Evaluation of hand-crafted features in conjunction with contextual word representations for multi-domain sentiment classification

Anish Aritakula

1 Abstract

The introduction of BERT (Devlin et al., 2019) has simplified a lot of NLP tasks. BERT pre-trained models can solve up to 11 downstream NLP tasks by adding a task specific layer on the top of the BERT embeddings. This includes sentiment classification as well. BERT achieves state of the art results on the SST-2 dataset. Since the advent of BERT, the usage of hand crafted or custom features has been very limited given the excellent performance provided by some of the pre-trained transformer models. In this paper, we explore the idea of adding custom or hand-crafted features to the existing BERT embeddings to see if there is a further improvement in performance of sentiment classification.

2 Introduction

Sentiment classification is the task of assigning sentiment labels for reviews or opinions of people. This relates to different areas such as movie reviews, product reviews, restaurant reviews etc. In some cases(example Amazon), you would have thousands or millions of reviews and it is not humanly possible to annotate or tag each review a sentiment label manually. A machine learning framework could achieve this in a fraction of time and solve the problem.

" I loved the movie"-Positive

"What a waste of time"-Negative

The initial challenge in this field was lack of solid labeled data. This made it difficult to convert this into a classification problem and remained mainly an unsupervised problem. Turney(Turney and Littman, 2002) used a technique called Pointwise Mutual Information to extract sentiment from text without using labeled data. This was an unsupervised technique. (Pang and Lee, 2008) used movie reviews dataset(labeled), which were a combination of publicly available customer reviews data + annotated datasets, with a Naïve bayes classifier. While this was a great advancement at that time, The Pang Lee model relied on a bag-of-words approach to sentiment analysis, which means that it did not take into account the context or tone of individual words or phrases. As a result, it was not able to detect sarcasm or irony. Sentences like below were difficult to classify accurately.

"Well, that's just great. I got a flat tire on the way to my job interview."

"Wow, I just love spending my entire weekend doing laundry."

In 2013, (Socher et al., 2013) introduced a deep learning architecture called the Recursive Neural Tensor Network (RNTN), which uses recursive neural networks to model the structure of sentences. They applied this model to sentiment analysis and achieved state-of-the-art results on several benchmark datasets. This paper also marked the introduction of Stanford Sentiment Treebank(SST) which was a collection of movie reviews annotated with sentiment labels at the sentence and phrase level. The dataset was created by parsing the reviews using a syntactic parser and then manually labeling each phrase and sentence with a sentiment label (positive, negative, or neutral). The SST has since become a widely used benchmark dataset for sentiment analysis, and several variants of the dataset have been created with different levels of granularity and annotation schemes.

(Tang et al., 2015) brought to the NLP field the concept of attention mechanisms gated recurrent neural network (GRU) which were able to capture the context and also the relation between sequential structure of sentences which earlier models could not capture.

Then came BERT (Devlin et al., 2019) which was pre-trained on a large corpus of text. So this mostly eliminated the need for training a new model from scratch. The pre-trained embeddings could then be combined with a task specific layer to achieve the desired outcome. BERT can be fine-tuned on specific tasks with only a small amount of labeled data, which makes it more flexible and adaptable to a range of different applications and domains. Also the fact that it was bi-directional

meant that it takes into account the entire context of a sentence, both before and after each word, when making predictions. This allowed BERT to capture more nuanced relationships between words and phrases and resulted in better performance. Other similar transformer based models which came later were RoBERTa and ELECTRA

One area less explored is if hand crafted features provide any incremental performance uplift in conjunction with contextual word representations. There has been a lot of research around use of handcrafted features in image processing and computer vision but very limited in the field of NLP. Hand crafted features are features which are custom made(not complex feature embeddings) yet could prove to be powerful in predicting the outcome. While contextual word representation models like BERT try to capture the semantic aspect of words in a sentence based on its context, there could be certain subtle nuances which may or may not be captured by the word representations. For example: The presence of multiple exclamation marks might indicate excitement or enthusiasm or even frustration. One of the hypotheses(to be proved or disproved) is that longer reviews tend to be more negative than positive. Having conjunctions like 'but', 'while' might indicate a change in sentiment or potentially a neutral polarity. Sentiment lexicons with associated sentiment scores specific to the domain(restaurant reviews or movie reviews) might have more pertinent information which the generic BERT embedding may not capture.

The hypothesis for the experiment would be that concatenating hand crafted features to the existing embedding vectors provides an improvement in performance for multi-domain sentiment classification tasks.

3 Related Work

Based on secondary research, there were not many references to usage of hand built features or custom features for sentiment classification since the advent of pre-trained models. There is however a lot of work done in the area of cross-domain sentiment analysis.

(Chi et al., 2020) propose BERT post training which is essentially taking BERT pre-trained weights as the initialization for base language understanding and then adapts BERT by self-supervised pre-trained tasks. To achieve the latter, they use a) Domain distinguish task and b) target domain masked language model The domain-distinguish task(DDT) pre-training is a classification task with an aim to identify the domain of the sentence. In post training, instead of NSP task, it is replaced by DDT task. In addition to this, adversarial training was done to eliminate the domain-specific features to derive the domain- invariant features. This is done using a domain discriminator whose purpose is to predict domain label of samples.

In the paper by (Yu et al., 2019), the authors propose a novel method of solving the text classification problem with better domain knowledge by creating an auxiliary sentence to turn the classification task into a binary sentence-pair task. The idea is that this would help tackle the limited training data problem and task-awareness problem. 4 variants are considered. 1)BERT4TC-S: This is the vanilla BERT model. 2)BERT4TC-AQ: This is the BERT model plus an auxillary sentence which is a question which does not contain label information 3)BERT4TC-AA: This is the BERT model plus an auxillary sentence which is only contains the label information 4)BERT4TC-AWA: The auxillary sentence here contains both the label and some other words. The authors mention that the auxiliary sentence would significantly improve the performance. In addition to the above, a post training mechanism using a domain corpus is defined to improve domain aware predictions. It includes MLM for infusing domain knowledge and NSP to learn contextual representations beyond word level. The idea is reduce the total loss(MLM loss + NSP loss)

4 Data

The intent is to use datasets that have sentences with ternary sentiment labels of 'positive','neutral' and 'negative'. The standard datasets that can be leveraged are

SST-3: The SST-3(Stanford Sentiment Treebank) dataset has become a standard benchmark dataset for evaluating models for sentiment analysis tasks. It has sentences and associated ternary labels. The dataset is split into train, development (dev), and test sets. The train set contains 8,544 sentences, the dev set contains 1,101 sentences and the test set contains 2,210 sentences. The dataset was created by annotating sentences extracted from movie review labels.The sentiment labels were obtained by asking annotators to rate the sentiment of the sentence on a scale from 1 (very negative) to 5 (very positive), and then mapping the ratings to a three-way sentiment label (positive, negative, or neutral). The dataset was first introduced in "The Stanford Sentiment Treebank with Recursive Deep Models" by (Socher et al., 2013). It also includes labels for the parsed subtrees for the train dataset which would be very useful for the training purposes.

Restaurant reviews: A new restaurant

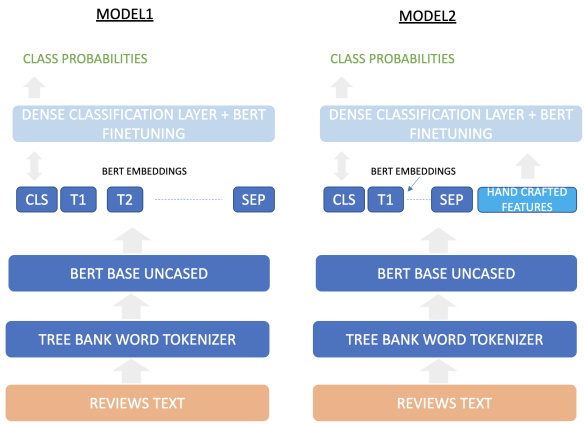

Figure 1: Models representation

dataset has been shared by Prof. Chris Potts as part of the course XS224u which has sentences/reviews with ternary labels.

Dynasent: The DynaSent dataset consists of 25,000 reviews from Amazon, Yelp and IMDB. The dataset is split into train, dev and test sets. The train set contains 20,000 reviews, the dev set contains 2,500 reviews and the test set contains 2,500 reviews. The DynaSent dataset was created by the authors of the paper "DynaSent: A Dynamic Benchmark for Sentiment Analysis" (Tang et al., 2021)

IMDB reviews: This dataset has 50K reviews. These are highly polar reviews with labels for 'positive' or 'negative' sentiments. This includes 25K reviews for training and 25K reviews for testing. Although this dataset does not contain ternary labels, this dataset could be highly useful for identifying extremely positive or extremely negative sentiments.

The idea is to a create a sentiment classification framework that works well on multiple domains. Hence the attempt to combine diverse datasets.

5 Models

Figure 1 gives a high level illustration of the models under comparison for this experiment.

BERT: For extracting the contextual word embeddings, we are using the pre-trained BERT model(BERT base uncased) as this has shown to provide good performance when combined with a downstream classification layer.(Figure 2)

Hand crafted features: The hand-crafted features that were used are

a)Length of the sentence(Number of tokens): Hypothesis is that longer reviews may

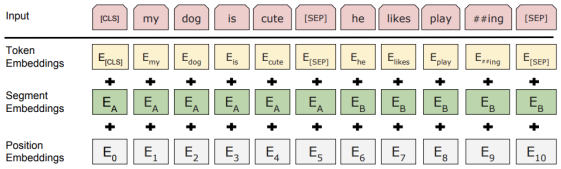

Figure 2: BERT input representations

be associated with negative emotions. This variable was normalized with respect to the maximum length so that the range of the variable is between 0 and 1. This way we are trying to avoid the model giving higher weight to the higher numeric values that this variable could potentially take otherwise.

b)VADER sentiment lexicon scores: Valence Aware Dictionary and Sentiment Reasoner((Hutto et al 2014) is a lexicon-based approach for sentiment analysis that assigns a score to a given text based on the presence of specific words or phrases that are indicative of positive, negative, or neutral sentiment. The VADER sentiment lexicon scores are computed using an algorithm that takes into account the frequency and intensity of the lexical features present in the text. The reason for adding this custom feature is that particularly for short text snippets like tweets or headlines, lexicon-based approaches can be more effective than more complex models that require large amounts of labeled training data. We use the 4 scores of neutral, negative, positive and compound from VADER.

c)Presence & count of Exclamation marks: Exclamation marks are usually associated with excitement or frustration. The count and presence of them could indicate certain sentiment polarity when used in combination with other features. The count variable was also normalized with respect to its maximum value before feeding it into the model similar to the length variable earlier.

d)Presence of Subordinate conjunctions: The neutral category is usually a difficult one to predict. Most of the time a line of text could have both positive and negative emotions in it which could then make the whole sentence neutral. We are looking for the presence of subordinate conjunctions like 'but', 'while', 'although', 'yet', 'despite', 'however' . We were seeing low f1-scores for the neutral category for SST dataset using basic models. Hence this indicator could be a strong indicator for the neutral category.

In Figure 3 we have plotted the bi-variate views of hand crafted features with respect to the target sentiment label. From the graphs, It is evident that The VADER scores of positive, negative, neutral and compound are highly correlated with the final class labels. Similarly exclamation marks have a

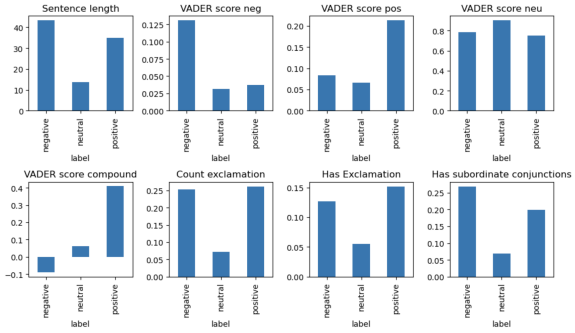

Figure 3: Hand crafted features distribution with respect to Sentiment labels(Note the low sentence length for neutral in the 1st graph is due to the inclusion of subtrees)

skew towards positive or negative labels and sentence length is skewed towards negative labels. In the experiment, we want to see if most of this is already captured by the BERT embeddings or not.

Classification layer: For classification layer on top of BERT embeddings, we used a linear layer(input dim=768+8 hand crafted features, 64 hidden dimensions) + ReLU + Dropout layer + a final linear layer for classification.

Using the initial linear layer to reduce the dimensionality to 64 dimensions. The ReLU activation layer is to tackle the vanishing gradients problem. A drop out layer with factor 0.1 was added to prevent overfitting of the model.

6 Experiment

A BERT fine tuned model plus a classification layer on top should form a solid baseline model. This was used as the baseline/champion model. The challenger or comparison model would be the same plus a few hand-crafted features. Refer to Figure 1 for a high level depiction of the two models under comparison.

We are using 4 different datasets as mentioned above 1) SST dataset 2) Restaurant reviews dataset 3) Dynasent datasets 4)IMDBdatasets

Data pre-processing

The SST datasets were tokenized using the Treebank format. While the other datasets had different formats. In order to maintain consistency of datasets, we tokenized the other 3 datasets also to reflect the same tokenization scheme. This makes it easier for the model to identify patterns while training. Also leveraged the parsed subtrees of SST train dataset which were labeled for training. The maximum length of tokens for the selected BERT model was 512. While the 1st 3 datasets had sentences of smaller length, the IMDB dataset has long reviews. So we removed sentences that were >150 words.

For training, we used the SST-3 train, Restaurant reviews train(We split the dev data into train and dev), Dynasent1 Dynasent2 datasets and the IMDB reviews dataset.For validation, we used the SST-3 dev,Restaurant reviews dev dataset.For test, we used the SST-3 test Restaurant reviews test dataset. The total number of records in train, dev and test were 252,687, 2,282 and 4,574 respectively.Table 1 gives the distribution of sentiment labels for the train dataset.

Table 1: Distribution of sentiment labels in the train dataset

Sentiment label	sentences count
Neutral	122,043
Positive	73,589
Negative	57,055

Training

Given the resulting training dataset was very large, it was not possible to perform hyperparameter tuning with the available compute resources available. Hence performed hyperparameter tuning on a small sample of observations (20% of training records randomly sampled). The parameters which were used for tuning were

Learning rate: [0.00005, 0.0001, 0.001]

Hidden dim (In the classifier layer): [64,128,200]

Gradient accumulation steps: [1,4,8]

The best performance on dev for macro f1-score was obtained with Learning rate=0.00005, Hidden dim=64 and Gradient accumulation steps=4. These parameters were then used on the 100% training data.

Model1: BERT embeddings fine tuned + classifier layer
Model2: BERT embeddings + hand crafted features finetuned + classifier layer

The results of the two models are shown in Tables 2, 3 & 4

7 Analysis

We used the macro average of f1 scores across the 3 classes as the main evaluation metric. This way we ensure that all 3 classes are given equal importance in terms of prediction. At an overall level, the restaurant dataset had consistent f1-scores across all 3 categories with both the models. In the SST-dev dataset, The f1-scores for 'positive' and 'negative' categories was quite good. However the scores

Table 2: Model1 performance

Dataset	Label	Precision	Recall	F1
SST-dev	Negative	.78	.85	.82
	Neutral	**.53**	**.23**	**.32**
	Positive	.75	.90	.82
Rest-dev	Negative	.69	.77	.73
	Neutral	.82	.69	.75
	Positive	.73	.81	.77

Table 3: Model2 performance

Dataset	Label	Precision	Recall	F1
SST-dev	Negative	.80	.78	.79
	Neutral	**.43**	**.29**	**.35**
	Positive	.74	.89	.81
Rest-dev	Negative	.75	.69	.72
	Neutral	.76	.79	.77
	Positive	.80	.79	.79

on the neutral category were very low. Upon performing error analysis, we found two potential reasons why this was happening a) Having a combination of positive and negative sentiments in the same sentence b) Unable to identify/differentiate opinion vs facts.

At first glance, The macro f1-scores for Model 2 was slightly better than Model 1. However we need to perform some sort of statistical test to ascertain if the increase was statistically significant or not. Given the time taken to run this experiment, Mcnemar's test seemed to be a good test to understand the difference in prediction accuracy of the two models.

Mcnemar's test is a statistical test used to compare the performance of two different models or algorithms on a specific task. In sentiment classification, it can be used to compare the performance of two different classifiers on the same dataset.The test is based on a contingency table Table 5 that shows the number of instances that were correctly or incorrectly classified by both classifiers.

Chi-square statistic is calculate based on equation (1). Once the chi-square statistic is computed, the p-value can be calculated using the chi-square distribution with 1 degree of freedom. If the p-value is less than 0.05, we reject the null hypothesis and conclude that the two classifiers have significantly different performance. If the p-value is greater than the chosen level of significance, we fail to reject the null hypothesis and conclude that there is not enough evidence to conclude that the two classifiers have significantly different performance.

Table 4: Summarized models performance

Model	Macro avg F1-score(Dev set)
Model1	.701
Model2	.705

Table 5: Contingency table for Mcnemar's test

		Classifier 2	
		Correct	Incorrect
Classifier 1	Correct	a	b
	Incorrect	c	d

$$\chi^2 = \frac{(b-c)^2}{b+c} \tag{1}$$

We performed Mcnemar's test on 1) SSTdev 2) Restaurant dev and 3) combination of SST dev and Restaurant dev dataset. Kindly note we could not perform it on the test-set given the test set labels were not accessible.

Table 6: Summarized performance Model1 Model2

Dataset	p-value	Better Model
SST-test	.025	Model1
Rest-test	.145	No difference
SST + Rest	.77	No difference

It was interesting to see in Table 6 that Model 1 performed better than Model2 for SST dev dataset. There was however no difference between Model1 and Model2 on Restaurant dev and combination of SST dev and Restaurant dataset. This shows that the addition of the hand crafted/custom features didn't result in an improvement in prediction performance for sentiment classification tasks and that BERT embeddings already capture the information contained in these hand-crafted features or these hand-crafted features do not provide any incremental information for prediction which BERT embeddings are already capturing

8 Conclusion

In this paper, we tried to analyze the impact of adding hand crafted features to an existing contextual word representation framework like BERT and see if there is an improvement in prediction scores(macro average f1-scores) for ternary sentiment classification on two different domains(movies, restaurants). Based on the experiment, we see that there is no statistically significant improvement in f1-scores. However, we would like to note that with elaborate hyperparameter tuning on the data, it is possible to uncover better performance for both models. It was not feasi-

ble in this experiment given the availability of limited computing capacity. Also the model was only run on 1 epoch given the computing restraints. It would have been interesting to see what the performance looks like on multiple epochs.

References

Chunning Chi, Yongjian Huang, Xiaoyan Li, and Peng Li. 2020. Adversarial and domain-aware BERT for cross-domain sentiment analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: ACL 2020*, pages 4084–4094, Online. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. In *Foundations and Trends® in Information Retrieval*, volume 2, pages 1–135. Now Publishers Inc.

Richard Socher, Alex Perelygin, Jean Y 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*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Duyu Tang, Bing Qin, Ting Liu, and Xiaodong Yang. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1422–1432.

Peter D. Turney and Michael L. Littman. 2002. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 417–424. Association for Computational Linguistics.

Shanshan Yu, Jindian Su, and Da Luo. 2019. Improving bert-based text classification with auxiliary sentence and domain knowledge. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 5145–5152. IEEE.