BERT-SENTI: Sentiment Analysis into Tweets

Anonymous Author(s)

Abstract

Sentiment analysis serves as a subfiled of NLP and has drawn much attention from researchers these days. In this paper, we target at the sentiment analysis specially on Twitter(X). After detailed statistical research on related dataset sentiment140, we find the relationship between the tweet content, tweet posting time and the sentiment. We build our own model BERT-SENTI to get the sentiment classficiation, which exceeds the baseline models in performance.

1. Introduction

Sentiment analysis is a computational study of people's opinions or sentiments of an entity[1]. Generally sentiment analysis can be treated as a classification problem with multiple levels such as sentence-level, document-level, corresponding to different entities.

Sentiment analysis is an useful technique upon researching the feedback from the market. For example, on analyzing the sentiments of product reviews of users, the company can make better decisions on telling from users' likeness and improving their products. Besides this, sentiment analysis is also useful in business investigation.

In this paper, we draw our attention into one of the largest social media: twitter(or X), which contains large number of users and different topics. In twitter, users primarily write tweets in text-form. Twitter is an excellent platform for sentiment analysis because of its popularity as well as the fact that those tweets posted on Twitter usually contain opinions about those topics related to users' interests[2].

2. Dataset

In this paper, we utilize the **Sentiment140 dataset**, which contains 1.6 million tweets with metadata and their sentiments. In this section, besides the basic statistics of it, we analyze the dataset from three aspects:

- Length analysis.
- Tweet content.
- Tweet posting time.

For each query in the dataset, it contains six types of information: sentiment, id, tweet posting time, searching query, user id and the tweet content. A typical query sample can be seen in 1.

Sentiment	0
Id	1467811592
Tweet Posting Time	Mon Apr 06 22:20:03 PDT 2009
Searching Query	NO_QUERY
Tweet Content	Need a hug

Table 1. A sample query in the dataset. Sentiment includes 0(negative) and 4(positive).

In this balanced dataset, positive and nagative tweets take both half of it. There are in total 659,775 unique user ids. All these tweets are collected from April 2009 to June 2009.

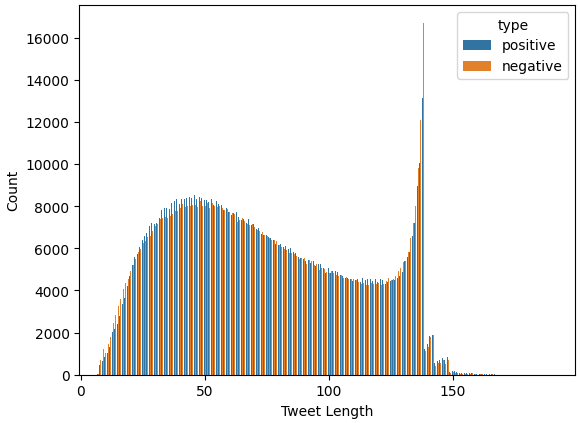

Figure 1. Tweet length distribution of positive and negative tweets. Here we exclude a very little part of whose lengths are bigger than 200.

Rank	Positive			Negative		
	Unigram	Bigram	Trigram	Unigram	Bigram	Trigram
0	i'm	can't wait	get 100 followers	i'm	wish could	. . .
1	good	good morning	100 followers day	get	feel like	wish could go
2	love	i'm going	followers day using	go	last night	i'm gonna miss
3	-	looking forward	add everyone train	like	i'm going	i'm going miss
4	like	good luck	everyone train pay	work	want go	please help find
5	get	cant wait	train pay vip	got	wanna go	help find good
6	day	getting ready	can't wait see	going	looks like	find good home.
7	going	last night	day using www.tweeteradder.com	can't	i'm sorry	lost. please help
8	got	good night	using www.tweeteradder.com add	miss	go back	hope feel better
9	u	happy birthday	www.tweeteradder.com add everyone	really	i'm gonna	i'm sorry hear

Table 2. Top10 unigrams, bigrams and trigrams for positive and negative tweets.

For exploring the **relationship between tweet length distribution and its sentiment**, among the 1,600,000 tweets, the majority of them have the length shorter than 200 characters1. To compare the tweets of different sentiments, we can find out that generally negative tweets distribute more densely in shorter and longer lengths while more positive tweets are in the medium lengths.

For exploring the **relationship between tweet content and its sentiment**. In order to provide the most directive impression, we generate the word cloud picture of negative and positive tweets in 2 and 3. Except for those common high-frequency words, we can find that some words such as "work", "now", "want" appear most in the negative tweets. While some words such as "love", "thank" appear most in the positive tweets.

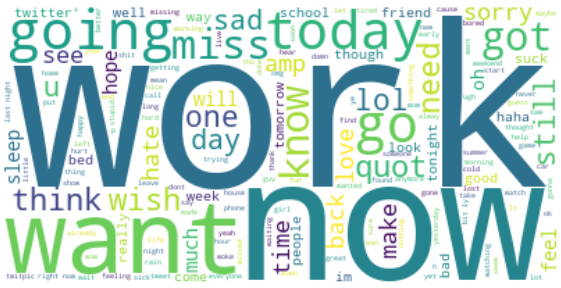

Figure 2. Word cloud picture of the negative tweets.

Figure 3. Word cloud picture of the positive tweets.

At the same time, we provide with detailed data of

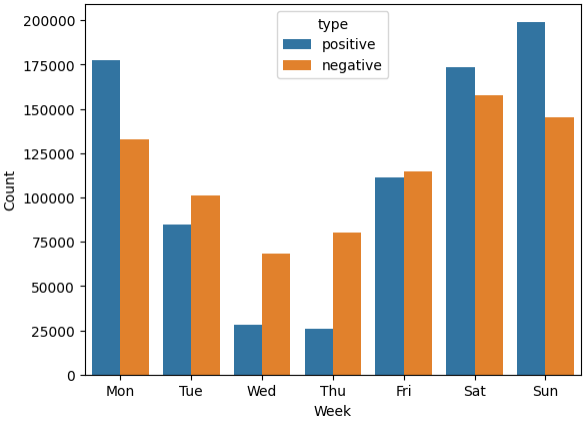

Figure 4. The number of tweets posted in different days in a week.

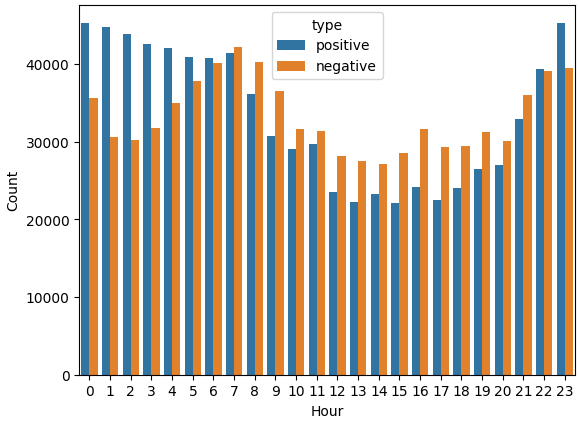

Figure 5. The number of tweets posted in different hours in a day.

the frequent unigrams, bigrams and trigrams in positive tweets as well as negative tweets2. We have eliminated the stop words and standardize words to be in lower cases.

For exploring the **relationship between tweet posting time and its sentiment**, we provide on how different tweets are posted in each day in a week4 and in each hour in a day5.

For seven days in a week, there are generally more positive tweets posted on Monday, Saturday and Sunday, especially more positive posts than negative posts on Sunday. On the contrary, there are more negative posts on Tuesday, Wednesday and Thursday. On Sunday there are extremely higher percentage of negative posts.

For different hours in a day, in the daytime there are usually more negative posts while at midnight it's the opposite.

In conclusion of above, the sentiment of one tweet has clear relationship with its content as well as its posting time.

Specially mention that, due to the machine capacity, the results below are conducted under a randomly-selected 10% subset of the original sentiment160. The data distribution is remained the same.

3. Related Works

Over recent decades, sentiment analysis has been explored by a large number of researchers. The methods for sentiment analysis has gone through multiple developing periods.

At the beginning periods, researchers mainly use lexicon-based approaches. The words in the documents are marked with the sentiment orientation in a dict[3, 4]. To create such dict, researchers can use seed words in a corpus, or derive from open resources[5]. Lexical-based methods have high interpretability. However, the cost and ability of such models are not that satifying.

Later, more researchers delve into the area of machine learning, where models such as decision tree, linear classification, naive bayes become popular and reveal great performance[6, 7]. The machine learning models achieve great performance and can be easily tuned to different subdomains. The challenge lies in that the researchers are required to complete much feature engineering to find the suitable features for the datasets. Common features include bag-of-words(BOW)[8], ngrams[9] and TF-IDF[10].

Over recent years, the popularity of deep learning models have overcome the problems of cumbersome feature engineering work[11]. Those emerging deep learning models achieve state of the art results as well as reduce human work. The only drawback is the large data and resources consuming.

The challenges for sentiment analysis now are first the domain dependency of documents. Secondly, the implicit meaning in the sentiment-related contents strongly influence the model performance as well[12].

Besides the seniment140 dataset we use, there are multiple other datasets on different domains including but not limit to IMDb Movie Reviews[13], Stanford Sentiment Treebank[14] and MPQA Opinion Corpus[15].

4. Methods

4.1. Task

As mentioned in previous sections, we'll complete sentiment analysis on tweets based on the Sentiment160 dataset. In this specific situation, we complete on the binary predictive task as6:

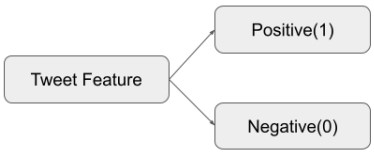

Figure 6. Word cloud picture of the negative tweets.

For the total dataset, we randomly select 20% of it, which is of 32,000 samples, to be the test set. The test set equally contains the same number of negative samples and positive samples. And the performance on this task is evaluated on the model accuracy on the test set.

The model training and validating will utilize the remaining train set of size 128,000.

4.2. Models

The models for sentiment analysis can be divided into machine learning models and deep learning models, whose strengths and weaknesses are discussed in 3.

For comparison and baselines, we adopt traditional machine learning models including:

- Naive Bayes[16]
- Random Forest[17]
- SVM[18]
- Decision Tree[19]

The strong performance of recent deep learning models has greatly exceeded the traditional methods

on the sentiment analysis. Thus we utilize some deep learning models as baselines and implement our own deep learning model as below:

- CNN[20]
- RNN[21]
- **BERT-SENTI(Ours)**7

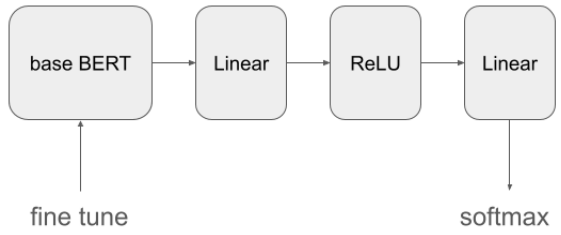

Figure 7. Word cloud picture of the negative tweets.

The large language models(LLMs) show extremly high performance on different kinds of downstream tasks. These LLMs rely on large corpus and have plenary latent meanings. Among which BERT[22] owns reputation for great generalization ability. To make use of the large semantics space behind the LLM, we fine-tune BERT on our dataset. The finetuned model **BERT-SENTI** can benefit from the BERT base model as well as the specific information from our dataset. To reduce the problems of overfitting, we adopt early-stopping.

In our task, the features for above will contain the tweet content and the tweet posting time. The usability of these two kinds of information are proved in previous analysis on the dataset.

For traditional machine learning models, we extract the TD-IDF feature of one to three n-grams and the one-hot embedding of the weekdays, months, days in a month and the posting hours3.

Feature	Dimension
TF-IDF	1,272
One Hot for Time	61
Sentence Embedding	300

Table 3. The feature dimensions.

For deep learning models, as the neural network is more capable for processing dense information, the TF-IFD sparse matrix is not suitable[23]. Thus we first train a word2vec model based on the original Sentiment140, the embedding of one sample is represented by the average value of the word vectors in the sentence.

5. Results and Analysis

Using the experiment setting as above, we implement our model and measure some baseline models results4.

Model	Accuracy	Accuracy*
Naive Bayes	0.7078	0.7168
Random Forest	0.7430	0.7815
Support Vector Machine	0.7614	0.7973
Decision Tree	0.6888	0.7270
CNN	0.6989	0.7487
RNN	0.7623	**0.8081**
BERT-SENTI	**0.8084**	-

Table 4. Baseline results of model accuracy on the test set. We have done abalation experiments by eliminating the time-related features. The ablation accuracy is shown without *.

Among all the models we could find out that generally deep learning baselines have better performance than the traditional machine learning baselines. Among which the RNN has the highest accuracy score of 0.8081.

From the ablation study we find that for all the baselines, those with two input areas(content, posting time) appear better than those with only one input area(content).

For our BERT-SENTI model, among all the models with the content-related features, it's of the highest accuracy, higher than Decision Tree by around 10%, and higher than RNN by around 4%.

6. Conclusion

In the analysis section, we figure out that the positive and negative tweets have different length distribution: negative tweets distribute more in the shorter and longer ranges. At the same time, we have shown the top 10 unigrams, bigrams and trigrams for both positive and negative tweets, with wordcloud pictures. In understanding the relationship between sentiment and posting time, we find out that people tend to post positive tweets during weekends and midnights.

On the sentiment prediction models, we have implement several baselines including Naive Bayes, Random Forest, Support Vector Machine, Decision Tree, CNN and RNN. These models reveal acceptable performance. To furthur increase the accuracy, we have implemented our own model BERT-SENTI, which is the fine tuned model on the base BERT. BERT-SENTI achieves highest accuracy of 80.84%.

There are several limitations on our BERT-SENTI model: first, although we have proved that the tweet posting time contains some degree of information by

analyzing the dataset as well as measuring the baseline models, we didn't implement the BERT-SENTI with time-related features because the the base model which we fine tune on is with only content-related features. Secondly, due to machine capacity, we only train our model on a subset of size 160,000. Although the size is satisfying for a common classification model, the performance may increase using the whole dataset.

References

[1] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams engineering journal*, vol. 5, no. 4, pp. 1093–1113, 2014.

[2] A. Giachanou and F. Crestani, "Like it or not: A survey of twitter sentiment analysis methods," *ACM Computing Surveys (CSUR)*, vol. 49, no. 2, pp. 1–41, 2016.

[3] C. S. Khoo and S. B. Johnkhan, "Lexicon-based sentiment analysis: Comparative evaluation of six sentiment lexicons," *Journal of Information Science*, vol. 44, no. 4, pp. 491–511, 2018.

[4] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-based methods for sentiment analysis," *Computational linguistics*, vol. 37, no. 2, pp. 267–307, 2011.

[5] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[6] M. Neethu and R. Rajasree, "Sentiment analysis in twitter using machine learning techniques," in *2013 fourth international conference on computing, communications and networking technologies (ICCCNT)*, pp. 1–5, IEEE, 2013.

[7] B. Agarwal, N. Mittal, B. Agarwal, and N. Mittal, "Machine learning approach for sentiment analysis," *Prominent feature extraction for sentiment analysis*, pp. 21–45, 2016.

[8] Y. Zhang, R. Jin, and Z.-H. Zhou, "Understanding bag-of-words model: a statistical framework," *International journal of machine learning and cybernetics*, vol. 1, pp. 43–52, 2010.

[9] S. Banerjee and T. Pedersen, "The design, implementation, and use of the ngram statistics package," in *International Conference on Intelligent Text Processing and Computational Linguistics*, pp. 370–381, Springer, 2003.

[10] A. Aizawa, "An information-theoretic perspective of tf–idf measures," *Information Processing & Management*, vol. 39, no. 1, pp. 45–65, 2003.

[11] A. Yadav and D. K. Vishwakarma, "Sentiment analysis using deep learning architectures: a review," *Artificial Intelligence Review*, vol. 53, no. 6, pp. 4335–4385, 2020.

[12] D. M. E.-D. M. Hussein, "A survey on sentiment analysis challenges," *Journal of King Saud University-Engineering Sciences*, vol. 30, no. 4, pp. 330–338, 2018.

[13] A. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pp. 142–150, 2011.

[14] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013.

[15] J. Wiebe, T. Wilson, and C. Cardie, "Annotating expressions of opinions and emotions in language," *Language resources and evaluation*, vol. 39, pp. 165–210, 2005.

[16] I. Rish *et al.*, "An empirical study of the naive bayes classifier," in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol. 3, pp. 41–46, 2001.

[17] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.

[18] W. S. Noble, "What is a support vector machine?," *Nature biotechnology*, vol. 24, no. 12, pp. 1565–1567, 2006.

[19] Y.-Y. Song and L. Ying, "Decision tree methods: applications for classification and prediction," *Shanghai archives of psychiatry*, vol. 27, no. 2, p. 130, 2015.

[20] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: analysis, applications, and prospects," *IEEE transactions on neural networks and learning systems*, 2021.

[21] A. Sherstinsky, "Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network," *Physica D: Nonlinear Phenomena*, vol. 404, p. 132306, 2020.

[22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[23] Y. Zhao, J. Li, C. Liao, and X. Shen, "Bridging the gap between deep learning and sparse matrix format selection," in *Proceedings of the 23rd ACM SIGPLAN symposium on principles and practice of parallel programming*, pp. 94–108, 2018.