Temporal Features in Sentiment Analysis

W266 Natural Language Processing (MIDS - UC Berkeley)
Mrinal Chawla and Aditya Khurana
{mrinalchawla, adi_khurana}@berkeley.edu

Abstract

With the growing usage of Internet spaces for discussion, shopping, and all sorts of interaction, lots of sentiment is now being expressed online. Strong sentiment can sway the opinions of others and lead to echo chambers during online interaction. In order to prevent this phenomenon, a reliable method of sentiment classification can be used to monitor sudden changes in sentiment. Previous work has suggested that the year some text is written is important to the accuracy of a sentiment classifier. The experiments conducted in this paper attempt to verify that hypothesis. The research will use several configurations of CNNs and BERT models to find if passing in the year a piece of text was written can improve the accuracy of a classifier. Ultimately, the temporal features did not prove much value, as the loss in accuracy for the classifiers was due to other characteristics of the text.

Introduction

In the current landscape of the Internet, communities such as Twitter and Reddit are hubs for news, politics, and all sorts of social discussion. Within these spaces, lots of sentiment is expressed, and that sentiment can shift readers' perception, leading to phenomenon known as "echo chambers" or "hiveminds". These can even be prevalent in places such as retail websites, where a few reviews with extreme sentiment can sway an individual's perception of a product. Therefore, sentiment analysis of language on the internet is a strong area of application for natural language processing. However, previous work has shown that the accuracy of a sentiment classifier can be affected by the passage of time. Thus, the time that a comment or post was written on the internet may be relevant information when classifying its sentiment. The hypothesis researched in this paper, is that including the year when text is written will improve the accuracy of a sentiment classifier. A variety of classifiers will be used in this research, namely different configurations of Convolutional Neural Nets (CNN) and Bidirectional Encoder Representations from Transformers (BERT) models. The methods for incorporating the year into the model will also vary. The research will attempt to improve the accuracy of some baseline classifiers by using the year in conjunction with the text.

Prior Research and Related Work

The main point of previous research referenced in this work is the experiment that was run by Lukes and Søgaard [1]. In their experiment, a sentiment analyzer was created using a logistic regression model. The data they used was Amazon product review data, where a review rating of 4 stars or higher was considered a positive review, while a review rating of 3 stars or lower was considered a negative review. In their experiment they trained the classifier with data from a certain time period, and then evaluated the model against data from a subsequent time period. In their research, it was found that the accuracy of the model dropped ~2% during the evaluation step. Their approach to find a solution for this problem revolved around predictive feature selection, using the polarity of certain n-grams as a decision metric. However, the research done in this paper tests the inference that the time a text is written is important for accuracy of a sentiment classifier, and a model can learn valuable context if given the year the text is written.

While Lukes and Søgaard used a logistic regression classifier, other papers show that other models may be better suited to sentiment classification. As shown by Liao et. al., CNN models have an advantage in image recognition and text classification tasks [2]. The convolutional layers allow the model to learn distinct features in a global context. Therefore, these pieces of information are tied together with a shared context, allowing the model to make an overall classification of a particular piece of text. Their research proved that a CNN performing sentiment analysis achieved an accuracy of 75.39% using unigrams, while a State Vector Machine achieved 71.35% accuracy [2,6]. Since evaluating the hypothesis in this paper will involve sentiment analysis, a CNN would be a good model construct to experiment on.

Additionally, as shown by Hoang et al., BERT models are also able to learn contextual information when used in a classification context [3]. The advantage of BERT models is that they are able to learn both forward and reverse context, due to the bi-directional nature of the model. Hoang et. al.'s research showed that BERT can outperform state-of-the-art models for aspect-based sentiment classification by ~5% [3]. While the research done in this paper will focus on traditional sentiment classification, without the aspect component, BERT's ability to gather context could be crucial in improving the accuracy by using the year the text was written.

All models used in the experiments will be evaluated on the basis of accuracy. 20% of the data will be held out for evaluation, which will be used to see how the model performs on unseen data.

Data

The data set used to train and evaluate these models is provided by McAuley et al [5]. The data contains reviews for various Amazon products, ranging from 1999 to 2018. 1 million records are sampled from the data, ensuring that sentiment is balanced between positive and negative. Of this data, there are only sufficient entries for the years 2013 to 2018, so the analysis will focus on these years. These years are also further sampled to 10,000 records per year to create the final dataset for analysis. In total this dataset contains 60,000 records. An 80-20 train-test split is used on this data for all model training and evaluation. The rating distribution is shown in Figure 7.

In order to prepare the data for modeling, the text needs to be cleaned and tokenized. Each review is separated into unigrams and converted to lowercase. Stop words are removed using the nltk english stop words library. Words that occur less than twice in the corpus are also filtered. The result is the vocabulary that is used. The vocabulary is used to train a Keras Tokenizer to create numerical values of each text record. Finally, the text is truncated at 80 tokens, in order to limit overfitting for the models. These tokens will be the inputs to the models described below.

Model Construction

The initial experiments were run using CNN models. In order to have a baseline to compare to, a straightforward Convolutional Neural Net was created. The input to this model was the word tokens. Tokens were fed into a Keras embedding layer that created a 100 dimension word vector for each token. The word vectors were then fed into the convolutional layer. This layer contained filters with 8, 4, and 2 vector sized windows, borrowing the 3-sized filter architecture of the CNN used by Liao et al [2]. There were 32 filters of each size. The output of each filter was fed into a max pooling layer, and the maximum value of each filter was selected. This created 3 32-dimension vectors, corresponding to each filter size. Finally, these vectors were concatenated and fed into a 10-dimensional dense layer utilizing a ReLU activation function. The output of the dense layer was then fed into a single classification node with a sigmoid activation function. If the output of the classification node was classified as positive if the value was over 0.5, negative otherwise. This is the base model that was used as a comparison point for all CNN experiments in this research [Figure 1].

The second model used is identical to the first [Figure 1], however the year that the text was written is prepended to the text. The tokenization and embedding process remains the same, along with the rest of the neural net. This was the initial experiment attempting to pass in the year a text was written. By prepending it to the text, the year becomes a token and a word vector that is passed through the convolutional stage. However, due to the convolutions, the vector corresponding to the year is not seen often. Only the first few convolutions of each size see the year vector. This is supported by the results in Figure 1, where the accuracy was remarkably similar after adding the year to the text. The next models will attempt to solve this issue.

The first attempt at boosting the signal of the year token is the concatenation model. This model utilizes two sets of inputs. First is the text input, which is tokenized the same way as the baseline model. The second is the year the text was written. This year is tokenized in chronological order, with the year 2013 corresponding to token "1" and the year 2018 corresponding to token "6". After tokenization, both the text and the year are fed through independent embedding layers to create vector representations. From here, the text vectors are convoluted, pooled, and concatenated in the same configuration as the baseline model. However, before passing through the dense layer, the year embedding vector is concatenated to the concatenated text vectors. This total concatenation vector is then fed through the dense and classification layers, same as the baseline model [Figure 2]. The intuition behind this model configuration is that the year embedding will have a greater effect on the classification output if it is fed in after convolution. Therefore, at classification time, the year vector is present at every node in the dense layer. Another configuration to boost the year token signal is to sum the vector rather than concatenate it. This model is identical to the concatenation model, but the year vector is summed with the text representation vector rather than concatenated [Figure 3]. As can be seen in Table 1, neither configuration provided substantial results, with the accuracy being similar to the baseline model accuracy.

The final experiment utilizing CNNs used pre-trained GloVe embeddings. These embeddings are able to learn using contextual information. Therefore, the embeddings that are generated with GloVe have some meaning attached to them. As shown by Pennington et. al, models using GloVe embeddings are able to outperform other models in word similarity tasks by ~2-5% [4]. The intuition for using GloVe in this research is that the model can learn some meaning about the word and use that in the year to improve sentiment accuracy. In combination with the year a text was written, the model may be able to classify text into positive and negative with higher accuracy. After the embedding layer, the setup of the model is identical to the concatenation model, to keep the signal of the year strong during classification. As shown in Table 1, this classifier actually drops in accuracy. This can be attributed to the use of pre-trained vectors. Since words that are not in the vocabulary are filtered, more words are filtered from the GloVE models than the other models. The baseline model has 21413 words. The pre-trained word vectors only encompass a vocabulary of 18748 words.

As the CNN's were not successful in improving the accuracy with the year added, BERT models are tried next. The BERT model construction follows a similar process, where a baseline is established and then different methods of passing in the year are utilized. The baseline BERT model is created from a pre-trained model from the 'bert-base-cased' library [Figure 4]. This model is then run on the review text without any modifications. The tokenization process is also pre trained with the BERT model. This model establishes the baseline to compare against.

The same BERT model is then run on the review text prepended with the year the text was written. This is the only modification made to the model. Similar to the CNN, this means that the underlying LSTM in the BERT model only sees the year data once, at the beginning of the input sequence. Ideally, there should be a stronger signal present of the year the text was written. Therefore, another technique is used to add the year to the model. This involves passing in the year and the written text as a tuple to the initial input layer in BERT. This creates two types of input tokens to the BERT model. The first is the year token, the second is the text. This also allows for a better attention mask, where the year is combined with each word's vector during the attention operation. The intuition is that this will boost the amount of times the model sees the year the text is written and help improve the accuracy. The additional attention boost should be critical as the year token is more frequent and differentiated from other similar number tokens.

The final experiment is running using RoBERTa. As Liu et al.[8] showcase that BERT models may have under-emphasised on the impact of dynamic masking and changing the masked token during each training epoch. Also, given the dataset is product review, the intuition was that the product name or keywords may benefit by using RoBERTa's byte-level BPE tokenizer which would split the unknown words into common subwords present in the vocabulary. Review may contain brand or product names or domain specific vocabulary which when tokenized may provide a better context via roBERTa or provide fewer tokens. As the token size in the input layer is limited to 80 tokens, better tokenization may add additional attention and more context especially with verbose reviews. An example of the tokenization is below:

Original:
'New NVIDIA RTX30 has 24GB of V-RAM'.
roBERTa-base:
['New', 'NVIDIA', 'RTX', '30', 'has', 24, 'GB', 'of', 'V', '-', 'RAM'] (token count =*13*)
BERT-base-cased:
['New', 'N', 'VI', 'DI', 'A', 'R', 'T', 'X', '30', 'has', '24', 'GB', 'of', 'V', '-', 'RAM'] (token count =*18*)

The baseline RoBERTa model is created from a pre-trained model using the 'roberta-base' tokenizer. The model inherits from the pertained model 'roberta-base' and follows with a pooling layer, fed into a tanh classifier layer. The RoBERTa model is used in two of the three configurations as the BERT model: a baseline, using the review text and the year and text prepended. Since the rest of the model setup remains identical, a direct comparison can be made between BERT and RoBERTa results.

Given the majority (88%) of the text in the balanced dataset is under 100 words, 80 tokens was selected as the max length for all BERT input models which should be enough tokens to capture the sentiment of each review text [Figure 6]. Also, given the training set is 8k records and test set is 2k records, a batch size of 16 was selected, as a trade-off between generalization, overfitting and training time.

Results:

Baseline model	83.94%
Year prepended to text	83.68%
Year concatenated to text	83.66%
Year summed to text	83.58%
GloVE embeddings	81.53%

Table 1: Results from CNN based models

Baseline model - (BERT)	88.70%
Year prepended to text (BERT)	88.05%
Year as separate signal (BERT)	88.90%
Baseline Model (RoBERTa)	91.48%
Year prepended to text (RoBERTa)	91.73%

Table 2: Results from BERT based models

Discussion of Results

The different techniques of inserting the year into the CNN did not result in improvements in accuracy for the CNN. With a variation of under 1% from the baseline model to the prepended, concatenated, and summed models, and with the GloVE embeddings not making a difference, the hypothesis that sentiment accuracy can be improved by including the year is unable to be proven. Investigating the data where the models made an incorrect prediction reveals some interesting patterns. As seen in Figure 5, the error patterns for the models are roughly identical. The models have the largest issues with reviews that are rated 3 stars. While the data processing in this experiment labels these reviews as negative, it may be more conducive to introduce a 3rd class for neutral reviews. The CNNs also struggle with reviews that are either very short or very long. Reviews that are under 5 words do not offer much in terms of sentiment, hence the model has a hard time classifying it as negative or positive. Long reviews lose a lot of context since the tokenization process cuts off the review at 80 tokens. This is especially true for those reviews where the writers explain positives in the first half, and then negatives in the second half. This is also compounded by language that negates certain adjectives. For example, the text “Will never buy again” is very similar to “Will buy again”, and since the model is tokenized by unigrams, it is likely to give both of these reviews a positive rating. In order to counteract the issues of varying review length and negation language, the experiment can be rerun with a different tokenization process, using bigrams and trigrams in conjunction with unigrams. However, the main hypothesis of this paper was adding in the year that the text was written. It did not appear to have any effect as the CNN models were all plagued by similar issues, and did not stray far from the baseline model.

Similar to the CNN model, BERT models did not significantly improve the accuracy and model predictions were similar to the base models when passing the year as an additional text token or two separate inputs. Although, passing year token had very little effect on the sentiment, BERT did great intrinsically when understanding the text.

During the evaluation phase, error analysis showcased that product ratings may not always have the similar sentiment conveyed within the feedback as demonstrated below.
Example:
‘Works well with fast delivery.’(rated - 1),
‘Good for a 4 year old or a senior citizen.’(rated - 3) and ‘Good for the price’ (rated - 5) convey similar satisfactory sentiment while rating is quite different. BERT had similar sentiment predictions for these examples. Further problematic reviews are shown in Table 3.

Review (Text)	Rating	Label (Actual)	Label (Predicted)
Works well with fast delivery.	*1*	*Negative*	*Positive*
Good for a 4 year old or a senior citizen.	*3*	*Negative*	*Positive*
Good for the price	*5*	*Positive*	*Negative*
a *waist*, but exactly as *promissed*	3	Negative	Positive
Still learning all that it has to offer. The customer service is great! They always walk us through any situation we need to handle.	3	Negative	Positive
It's ridiculous how much easier this makes root touch up.	5	Positive	Negative
2 cute n very cheap	5	Positive	Negative

Table 3:Examples of BERT Baseline model of errors.

Future work

In addition to adding different sized n-grams and adding a neutral class, as discussed in the previous section, removing text tokens, special symbols (emojis, URLs, etc.) should be considered. The BERT model learns by creating an association with other words in the sequence. If the sequence contains noise, this may have an impact on the model's performance as it does not contribute to the classification purpose. As Ek, Bernardy et al.[10] showcased, irrelevant changes in punctuation are correctly ignored by the recent transformer models (BERT) but the review text data contains other noise elements such as links, external reference and other irrelevant noise which can be further explored.

Based on the evaluation matrix and reviewing false negative and false positives distribution, it is inferred that this problem can be better explored as a three-class model, ratings 0-2 can be tagged as negative, 3 being neutral and 4-5 being positive. This is similar to the issues seen with the CNN models.

Review Ratings	Label	Correct	False Negative	False Positive
1	0	18.10%	-	0.45%
2	0	10.70%	-	1.05%
3	0	17.60%	-	2.55%
4	1	6.70%	2.85%	-
5	1	36.85%	3.15%	-
Total		89.95%	6.0%	4.05%

Table 4: Evaluation BERT Baseline model

Conclusion

Overall, the year the text was written does not seem to have a large factor in analyzing the sentiment. Rather, the issues seen with sentiment revolve around the differences in how people write reviews. Reviews that vary in tone or reviews that are not descriptive enough are the common pitfalls for the classifier, rather than not having enough context about the text itself. As shown in the discussion of results, further experimentation can be conducted in order to combat some of those issues. In terms of how to handle the problem of internet "hiveminds" it will depend mostly on the context of the problem. With internet shopping reviews, it was seen that individuals thresholds for a positive review vary with neutral examples being difficult to classify. Twitter or Reddit spaces may have much more defined opinions leading to other pitfalls in accuracy. Ultimately, further experimentation is needed in order to determine which features and configurations are most important in differing situations.

Appendix

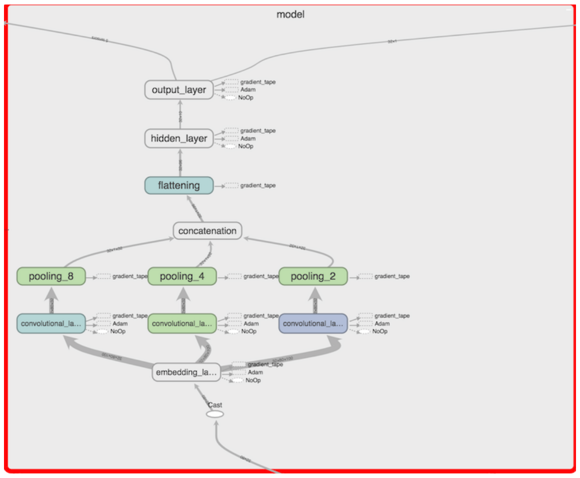

Figure 1: Baseline CNN Model. This model was used as the baseline and with the year prepended to the text

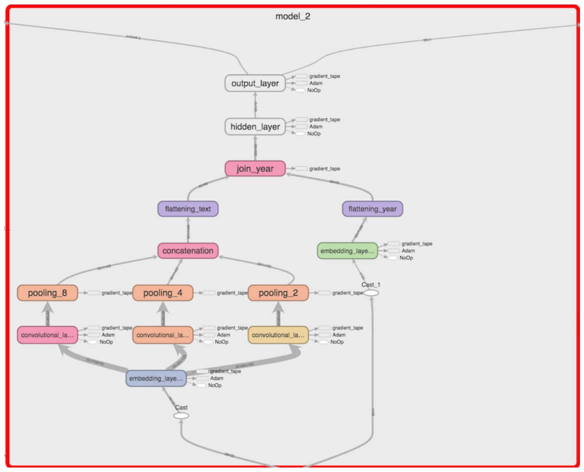

Figure 2: CNN Model with year embedding concatenated to the text embedding. This configuration was used with keras embeddings and with GloVE embeddings.

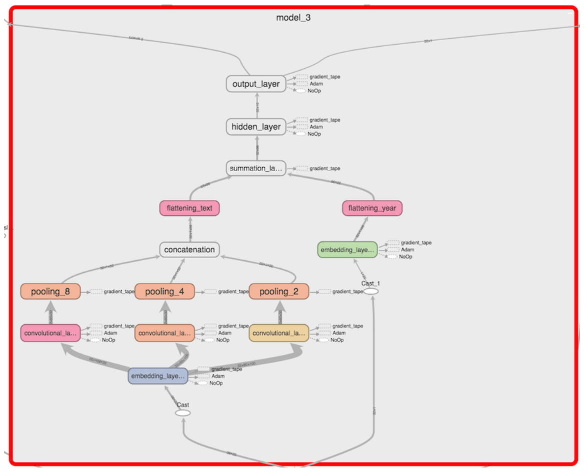

Figure 3: CNN Model with year embeddings summed with the text embeddings

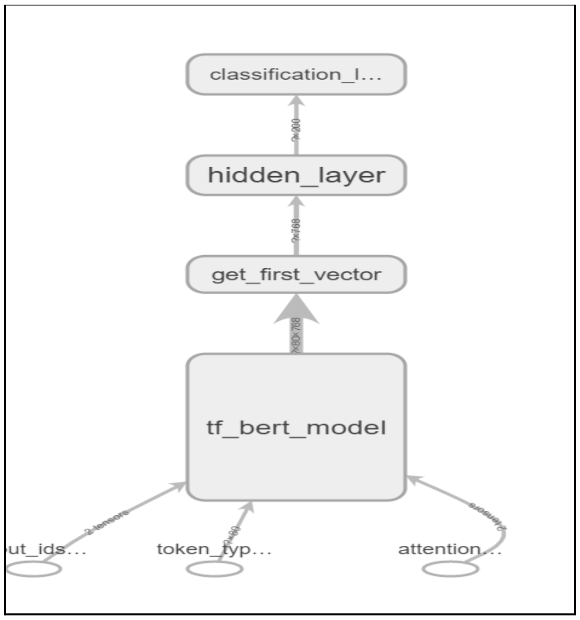

Figure 4: BERT model for base case.

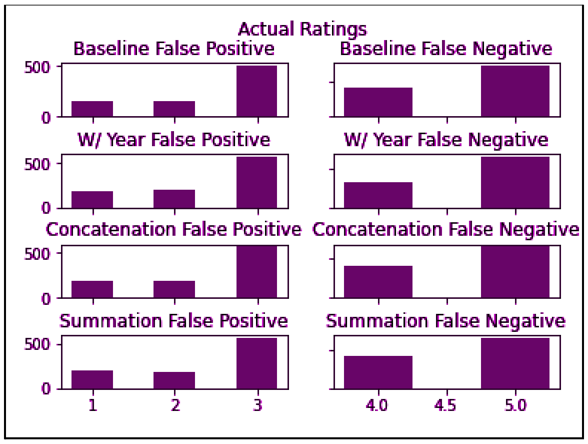

Figure 5: Error patterns of the CNN models

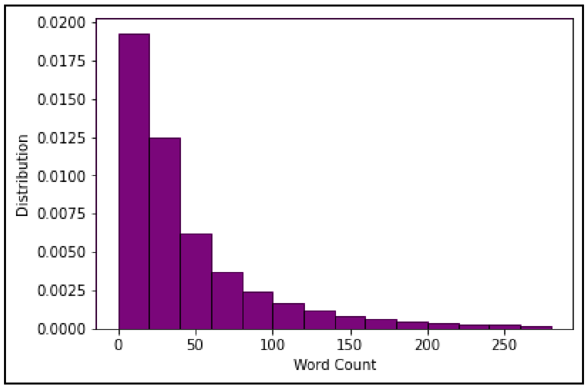

Figure 6: Word length distribution

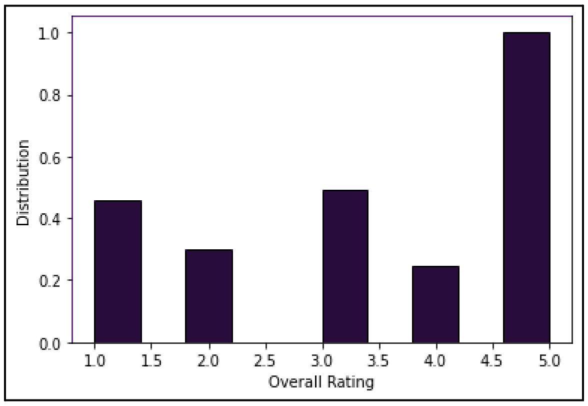

Figure 7: Overall Ratings (Sentiment) distribution

References

1. Lukes, J and Søgaard, A, Sentiment analysis under temporal shift. *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 2018, p. 65-71
2. Liao, S, Wang, J, Yu R, Sato, K, and Cheng, Z , CNN for situation understanding based on sentiment analysis of twitter data. *Procedia Computer Science*, 2017, vol. 111: p. 376-381
3. Hoang, M, Bihorac, O, and Rouces, J, Aspect-Based Sentiment Analysis using BERT. *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, 2019, p. 187-196
4. Pennington, J, Socher, R, and Manning, C, GloVE: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP),* 2014, p. 1532-1543
10. Ek, A, Bernardy, J, and Chatzikyriakidis, S, How does Punctuation Affect Neural Models in Natural Language Inference. *Proceedings of the Probability and Meaning Conference (PaM 2020)*, 2020, p. 109–116
5. McAuley, J, Targett, C, Shi, Q, and Hengel, A, Image-based recommendations on styles and substitutes**.** *SIGIR,* 2015
6. Agarwal, A, Xie, B, Vovsha, I, Rambow, O, Passonneau, R, Sentiment Analysis of Twitter Data. *Proceedings of the Workshop on Language in Social Media*, 2011, p. 30-38
7. Wolf, T, Debut, L, Sanh, V, Chaumond, J, De- langue, C, Moi, A, ... & Brew, J, Trans- formers: State-of-the-art Natural Language Pro- cessing. 2019.
8. Liu, Y, Ott, M, Goyal, N, Du, J, Joshi, M, Chen, D, Levy, O, Lewis, M, Zettlemoyer, L, and Stoyanov. V, Roberta: A robustly optimized bert pretraining approach. 2019.
9. Sennrich R, Haddow, B, and Birch A, Neural machine translation of rare words with subword units. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics,* 2016, vol. 1: p. 1715–1725.