within the feature space, it requires each data sample to be represented as a fixed-dimension feature vector, which is not intuitive for sequence classification tasks such as sentiment analysis.

To address this issue, we adopt two approaches to generate a fixed-dimension representation for

[*] Department of Computer Science and Engineering, SEIEE
[†] Department of Electrical and Computer Engineering, UM-SJTU Joint Institute
[‡] School of Media & Communication, SJTU
[§] School of Mechanical Engineering, SJTU

each input sentence. The first is to naively tokenize the raw sentence by the occurrence of white spaces, and average the 1024-dimension word2vec (Mikolov et al., 2013a,b) representation of each token in the sentence. The word vectors are learned from the training corpus, and any unknown tokens in the test samples are ignored.

The second is to tokenize the raw inputs with RoBERTa's tokenizer instead, and process the tokens with RoBERTa's embedding layer to obtain one 1024-dimensional feature vector for each token. The feature vector for the input sentence is then computed by simply taking the average of each token's feature in that sentence. However, we note that this approach differs from the first one in more than one aspect: on one hand, RoBERTa's tokenizer is based on BPE (Sennrich et al., 2016) and can process any token in the test vocabulary; on the other hand, RoBERTa's embeddings are pretrained on a much larger corpus than word2vec. Thirdly, the sentence representation generated by averaging word2vec vectors is strictly a bag-of-word model and contains no information of word ordering in the sentence, while RoBERTa's embedding vectors are the summation of word embedding, token type embedding, and positional embedding. Also, we postulate that since the dimension of RoBERTa embeddings is 1024 - much larger than the average length of Twitter comments (Figure 4) - most of the information contained in token embeddings would be preserved even after being averaged over the dimension of sentence length.

2.2 Training Details

We randomly sample 10 thousand instances from the original dataset, and preserve 1/10 of this smaller training set for validation to determine the best kernel and regularization strength C. We choose this smaller training set for SVM both because SVM has limited expression power in face of such a large amount of data, and because the optimization procedure of SVM is much more complicated then the simple forward pass and gradient descent used in neural networks, and it may take quite a long time to fit an SVM on even only several tens of thousands of training samples. In section 2.3, we also briefly explore the impact of the size of training set on SVM's performance.

2.3 Results

The results of sentiment classification using SVM are recorded in Table 1. Unsurprisingly, using features extracted by RoBERTa's embedding layer yields much better results than word2vec, even though they are both static and of the same dimension. Line 3, 5, 6, 7 of Table 1 show that the size of training corpus matters up to a certain point, but keep increasing the amount of data beyond that starts to confuse the model, as the best performance is achieved at 10 thousand training samples.

Embed	**Train Size**	**Acc**
word2vec	10k	68.80
word2vec	10k	62.95*
RoBERTa	10k	**83.84**
RoBERTa	10k	82.45*
RoBERTa	1k	76.60
RoBERTa	5k	81.62
RoBERTa	50k	81.62
no positional embedding:		
RoBERTa	10k	83.29
RoBERTa	10k	82.45*

Table 1: The performance of SVM classifiers. Results marked with * are obtained without sentence cleaning before tokenization.

As ablation studies, we first evaluate the contribution of sentence cleaning before the tokenization step, which normalizes the text to lower case and removes all punctuation. The results are recorded in Table 1, marked with *. It can be observed that for classifiers using RoBERTa embedding, removing sentence cleaning only leads to 1 point drop in performance, but for classifiers using word2vec embedding the drop is almost 6 points. We hypothesize that this is due to the fact that word2vec is much sensitive to non-standard utterances, such as emoticons, most of which are removed during this preprocessing step.

We also replace the positional embedding sublayer in RoBERTa's embedding module with an identity matrix, as in the last two lines of Table 1, which show that positional embedding has almost no impact on the downstream classification performance. We posit that this is because the expectation of cosine positional embeddings of tokens within a sentence is approximately 0, and this positional information is lost when being averaged, reducing the model back to the bag-of-word level.

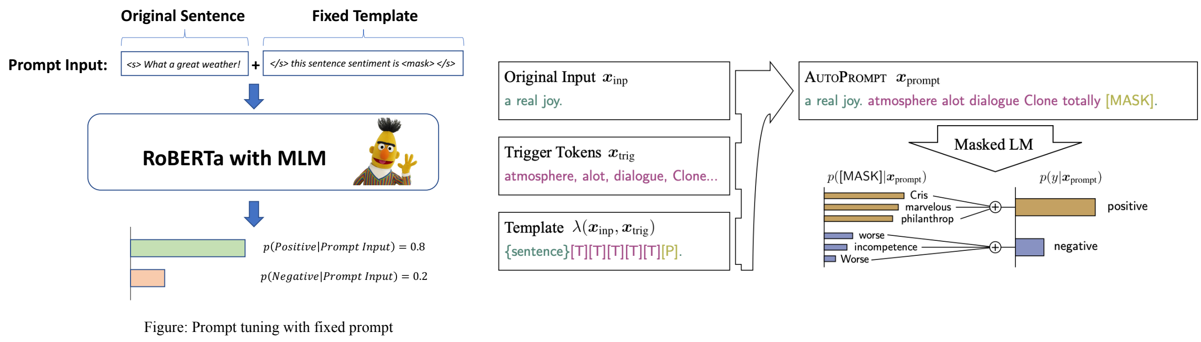

Figure 1: An illustration of fixed-prompt tuning (left) and AUTOPROMPT from (Shin et al., 2020) (right).

3 RoBERTa-based Classification

3.1 Fine-tuning

In this age, where deep learning holds sway over the vast domain of artificial intelligence, the default, simplest, and probably best-performing method for sentiment analysis on millions of training data is obviously fine-tuning a pre-trained deep model. And that is exactly what we did - fine-tuning a RoBERTa. The network structure and pre-training details of RoBERTa we omit here, but refer readers to the source works of Vaswani et al. (2017); Devlin et al. (2019); Liu et al. (2019) instead.

However, considering that the training data collected from Twitter is quite noisy and may contain many tokens that do not fall into RoBERTa's relatively small vocabulary (which has about 50 thousand tokens), we also fine-tune an XLM-R (Conneau et al., 2020), the multilingual counterpart of RoBERTa-large with a vocabulary size of 250 thousand tokens. To investigate whether unknown tokens (such as comments in non-English languages) pose a bottleneck to RoBERTa's performance, we train XLM-R with strictly the same set of hyper-parameters. However, it should be noted that recent works in the literature have found multilingual models to underperform on high-resource languages' downstream tasks compared with monolingual ones, in the case of both natural language understanding and machine translation (Conneau et al., 2020; Arivazhagan et al., 2019).

3.2 Prompt-tuning

While RoBERTa fine-tuned with task-specific supervision has achieved state-of-the-art performance on GLUE benchmark (Liu et al., 2019), this pre-training-fine-tuning framework has several drawbacks. The first is that registering a new classification head for each downstream task introduces extra parameters during the stage of fine-tuning, which must be trained from scratch on the limited task-specific data. Secondly, formulating downstream tasks as sequence classification is also inconsistent with the masked language modeling objective, preventing RoBERTa from maximally exploiting the knowledge that has been learned during pre-training. Additionally, during the fine-tuning stage the parameters in the network's self-attentions layers are often adjusted along with the newly registered classification layer, entailing that one or more checkpoints need to saved for each downstream task, which could take up a considerable amount of disk storage for RoBERTa-large. In response to these issues, prompt-tuning has been proposed as an alternate to fine-tuning.

In this work, we first address the second issue, and utilize RoBERTa's `<mask>` token to elicit a prediction. The model's architecture is shown in Figure 1. For each input sentence, we append a suffix `</s> this sentence sentiment is <mask>` before sending it into the tokenizer. During the training stage, one feed-forward layer learns to project the hidden state at the mask token's position into the label space. In this way, the number of extra parameters introduced in the downstream task is the same as fine-tuning, but the input of that extra layer now corresponds to the mask token rather than the classification token `<s>` at the beginning of the sentence. This trick exploits RoBERTa's pre-training objective in a much more data-efficient way, as RoBERTa's pre-training procedure does not include next sentence prediction (NSP) task, and the hidden representation of `<s>` is actually meaningless before fine-tuning.

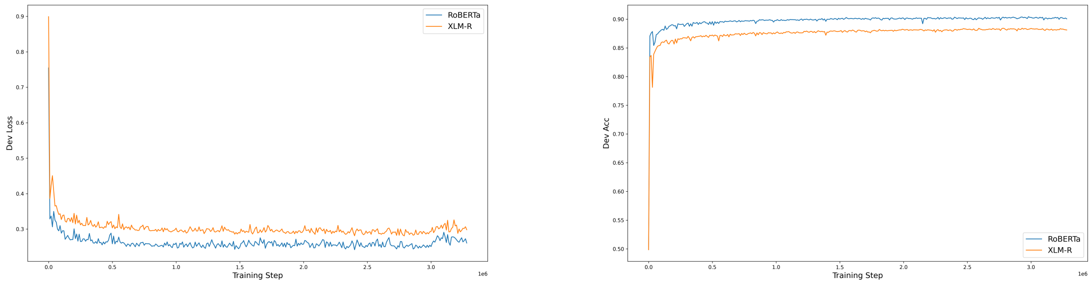

Figure 2: Loss and accuracy on the development set during fine-tuning of RoBERTa and XLM-R.

3.3 Automatic Prompt-tuning

We also take one step further, and try to automatically generate the prompting suffix using AUTOPROMPT (Shin et al., 2020). The ideas behind AUTOPROMPT is illustrated in Figure 1. Each input sentence $\mathbf{x}_{inp}$ is reformulated to $\mathbf{x}_{prompt}$ using a template $\lambda(\mathbf{x}_{inp}, \mathbf{x}_{trig})$. $\mathbf{x}_{trig}$ is a series of trigger tokens that are found using a gradient-based search, and fills in the slots marked as `[T]` in the template. `[P]` is replaced with the mask token `<mask>` upon the template's instantiation, and directly elicits an output-token distribution from the pre-trained masked language model. The probabilities of two pre-defined sets of tokens - one for positive comments and the other for negative ones - within this distribution is then marginalized to produce the final result. These two sets of tokens can be either manually specified, or automatically constructed. In the second case, a logistic classifier is first trained on top of the masked language model's last hidden layer (i.e. the layer right before output word embedding) at the position that corresponds to the `<mask>` token, to predict class labels. The top-k tokens whose word embeddings (obtained from the output layer of RoBERTa) have the highest correlation with each class are then returned as the candidate labels for that class. Once tuned, the only additional "parameters" that this model introduces to RoBERTa are the two sets of candidate labels and the set of trigger tokens, which can be conveniently applied in an off-the-shelf fashion.

3.4 Training Details

For fine-tuning both RoBERTa-large and XLM-R, we use AdamW (Loshchilov and Hutter, 2019) to tune all of the models' parameters with learning rate 5×10^{-6}, weight decay 1×10^{-2} and batch size of 8. We randomly sample 100 thousand samples from the data to serve as development set, and train the models on the rest 1.5 million samples for 3 epochs. Training each model takes about 30 hours on an RTX 3090.

When fine-tuning these large models, unlike the traditional models in section 2, we do not apply any sentence cleaning during preprocessing, the rationale being that cleaning based on regular expression can never cover every strange new word in such a large training corpus, let alone the unknown test corpus. So we might as well let these models learn to deal with bizarre utterances (including emoticons) by themselves. For example, the sentence `0ff t0 the meetin i hate when ppl v0lunteer my free timegrrr` obviously contains misspelling, shorthand, as well as OCR-induced errors, all of which occur in patterns and can be learned when there is enough data, but are almost impossible to cover with hand-written rules. Another consideration is that sentence cleaning often obliterates information that is actually helpful for sentiment analysis, especially capitalization, which is often the symbol of sarcasm in English, as in `I HATE to admit it but, I LOVE admitting things`.

For prompt-tuning RoBERTa, we use AdamW with learning rate 2×10^{-5} and a linear scheduler with 100 warmup steps. As the main idea behind prompt-tuning is to increase data efficiency, we only us 1 thousand training instances randomly sampled from the corpus, and reserve one-tenth of them as validation set. We repeat the procedure using 3 random seeds, and report their median performance on the test data. For comparison, we also repeat the fine-tuning procedure on the same amount of data.

3.5 Results

The training curves of fine-tuning the two pre-trained models are plotted in Figure 2, and it can

Model	Acc
RoBERTa$_{large}$	**88.58**
XLM-R	88.30

Table 2: The performance of pre-trained language models.

Method	Acc
SVM	76.60
fine-tune	86.35
prompt-tune	**86.91**

Table 3: Performance comparison on only 1k training data.

be observed from the development loss curves that both models start overfitting on the training set after about 3 million steps. We choose the checkpoint with the lowest development loss for each model, and test their accuracy on the test set, as recorded in Table 2. RoBERTa-large, as expected, achieves a state-of-the-art result of 88.58. What's more surprising is that XLM-R follows closely behind RoBERTa, with only a lag of less then 0.3 points. We hypothesize that this may be a result of the extremely large training set of this task, whence XLM-R may make up for its relative insufficiency of English representation after multilingual pre-training. Also, from Figure 3 it can be found that there are some quite frequently-occurring tokens in the data that are not standard English (e.g. `Â` and `Ã`). These tokens are included in RoBERTa's vocabulary (as Figure 3 is plotted based on its tokenizer), but XLM-R may nontheless be better at dealing with accented utterances.

For the scenario of small training set, we compare the performance of fine-tuned RoBERTa, prompt-tuned RoBERTa, and SVM using RoBERTa embedding (the 5th line in Table 1) in Table 3. With relatively scarce training data, prompt-tuning is slightly better than fine-tuning. Perhaps the more surprising observation is that with only less than 1/1000 training data compared with the large-scale fine-tuning results in Table 2, fine-tuning RoBERTa on this tiny sub training set leads to only two points' drop in performance. We hypothesize that this is probably due to the homogeneity of the training set and the limited size of test set, whose combined effect is that one thousand randomly sampled training sentences are quite sufficient for fine-tuning a large model. For automatic prompt-tuning, we use the implementation of (Shin et al., 2020), but do not observe any performance gain over our fixed prompts.

4 Data Analysis

In Figures 3 and 4, we visualize some basic statistics of the training data. Figure 3 shows the tree maps of tokens most frequently occurring in the positive samples that are not frequent in negative samples, or vice-versa. While some of these most frequent tokens are inherently sentimental (such as `love`, `great` for the positive class and `sad`, `hate` for the negative class), others are not that intuitive (for example, why `Â` and `Ã` occur much more frequently in the positive class remains a myth). These complications justify our choice of large language models based on Transformer architecture rather than the more interpretable and traditional models such as Naive Bayes or TF-IDF embeddings.

In Table 4 and Figure 5, we summarize some of the most common patterns appearing in models' wrong predictions and their contributions to the errors of SVM (using word2vec features) and fine-tuned RoBERTa. The largest portion of error come from inherent obscurity in the sentences' sentiment, such as those with neutral sentiment or mixed emotions. Other than these, SVM classifier has a much higher error rate on sentences where a single tokens plays an important role in determining the whole sentence's sentiment, such as sentences with negation and transition words, or words that rarely occur in the training corpus or have multiple meanings. These phenomena corroborate our hypothesis in section 3.4. However, both models seem to have trouble dealing with emoticons. This is probably due to the unique characteristics of Twitter, and could probably be addressed by including Twitter text into RoBERTa's pre-training corpus.

We also observe that many of the errors can be improved by taking linguistic features into consideration, especially part of speech, as shown in Figure 5. Words with different part of speech are inherently different in emotional density, as adverb, adjective, verb and noun are subjective in a decreasing order. We also find that when there is a emotional conflict between different words, the sentiment of the sentences is largely dependent on the ideogram. For example, when verb/noun's emotion differs from adjective, the sentiment of the sentence should go with the verb/noun.

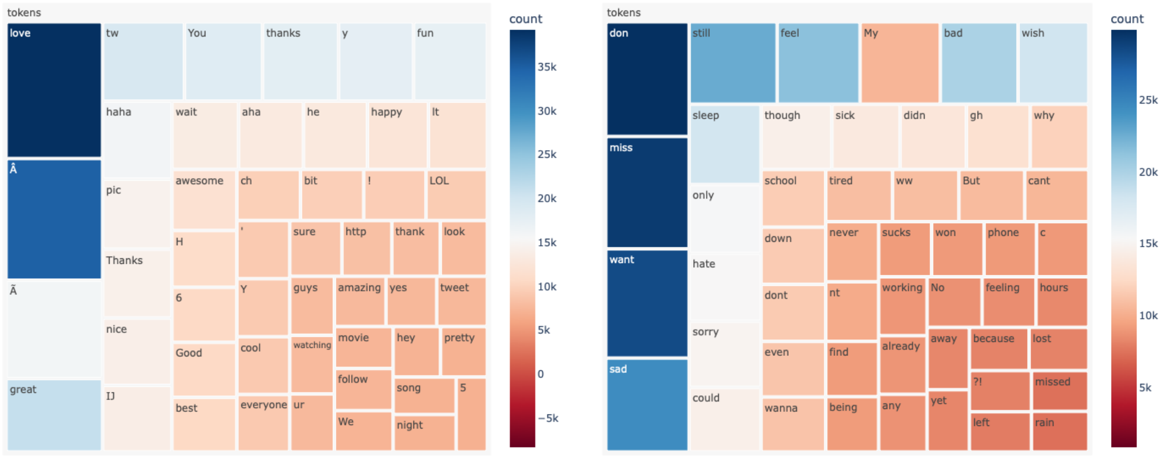

Figure 3: Tokens most contributing to the positive class (left) and negative class (right) in the training data.

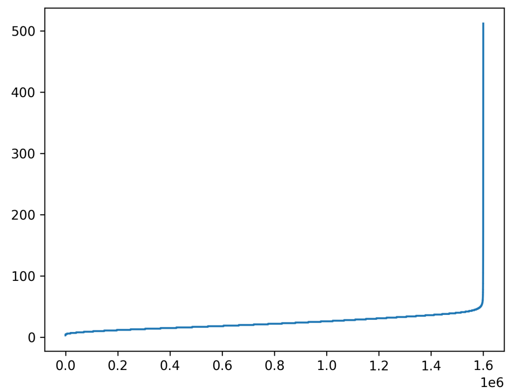

Figure 4: Sentence length distribution (tokenized by RoBERTa) of the 1.6 million training samples.

5 Conclusion

In this work, we compared the performance of SVM and RoBERTa on the task of sentiment analysis under various settings. For SVM, using BPE tokenization and pre-trained RoBERTa embeddings as input features leads to a significant gain in performance over the naive tokenization and word2vec embeddings, while for RoBERTa prompt-tuning demonstrates higher data efficiency than fine-tuning. We also analyzed in detail various linguistic phenomena that could mislead the models, and proposed corresponding solutions.

References

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George F. Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *CoRR*, abs/1907.05019.

Sumit Chopra, Suhrid Balakrishnan, and Raghuraman Gopalan. 2013. Dlid: Deep learning for domain adaptation by interpolating between domains.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8440–8451. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*,

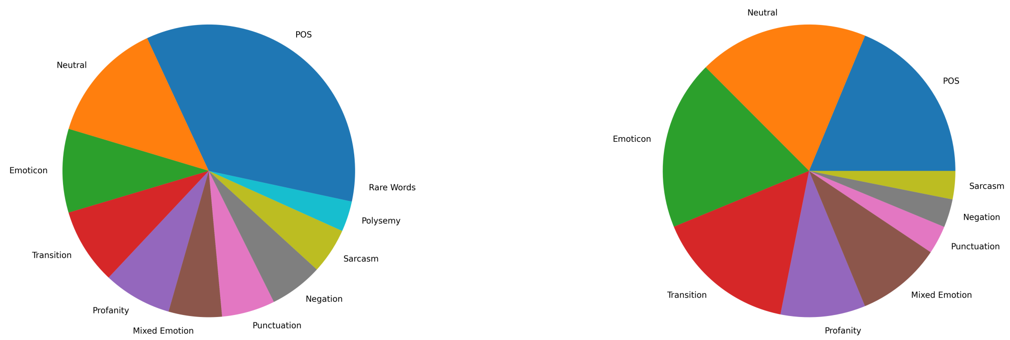

Figure 5: Statistics of error types in the predictions of SVM (left) and fine-tuned RoBERTa (right).

Type	Example	Label	Error Count	
			SVM	RoBERTa
Neutral	wanttss to go out is thinking of what to put on twitter	0 1	16/121	6/41
Rare words	Cheney and Bush are the real culprits - http://fwix.com/article/939496 @matthewcyan I finally got around to using jquery to make my bio collapse. Yay for slide animations.	0 1	4/121	0/41
Polysemy	@Lou911 Lebron is MURDERING shit. @wordwhizkid Lebron is a beast... nobody in the NBA comes even close.	1 1	4/121	3/41
Sarcasm	Time Warner Cable slogan: Where calling it a day at 2pm Happens.	0	6/121	0/41
Mixed emotion	@SoChi2 I current use the Nikon D90 and love it, but not as much as the Canon 40D/50D. I chose the D90 for the video feature. My mistake.	1	7/121	6/41
Emoticons	is going to sleep then on a bike ride:]	1	11/121	5/41
Punctuation	@ work til 6pm... lets go lakers!!!	1	7/121	3/41
Transition	@Pittstock $GM good riddance. sad though.	0	10/121	1/41
Profanities	Damn you North Korea. http://bit.ly/KtMeQ	0	9/121	1/41
Negation	GM files Bankruptcy, not a good sign...	0	7/121	1/41

Table 4: Case studies.

pages 4171–4186. Association for Computational Linguistics.

Judy Hoffman, Eric Tzeng, Jeff Donahue, Yangqing Jia, Kate Saenko, and Trevor Darrell. 2014. One-shot adaptation of supervised deep convolutional models. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751. ACL.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Name	Work	Contribution
Ziyin Zhang	fine-tune, prompt-tune, SVM, report	32%
Sizhe Zhou	prompt-tune, report, presentation	27%
Xin Xin	case study	24%
Yuqian Li	SVM	17%

Table 5: Suggested Contributions.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 3111–3119.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4222–4235. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1631–1642. ACL.

Chenhao Tan, Lillian Lee, Jie Tang, Long Jiang, Ming Zhou, and Ping Li. 2011. User-level sentiment analysis incorporating social networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011*, pages 1397–1405. ACM.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the Workshop: Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2018, Brussels, Belgium, November 1, 2018*, pages 353–355. Association for Computational Linguistics.

A Suggested Contributions

See Table 5.