# Search Term Specific Sentiment Classification for Business News


**Yuwei Wang**
New York University
yw1854@nyu.edu

**Yi Xu**
New York University
yx2090@nyu.edu

**Yunya Wang**
New York University
yw4509@nyu.edu

**Isabel Zhou**
New York University
yz6126@nyu.edu



## Abstract

In this project we propose an initial approach for sentiment analysis for given search terms. As observed in real world texts, when multiple entities exist together in one sentence, the sentiment scores towards each entity often varies and can't be represented by the overall sentiment of the entire sentence. We firstly propose an idea for sentence decomposition and sentiment assignment with Semantic Orientation, then we incorporate the information gained from syntax mining into neural network (BERT), tried out different methods of adjusting embeddings, and compared results. Our experiments showed that incorporating search term position vectors significantly outperformed fine-tuned BERT and improved accuracy for sentiment classification of given entities.


## 1 Motivation

Business news mining, especially sentiment analysis, has helped greatly in predicting stock prices and targeting market opportunities. However, most existing news sentiment classification projects suffer a ceiling of accuracy due to the limitation of generic sentiment libraries. Our project aims to conduct a sentiment analysis of business news by digging into the syntax level relationship, in hope to increase the accuracy of classification predictions.

In most of the sentiment analysis, the model takes the entire sentence or paragraph as a whole, and assign labels according to the overall tune. This method, however, doesn't help detect sentiment score for any specific target word in the sentence. For example, consider the following two pairs of news titles for company J.P. Morgan (target):

| News | Sent. target |
|---|---|
| Fidelity replaces J.P.Morgan as sub-advisor on 2 strategies | Negative |
| J.P.Morgan replaces Fidelity as sub-advisor on 2 strategies | Positive |
| Mercer drops J.P.Morgan as managers of bond fund | Negative |
| J.P.Morgan drops Mercer as managers of bond fund | Neutral |

The sentiment label for our search term can be completely shifted given different positions of the term appearing in the sentence. Our goal is trying to solve this tricky situation through taking syntax level relationships into consideration such as subject, object, active or passive verbs and the impact (sentiment) of the verb on the subject or the object depending on our studied targets.

## 2 Data Description

Although there exists many open sources text data sets of business news, they are not classified for any search term nor labeled with sentiment score with certain terms. Thus, we chose to construct a new dataset. We have decided on two major sources of news in financial market: Pension & Investment and Google News. Both websites allow free news search for a given organization name and the structures of news title sentences are similar.

We web-scraped news headlines from these sites. The reason of using headlines only is to exclude much noises caused by complicated structure of sentences and make our proposed algorithm feasible for the task. We also manually dropped observations without much useful information. After cleaning process, the dataset has roughly 1000 examples left with obvious sentiment score towards certain search terms. Then we manually label each headline. In risk management, negative news evokes stronger reactions than positive or neutral news so we decide to integrate neutral labels into positive labels, leaving only two categories: positive (60%) and negative (40%).

## 3 Related Work

Semantic Role Labeling (SRL) refers to the process that assigns labels to each word or phrases in sentences based on their semantic roles, such as Agent, Patient, Instrument, etc. These roles represents the relationship that a syntactic constituent has with a predicate. Recognizing and labeling semantic arguments is a very important part in our project.

The two main English corpora with semantic annotations from which to train SRL systems are PropBank and FrameNet. CoNLL-2004 and CoNLL-2005 are two famous conferences focusing on the recognition of semantic roles for the English language, based on PropBank predicate-argument structures. The overall best model from the conference is Generalized Inference with Multiple Semantic Role Labeling Systems, built by Koomen et al. (2005). This method takes the output of multiple argument classifiers and combines them into a coherent predicate argument output by solving an optimization problem, which is solved via integer linear programming. This method reaches 79.44 F1 score in test dataset.

Turney and Littman (2003) introduced a method for inferring the semantic orientation of a word from its statistical association with a set of positive and negative paradigm words. They tried two instances of this approach: pointwise mutual information (PMI) and latent semantic analysis (LSA) and were able to achieve an accuracy of 95%.

Devlin et al. (2018) introduced a state-of-art language model called Bidirectional Encoder Representation from Transformers (BERT). It pretrained deep bidirectional representation on both left and right context in all layers on the corpus of BooksCorpus (800M words) ((Zhu et al., 2015)) and English Wikipedia (2,500M words). And the pre-trained BERT can be futhur fine-tuned on dowmstream task with just one additional output layer.

# 4 Methodology - Semantic Orientation

Our initial idea consists of following two major parts for a sentiment classification algorithm.

## 4.1 Detect the role of a search term

To see the particular role of a given search term, we start by segmenting the news title sentence and extracting different parts. Our proposed tool for decomposing given sentence is *Stanford OpenIE* (Angeli et al. (2015)), as part of *Stanford CoreNLP* application. Open Information Extraction extracts the subject, relation and object of given sentence. We can then create indicator variables for the search term as subject or object in the title.

## 4.2 Sentiment label for relation verb

*OpenIE* also returns the relation term of the sentence, and we would extract the core verb in the relation. Then we would utilize *Semantic Orientation* (Turney and Littman (2003)) to assign sentiment label for the verb, indicating whether it's positive or negative to both subject and object specifically. (For example, the word "replace" in example would be classified as positive towards subject, and negative towards object). Because *semantics orientation* powerfully takes seeds of words in opposite directions of any given dimension, we would be able to distinguish the different function labels of a verb towards subject and object respectively. Our initial exploration has also proven this is highly feasible.

The final prediction label of news title would be a combination of two parts above, as a function of position of the search term and verb function label towards the position.

| News | sub | obj | rel | sent. s | sent. o | out |
|---|---|---|---|---|---|---|
| Fidelity replaces J.P.Morgan as advisor | 0 | 1 | replace | 1 | -1 | -1 |
| D.C. Retirement terminates J.P.Morgan | 0 | 1 | terminate | 1 | -1 | -1 |

Table 1: Examples of processing: sub and obj are indicators of position of search term, sent.s and sent.o are sentiments of relation verbs towards subject and object respectively; out is final prediction label

## 4.3 Limitation of Semantic Orientation

However, there are multiple limitations of the Semantic Orientation method. First of all, extraction of search term and relation verb is not always reliable, sometimes it may produce empty outputs. Secondly, training Semantic Orientation (detecting relation verb's contribution to subject/object) is tricky. It's highly dependent on input of seed words, which causes great variance of accuracy. Overall, 736 out of 1000 observations have valid outputs; and among valid outputs, validation accuracy is around 68.7%.

# 5 Methodology - Bert Based Model

Given the limitations mentioned in the previous section, we started to incorporate the state of the art model BERT into our project.

First of all, we just simply implement fine tuning BERT by calling BertForSequenceClassification directly and pass in the tokenized headlines into the model and training it against the target labels. This model is really simple and quick to give us an overall view of the BERT based classification

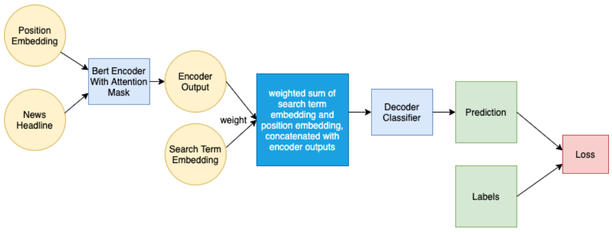


Figure 1: Bert Model Structure


Figure 2: Loss Curves During Training

performances on our dataset without any extra embedding or model structure change.

Then we further experiment it by change our model into encoder and decoder structure, where our encoder is the pre-trained BERT (bert-based-uncased) and the decoder is a linear classifier where it passes the BERT encoder output into output dimension which is 2 in our case. With this model we compare it with the fine tuning BERT to see if it has any improvement.

But in order to tackle the problem of search term specific sentiment classification, we firstly tried to add in search term embedding together with the headline into the model. In the final version, the best performing one is the one where we concatenate the search term and headline and pass the concatenated input into the BERT model encoder; additionally we also pass in attention masks into the BERT encoder as well.

Lastly, we added in the subject and object relationship on top of the previous version. The subject and object relationship are encoded as a list of indicators like [1,0] or [0,1], which it represents whether the search term is subject or object. In scenarios, where our algorithms (mentioned in precious section) could not differentiate the relation, we will return [0,0] as the output of the position encoding. As for the position encoding, we pass them through embedding layer in the decoder and concatenate it with encoder output. We tried different weights for the concatenation and the best working one is 3.5 to 1 for position encoding to encoder output. For the visualized model structure, please check Figure 1.

## 6 Experiments & Results

### 6.1 Experiments

In order to find out the optimal algorithm, we tried all methods mentioned in Part 4. For all BERT variations, we set the same configuration on number of epochs (20) to make the results more comparable, even though fine tuning BERT prone to over-fit very quickly after 4 epochs. Also we elaborated gradient clipping, separate optimizer and scheduler for encoder and decoder to help the convergence of the model.

In this paper, we take the final method mentioned in Part 4 as an example to illustrate the model training process. Figure 2 shows how the training and validation losses change over 20 training epochs. Starting from training, training loss constantly decreases until converge. In the meanwhile, the validation loss curve decreases as well in the beginning stages, and it reaches the minimum at epoch 11 but after that, the loss goes up instead. Increased validation loss and decreased training loss indicate the model over-fitting. So we save the model right before over-fitting during training as one of model candidates to test on test dataset. This same procedure is applied for all methods.

### 6.2 Evaluation Matrix

The label distribution is balanced in this problem, so it is natural to use accuracy as one of the evaluation matrices for classification. Taking consideration of the business usage of the model, we decided to use false positive rate as another matrix. False positive error refers to the situation where the search term related news is negative in fact but its prediction is positive. If the model mistakenly labels it as positive, it is very likely to impact users to overestimate the company's performance and make unreasonable investment decision. Thus, we want the false positive rate to be as low as possible.

Figure 3 shows the model performance on test dataset for the final version of BERT based model. The sum of diagonal value is the model accuracy, which is 93.15% in this case. Off-diagonal values contain false negative rate and false positive rate. In this case, 4.11% false positive rate means among all true negative labels, only 4.11% instances are

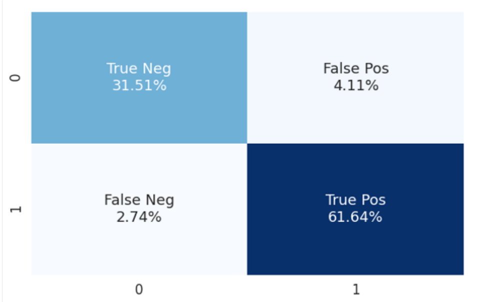


Figure 3: Confusion Matrix on Test Set

mistakenly predicted as positive.

### 6.3 Results

Table 6.3 presents all the model performance of BERT variations on test dataset. Fine-tuned BERT and BERT with self-defined classifier have close results in both accuracy and false positive rate, and BERT model with self-defined classifier even has a worse performance when incorporating search term information. It might be caused by the limited size of training dataset. And more complicated model might fail to learn the additional information given the additional information.

Overall, based on accuracy and false positive rate, the BERT Model with search term and position encoding yields the best result. And we take this model as our final optimal model.

| Model | Acc | FPR |
|---|---|---|
| Fine-tuned BERT | 0.86 | 8.22% |
| BERT + Self-defined Classifier | 0.87 | 8.22% |
| BERT + Self-defined Classifier + Search Term Embedding | 0.85 | 10.96% |
| BERT Encoder + Attention Mask + Decoder with Concat(Weighted Sum of Search Term Embedding and Positional Embedding, Encoder Output) + Self-defined Classifier | 0.93 | 4.11% |

## 7 Conclusion

In conclusion, neural network is very powerful comparing to other constructed methods, such as sentence segmentation and semantic orientation method we initially tried in Section 4. Incorporation of more information like object subject relationship and search term help training and significantly improve the model performance. And adding the state-of-the-art pre-trained BERT model helps overcome the limited dataset constraint and also achieve the best performance.

## References

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354, Beijing, China. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Peter Koomen, Vasin Punyakanok, Dan Roth, and Wen-tau Yih. 2005. Generalized inference with multiple semantic role labeling systems. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 181–184, Ann Arbor, Michigan. Association for Computational Linguistics.

Peter D. Turney and Michael L. Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Trans. Inf. Syst.*, 21(4):315–346.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

## Contributions

Yuwei Wang: Methodology Semantic Orientation
Yi Xu: Experiments, Results and Evaluation, Related Work
Yunya Wang: Methodology Bert Based Models
Isabel Zhou: Motivation, Data Scraping and Description, Related Work