Multiclass Sentiment Classification and Aspect Segmentation on Movie Reviews

Zelin Li
zl3611@nyu.edu

Wenhao Xu
wx2056@nyu.edu

Shuochen Zhao
sz3315@nyu.edu

Abstract

Sentiment classification, also referred to as opinion mining, is an approach that identifies the emotional tone behind a body of text. A movie review is a great corpus for doing sentiment classification. This paper aimed to go beyond just classifying sentiments but to advance the understanding of movie reviews by exploring movie aspects. To be specific, this paper focuses on **two** sections: 1. Identifying multiple sentiment classes. 2. Segmenting movie review sentences according to various aspects of the movie-watching experience. In this paper, we only segment movie review sentences into **five** aspects: theater facilities; movie plots, actor & actress performances; movies' special effects and scenes; and others. In the **sentiment classification section**, with the features containing word frequency, n-grams and TD-IDF, the model Multinomial Naive Bayes generated the best performance of 0.58 F1 score, outperforming the baseline's F1 of 0.20. In the **movie aspect segmentation section**, we implemented control variable experiments using Cosine similarity and four machine learning algorithms on TF-IDF and Word2Vec features on differently modeled aspect target datasets, among which K nearest neighbors model on word2vec embedding methods generated the best result of 0.83 F1 score, which significantly supersedes baseline F1 of 0.17. Overall, ultimate results from both sections outperform the baseline model and fulfill the requirement of sentiment analysis and aspect segmentation. By extending the scope of sentiment classification research, this paper has the potential to impact various applications beyond movie reviews, such as product reviews and social media analysis.

1. Introduction

Sentiment classification, also referred to as opinion mining, is an approach to natural language processing that identifies the emotional tone behind a body of text. A movie review is a great corpus for doing sentiment classification since it contains various emotions from different people based on the movie they have watched. There are profound research on different models /for sentiment classification. For instance, Socher introduces the Recursive Neural Tensor Network (RNTN), a model for sentiment classification that captures semantic compositionality over a sentiment treebank, which outperforms several baselines, achieving state-of-the-art performance at the time (2013). However, the plethora of sentiment research only focuses on the binary division of emotion. In reality, people may take different aspects into consideration when they check the movie reviews. In this paper, we focus on content-related aspects and decide to segment movie review sentences into only **four** important aspects: theater facilities; movie plots, actor & actress performances; movies' special effects and scenes. The rest of the sentences for movie reviews will be classified into the **fifth** aspect of "Others". We choose these four aspects because in real life, people might greatly care about these aspects when they are viewing movie reviews. By performing aspect segmentation on movie reviews after finishing the multiclass sentiment classification, we aim to provide a more detailed understanding of the aspects that

contribute to a movie's overall reception. Therefore, the paper will be divided into **two** parts, for one thing, a leap from binary classification to the detailed multi-class classification, and for the second thing, an experimental inclusion of the four aspect segmentations based on the movie review corpus.

2. Related work

2.1 Sentiment Classification on Movie Review

As for feature extraction, a study by Pang et al. (2002) used n-grams as features in a supervised sentiment classification model for movie reviews. And also, the term frequency-inverse document frequency (TF-IDF) vector, which captures the frequency of co-occurring words and their importance in the document, is widely applied (Masood et al., 2018). Thus, In this study, we use a combination of word frequency, n-grams, and TF-IDF vector as feature inputs, where word frequency and TF-IDF serve as the detector of possible sentiment-relating words, the n-grams dedicate to exploring the association among words.

For the choice of model, previous research has explored various models for sentiment classification. First, SVMs have been praised for their ability to handle high-dimensional feature spaces and have been applied successfully to sentiment classification (Pang et al., 2002). Additionally, CNNs have shown promising results in capturing spatial dependencies in textual data and have been widely used in sentiment analysis tasks (Kim, 2014), although it's more usually used for image classification. Moreover, Naive Bayes classifiers have been utilized due to their simplicity and efficiency, and have demonstrated competitive performance in sentiment analysis (Pang et al., 2002). Random Forests, on the other hand, have been known for their ensemble approach and have shown robust performance in sentiment classification tasks (Breiman, 2001). These models are widely used in sentiment analysis with high performance in each previous research of nearly 80% accuracy. Although CNN is considered as a sophisticated model, other models like SVM, Naive Bayes, and Random Forest still remain competitive since CNN requires fine-tuned parameters and pretrained word vector data to outperform others.

2.2 Aspect Segmentation on Movie Review

TF-IDF is a widely used method, which shows the importance level of each word in a document. It's a statistical feature that describes the document with a numerical vector and can be applied to Information Retrieval tasks (Soucy and Mineau, 2005). For document classification, Cosine Similarity applied to TF-IDF vectors is also one of the most frequently used metrics, which classifies documents that are similar from a large document collection (Dehak, 2010). Applying machine learning algorithms in document classification is also a common method, the idea is to use extracted feature vectors to feed machine learning models (Dumais et al., 1998; Joachims, 1998), and where TF-IDF as the feature can generate good accuracies (Han et al., 2000). Word2Vec is a logistic regression-based embedding framework, which takes word context into consideration, this embedding breakthrough was made by Mikolov et al. in 2013. (Mikolov et al., 2013a; Mikolov et al., 2013b)

2.3 Large Language Models (ChatGPT)

With the advent of GPT-3 (Brown et al., 2020), Large language models (LLMs) began to come into people's sight. They generally possess a large number of model parameters and are trained on extremely large amounts of raw data with huge costs. Lately, OpenAI released

ChatGPT, a chatbot fine-tuned from GPT-3.5 via reinforcement learning from human feedback (RLHF) (Christiano et al., 2017). Large Language Models can be used to explore various different topics, including sentiment classification. As ChatGPT is quite new, we have not found substantial research on the topic of sentiment classification directly related to ChatGPT. Some literature only focuses on the acceptance of the public towards ChatGPT such as exploring the sentiments of ChatGPT early adopters (Haque et al., 2022). However, there exist preliminary studies on the performance of ChatGPT in analyzing sentiment. Wang et al. demonstrated that ChatGPT possesses remarkable zero-shot sentiment classification capabilities and emphasized the effectiveness of ChatGPT in handling polarity shifts and open-domain scenarios (2013).

To the best of our knowledge, we stand at the forefront of research segmenting the four content-related aspects on movie reviews based on the analysis of multiclass sentiment classification. We contribute to the literature by providing a preliminary system to perform aspect segmentation for movie reviews.

3. Methodology

3.1 Description of the Data

For the purpose of the paper, we utilized the dataset **Large Movie Review Dataset v1.0** (Maas et al., 2011), which contains movie reviews along with their associated binary sentiment polarity labels and is intended to serve as a benchmark for sentiment classification. The core dataset contains 50,000 reviews split evenly into 25k train and 25k test sets. The overall distribution of labels is balanced (25k pos and 25k neg). For the supervised learning in sentiment classification part, we used 10% of the training set for validation, which would be 10% of 25,000, which is 2,500 sentences. For the aspect segmentation part, since we are segmenting the four aspects on this movie review dataset and the dataset don't have labels for the four aspects (theater facilities; movie plots, actor & actress performances; movies' special effects and scenes) for each sentence, therefore, we utilized Large Language models to assign aspect labels for next step's supervised learning, meanwhile using human-annotation validation to ensure label correctness.

In the entire collection of the dataset, no more than 30 reviews are allowed for any given movie because reviews for the same movie tend to have correlated ratings. Further, the train and test sets contain a disjoint set of movies, so no significant performance is obtained by memorizing movie-unique terms.

3.2 Multi-Class Sentiment Classification

3.2.1 Feature Extraction, Label Processing

We extracted features using two methods: word frequency, TF-IDF vector, and n-gram (unigram and bi-gram). For word frequency and n-gram, we used the CountVectorizer module from scikit-learn with ngram_range=(1,2) parameter to extract both unigrams and bigrams. For the TF-IDF vector, we used the TfidfVectorizer module from scikit-learn. The feature numbers for these three fields are as follows: top 3000 features from word frequency, top 1000 features from n-gram, and top 3000 features from TF-IDF vector. The overall shape of the combined feature input can be considered as (number of sentences, 7000).

And in the label processing part, we categorized the original sentiment values that served as labels in the raw corpus, in which **negative** ones ranged from 1 to 4 and **positive** ones are ranged from 7 to 10, into four categories with the following sentiment values: “Extra_Neg” ~ [1,2], “Neg” ~ (2,4], “Pos” ~ [7,8], “Extra_Pos” ~ (8,10]. The chart below (Figure 1) shows the distribution of those four different classes.

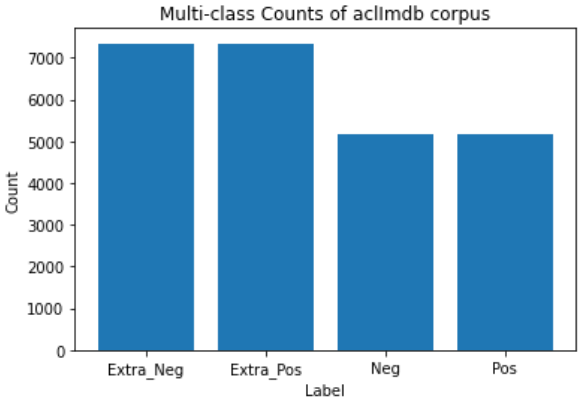

Figure 1: Multi-class Distribution Bar Chart

3.2.2 Baseline

For the baseline model, as a complex model frequently used in image classification rather than text analysis, we decided to test the performance of the Convolutional Neural Network (CNN) with three layers. Our baseline model consists of multiple layers, including convolutional layers, pooling layers, and fully connected layers, and uses backpropagation to adjust the weights of the network during training. Since the convolutional layers are time-consuming, we test our baseline model with 10000-sentence training and 10000-sentence testing.

The first layer was the embedding layer with a dimension of 7000. The second layer was the convolutional layer with 128 filters and a kernel size of 3. The third layer was the max-pooling layer with a pool size of 2. We used a batch size of 32 and trained the model for 10 epochs.

3.2.3 Advanced Models

We also experimented with three advanced models with combined feature input: multinomial Naive Bayes, Support Vector Machines (SVM), Ensemble Method, and Random Forest.

The Multinomial Naive Bayes model is a simple probabilistic classifier commonly used in natural language processing tasks, including sentiment classification. It assumes that the features are independent and the class probabilities are a product of individual feature probabilities, making it fast and efficient. Our model used the MultinomialNB module from scikit-learn.

The Support Vector Machine model is a powerful machine learning algorithm that aims to find the hyperplane between different classes by maximizing the margin distance between the closest data points and the decision boundary, SVM has been widely used in sentiment classification due to its ability to handle high-dimensional data and find non-linear decision boundaries. We used the SVC module with a linear kernel.

The Random Forest model is a popular learning method that constructs a multitude of decision trees at training time and combines their prediction to improve the model’s accuracy and

prevent overfitting. During our model-building process, we used the RandomForestClassifier module with 100 estimators

The Ensemble Method is a powerful machine learning technique that combines the predictions of multiple models to improve the overall accuracy and robustness of the final model, three different models, RandomForestClassifier, GradientBoostingClassifier, and SVC are implemented. The final prediction is made by combining the predictions of these individual models. The voting parameter is set to 'soft', which means that the ensemble model predicts the class label with the highest predicted probability among all the individual models. By combining the strengths of different models, the ensemble method can often achieve better performance than any individual model.

For the development and inspection of the models' performances, we applied the validation set, which is a set of 2,500 randomly selected sentences, before the final inclusion of the test set. The majority of F1 scores of the validation set is satisfactorily ranged from 0.5 to 0.6 which gives us a sight of models and fine-tuning process.

3.3 Large Language Models (ChatGPT)

3.3.1 Why use ChatGPT?

In brief, the large language model (ChatGPT) applied under human annotation quality control aims to **assist movie aspect segmentation** tasks discussed in section § 3.4, for assisting both cosine similarity and supervised machine learning classification approach. Because Large Movie Review Dataset v1.0 (Maas et al., 2011) has only sentiment labels but doesn't contain any movie aspect information, we applied ChatGPT to address data flaws. Two detailed reasons are as follows:

Firstly, although we have movie review training and testing sentences, for the specific cosine similarity approach, labeled sentences are required so that unlabeled movie review sentences can be classified toward. ChatGPT generates required labeled sentences to assist this classification approach and human annotation validation is applied. Sentence-generating details are discussed in § 3.3.2.

Secondly, it generates movie review aspects labels (theater facilities; movie plots, actor & actress performances; movies' special effects and scenes) for supervised machine learning tasks. Although unsupervised machine learning methods like clustering are able to classify movie review sentences by topics, they can only identify one aspect (cluster label) for each movie review. Due to the reality that each movie review sentence may talk about several aspects at the same time, supervised classification becomes a better approach, which is the focus in section §3.4. Hence, aspect labels are generated under human annotation validation to assist high-quality supervised classification. ChatGPT label-generating details are discussed in § 3.3.3.

3.3.2 Sentence Generation

In order to keep generating sentences with ChatGPT, we use ChatGPT API so that we can leverage ChatGPT's natural language processing abilities to generate sentences for us. For the ChatGPT API, we use "gpt-3.5-turbo" model to generate sentences based on our needs. Here's the prompt to tell ChatGPT what to do and a sample output of the generated movie review sentences on the aspect of movie plots:

Prompt: {"role": "system", "content": "You are an AI model that generates unique sentences about movie plots."}, {"role": "user", "content": "Generate 10 new separate sentences. Make sure they are sentences about movie plots."}

Output: That film had a plot that was both complex and engaging.
Since all the movie reviews in the dataset are real human reviews, to make the ChatGPT-generated sentence more "realistic", we fetch and sort the top 1000 most frequent words in the training set and feed it into the generation of sentences. In this way, the generated sentences will be closer to real human movie reviews. We modify the prompt as follows:
Prompt: {"role": "user", "content": "Generate 10 new separate sentences. Make sure they are sentences about movie plots. Make sure every sentence has at least two words in the following word list to generate sentences: {top_train}"} (top_train is a list that contains top 1000 words)
Output: There is no other film that portrays the struggles of young adulthood in such an authentic way.

We can see that indeed, the sample generate sentences seems much closer to real movie reviews, which will be a great help for the following process. 1000 Similar sentences are generated for each of the 4 different aspects and saved in files for future use, which is 4000 sentences in total.

3.3.3 Aspect Determination

In the second part, we need to use ChatGPT to help determine the aspects of sentences and use it as the answer key to evaluate the output of the system. Similarly, ChatGPT API with 'gpt-3.5-turbo' model is used, and here's the sample use for determining the aspect:
Prompt: {"role": "system", "content": "You are an AI model that determines if the paragraph about movie reviews has the following 4 aspects: movie plots; actor & actress performance; movie theatre facility; movie special effects or stage design. Paragraphs seem to talk about the content of the movie should be 'movie plots'. Only when none of the 4 aspects can be determined, then output the singly 'others', others should not be combined with other aspects in the output."}, {"role": "user", "content": "Determine aspects of the paragraph, the output format should only be aspect names, for example, 'movie plots, movie theatre facility': Context: {text}"}
Paragraph: Being the prototype of the classical Errol Flynn adventure movie and having a good story as well as two more brilliant co-stars in Maureen O'Hara (what an exquisite beauty!) and Anthony Quinn, I can only recommend this movie to all those having even the slightest liking for romance and adventure. Hollywood at its best!
Output: ['movie plots', 'actor & actress performance']

We can see that ChatGPT will give us the (nearly) right answer for the case of our study. It determines the aspects for a random of 4000 sentences from the training set. The aspect determination will be used in the evaluation part.

3.3.4 Reliability and Evaluation of ChatGPT

In § 3.3.2, we use ChatGPT to generate aspect sentences and determine aspects of the movie review sentences. We need to know how well ChatGPT performs on the two different tasks. We do this by human annotation. Two annotators will annotate a random of 1000 generated sentences on the 4 aspects and 5% (200) of the aspect determination movie review sentences. The annotators will decide whether the ChatGPT-generated sentences truly belong to the aspect and also decide if ChatGPT-determined aspects truly belong to the real human movie review sentences. The human annotation will serve as the **answer key** to evaluate the performance of ChatGPT in completing the two tasks. Two annotators will decide on the rules for annotating. For instance, for the 'plot' aspect sentence "Although the movie had a few scenes that didn't quite work, this one stands out because of its unique plot." will be labeled as 'plot'

because the sentence meaning emphasizes more on the plot. Similar rules are decided to help make the two annotations compatible. Before comparing ChatGPT's performance to manual annotations, we will calculate Cohen's Kappa (κ) (Fig. 8) to measure the inter-annotator agreement. This will help us understand the reliability of the manual annotations as a benchmark.

We calculate Cohen's Kappa and get the value κ to be **0.812**, which falls to the criteria of 0.81–1.00. This can be interpreted as "almost perfect agreement", indicating that the human annotations we have are compatible and reliable for use.

$$\kappa = \frac{P_o - P_e}{1 - P_e}$$

Figure 8 . Cohen's Kappa

Then we use the human annotation as the answer key to evaluate the performance of ChatGPT. For sentence generation, the F1 score for the first annotation is 0.843 while the second annotation is 0.862. The average F1 score for ChatGPT on generating sentences from 4 aspects is 0.853. For aspect determination, the F1 score for the first annotation is 0.827 while the second annotation is 0.871. The average F1 score for ChatGPT on aspect determination is 0.849. ChatGPT has a great performance in completing the two tasks based on the average F1 scores.

To conclude, since the value of Cohen's Kappa 0.812 reveals that the annotation is reliable and compatible, an F1 score of 0.853, and 0.849 from comparing human annotation and ChatGPT outputs can be interpreted that ChatGPT has a relatively good performance in terms of both precision and recall, but there is still some room for improvement. The good performance of ChatGPT is essential and necessary in this paper because the generated sentences are used for training purposes and aspect determination is used as the evaluation of the model. This will make the result of the paper reliable and reasonable.

3.4 Movie Review Aspect Segmentation

3.4.1 Cosine Similarity Single aspect labeling approach

The first step was to preprocess the data. We started by removing stop words using the nltk.corpus.stopwords.words library and eliminated punctuations using the String module. We also removed numbers from the data and made English words lowercase. During our observation, we found other HTML-like tags in the text that we removed using the regex module. Finally, we used the nltk.stem.PorterStemmer to create a stemmed version of each sentence. This preprocessing method reduces distractions from frequently apparent but not related words' distraction, meanwhile normalizing words to basic forms to help calculate the TF-IDF feature, which gives more statistical meaning. (Fig.2).

$$TF(t,d) = \frac{number\ of\ times\ t\ appears\ in\ d}{total\ number\ of\ terms\ in\ d}$$

$$IDF(t) = log\frac{N}{1+df}$$

$$TF - IDF(t,d) = TF(t,d) * IDF(t)$$

$$\cos\theta = \frac{A \cdot B}{|A||B|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \cdot \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Figure 2: Definition of TF-IDF and Cosine Similarity

Because TF-IDF is one of the most frequently used techniques in measuring word frequency and its appearance amongst documents (Soucy and Mineau, 2005), we first choose it to be the classification feature. For labeled sentences which movies review will be classified toward, they are generated by large language models (ChatGPT) and validated by human annotation discussed in section § 3.3, (to be brief, we call them **aspect sentences**). We calculated the TF-IDF score (Figure 2) for both movie review sentences and aspect sentences based on the definition above. To do this, we took each remaining preprocessed sentence in datasets as the basis and used these words to calculate the TF and IDF feature for all corresponding aspect sentences. This gave us a total of (4000 movie review sentences * 4156 aspect sentence) vectors. Because cosine similarity measures the similarity between two vectors of an inner product space and returns a score for two different documents that share the same representation, it's a proper tool to find documents with similar features (Dehak, 2010). We then calculated Cosine Similarity (Figure 2) between each sentence and its corresponding aspect sentences (1 vs 4156). Finally, we took the average of the cosine similarity scores from each aspect sentence group to assign the aspect label to the current sentence. We repeated this process for all train sentences, and finally, each sentence was assigned an aspect label.

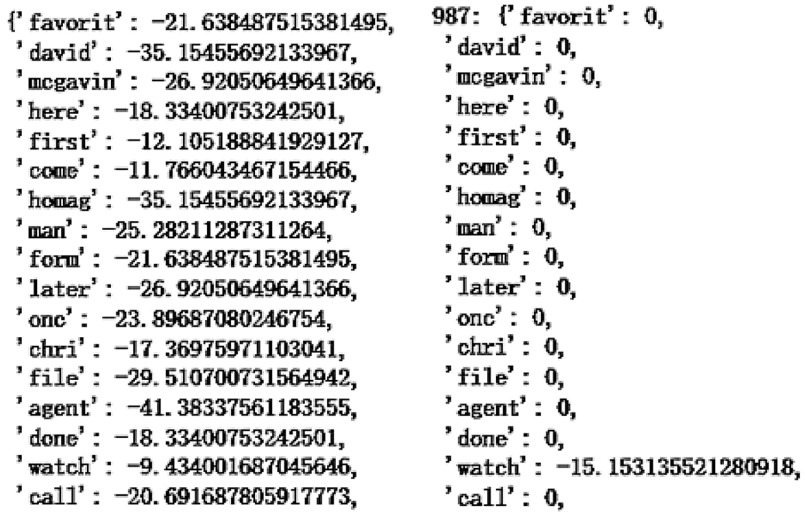

Figure 3: TF-IDF feature random sampled from train sentence contrast with corresponding aspect sentences

However, a problem occurred during this process. Because only a very small fraction of words from a train sentence appears in a single movie aspect sentence, random samples of movie review aspect sentences showed TF-IDF scores almost full of 0s except for only one or two words with positive values (Fig. 3). Then cosine similarity becomes not reliable and even nonsense because movie reviews' aspect labels will be assigned based on just a small portion of word's TF-IDF feature. The poor performance can be seen in the validation process where the F1 score is only around 0.2. To address this issue, we concatenated sentences from each aspect into a paragraph. Each combined passage contained 1039 movie aspect sentences, so it could contain more different words, which reduced the probability of its TF-IDF feature vector being full of 0s. Additionally, combined sentences enable words' appearance probability to be more realistic, mimicking the real distribution of possible words in movie reviews about a certain aspect. We

then calculated the TF-IDF score for prolonged aspect sentences again. This results in a total of (4000*4) vectors. Finally, we applied a cosine similarity algorithm to perform classification, where the aspect with the highest cosine similarity score among the total 4 aspects was assigned as the aspect label to that movie review sentence.

By testing with a test dataset, even though there is improvement in the F1 score, the second method still performs not so well. We consider three reasons. Firstly, cosine similarity algorithms can only assign one aspect label to each movie review sentence at once, but in reality, a movie review sentence can cover several aspects, which may cover both plots and actor performance in a single review. Then, in order to continue to use the cosine similarity approach to assign multi-aspect labels, we should determine if there is a threshold for a certain cosine similarity score to be indicative of a particular movie review sentence that may belong to an aspect category. But it is hard to quantify the decision boundary, which is usually case by case, even though TF-IDF is normalized. The second reason is that compared to human annotation's judgment, cosine similarity may underestimate the similarity of frequent words with other instances of the same word or other words across contexts, which may lead to classification inaccuracy, according to research done by Kaitlyn Zhou et al. (2022). Furthermore, TF-IDF feature vectors neglect word-word and word-context relationships, so we would try another word embedding method.

Hence, in the next subsection, we applied control variable experiments on different algorithms, differently modeled datasets and two different features to solve the above issues.

3.4.2 Machine Learning Multiple Aspect Labeling Approach

In order for machine learning models to learn from feature vectors, we transformed each sentence into a TF-IDF vector of the same length. Specifically, to achieve this, we utilized the "sklearn.feature_extraction.text.TfidfVectorizer()" method to extract the top 10000 frequency terms from the corpus as TF-IDF features while considering both unigram and bigram distributions. Details about the origin and quality of aspect labels are discussed in § 3.3, (to be brief, we call them Y labels). To make the Y label better suited for binary classification, we implemented one-hot encoding. This involved assigning a label of 1 to one aspect if a movie review falls into a certain aspect category, and 0 otherwise (Fig.4). In the training improvement process, due to the potential imbalance between each category, stratified K-fold cross-validation is implemented on each tuned model, and the stability measured by the average cv score.

	original	actor	plot	facility	scene	others
0	Cornel Wilde and three dumbbells search for su...	1	1	1	0	0
1	The limited scenery views were the only saving...	1	1	0	0	0
2	After an undercover mission in Bucharest to di...	1	1	0	0	0
3	I would bet a month's salary "The Magnificent ...	1	1	1	0	0
4	Let's put political correctness aside and just...	1	1	0	0	0

Figure 4: One-hot encoding on multi-class Y label

DummyClassifier is the baseline model, which utilizes the "uniform" strategy to make classifications. It randomly assigns labels based on uniform distribution. This model's prediction accuracy is tested as a baseline, but due to its multi-label classification approach, its f1 score is better than the simple single-label assign method discussed in § 3.3.1.

Random Forest is a tree-based machine learning algorithm that leverages the power of multiple decision trees for making decisions and making voted output. Its mechanism is already discussed in § 3.1.3. Because of the large dataset and huge vector space for this task, instead of a decision tree, random forests' randomness enables more accurate classification. Meanwhile, GridsearchCV is used to optimize model parameters. F1 score on the validation and test dataset both improved.

K nearest neighbor classifier is a supervised machine learning model, which uses proximity to make classifications or predictions about the grouping of an individual data point. We applied this distance based model, because sentences with similar feature values, namely containing the same movie aspect, tend to be close to each other in vector space. It is tuned with GridSearchCV to find the best model parameters for each aspect's classification process. The trained KNN model resulted in further improvements to the validation and test F1 score.

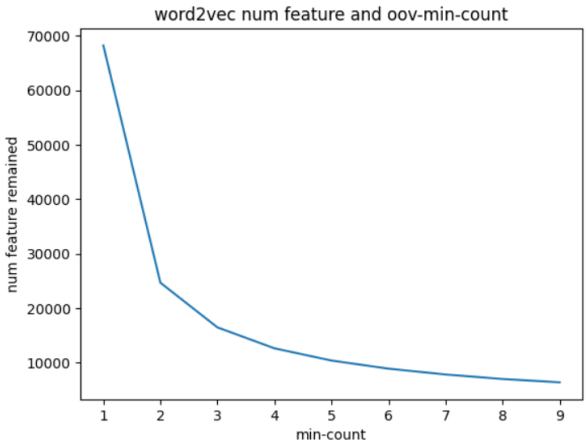

Figure 5: Select number of features to include in Word2Vec embedding

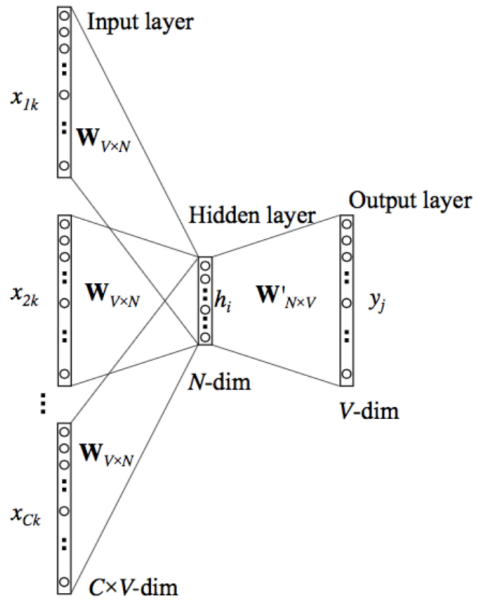

Figure 6: Word2Vec Continuous Bag of Words Neural network mechanism

In this control variable experiment, besides the implementation of different models, another feature selection is also implemented. We experimented with a word embedding method called Word2Vec, which extracts useful information about a word by considering not only words but also their context and neighbors. Due to the possibility of certain words belonging to out-of-vocabulary (OOV) or having a very low frequency, we set the threshold of minimum frequency 7 for a word to be a feature, which greatly removes distractions while keeping frequent non-stopping words. The threshold for min_count has been tested from (1-9), and 7 is selected in reference to the elbow's method. (Fig. 5) For the neural network strategy, we choose Continuous Bag of Words (Cbow) (Fig. 6). The first reason is that the mechanism of Cbow enables it to find the probability of a word occurring in a context, which can generalize over various contexts in which a word can be used. Besides, Cbow trains much faster on large datasets, which suits the situation that a huge amount of different tokens appears in aspect sentence collections. For the remaining parameters, they remained the default. After this preprocessing, the vector space becomes intense, containing more useful information while excluding distractions. At this stage, by feeding vectors into the K-nearest neighbors model, a slightly higher F1 score is generated.

4. Evaluation Metrics

4.1 F1 score

Precision, recall, and F1 score are used as evaluation metrics for both the baseline model and advanced models in sentiment classification tasks. To intuitively explain the evaluation formula shown in Figure 7, precision is the proportion of correctly predicted positive instances among all predicted positive instances. Recall is the proportion of correctly predicted positive instances among all actual positive instances. F1 score is the harmonic mean of precision and recall, which provides a balanced measure of both metrics. We calculated these metrics for each

sentiment class and reported the average across all classes. For F1 score specifically, it is also applied to aspect segmentation tasks, it evaluates the portion of output that's correct and the portion of correct answers to whole answer keys.

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{F1 Score} = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}}$$

Figure 7: Precision, recall, and F1 score formula

4.2 Confusion matrix

The Confusion matrix is another evaluation metric for sentiment classification tasks. After obtaining the best-performing model, we also used the confusion matrix to further interpret the final result. A confusion matrix is a table that summarizes the performance of a classification model by displaying the predicted and actual classes. The confusion matrix provides insights into the strengths and weaknesses of the model and can help identify areas for improvement. By analyzing the confusion matrix, we can determine which classes the model has the most difficulty predicting and adjust our approach accordingly.

5. Results Analysis and Conclusions

5.1 Multi-class Sentiment Classification Results and Conclusions

Based on the test set's evaluation metrics of precision, recall, and F1 score shown in the report table below (Figure 9), we found that although the baseline CNN model had the lowest performance, it still achieved reasonable precision and recall scores. Besides the potential overfitting, CNNs may not be able to capture the semantic relations between words, phrases, and sentences, which could be important for sentiment classification.. Moreover, another possible reason for the poor accuracy of CNN might be the result of data imbalance since "Extra_Neg/Pos" groups obtained a larger corpus than "Neg/Pos" groups in our feature inputs. The advanced models, in general, yield much better outcomes, partially because of the higher adoption of sparse text data or efficient handling of multiple classes. Overall, our findings suggest that the Multinomial Naive Bayes model is a robust approach for this sentiment classification task, which echoes the expectation that multinomial NB is suitable for text classification where the attributes correspond to word occurrences or frequencies within the documents being categorized (Mishra et al., 2016).

Model	Precision	Recall	F1
Baseline(CNN)	0.24	0.30	0.20
Random Forest	0.51	0.53	0.49
SVM	0.55	0.56	0.55
Ensemble Method	0.54	0.56	0.53
Multinomial Navie Bayes	0.58	0.58	0.58

Figure 9: Classification Report Table using F1 Evaluation

The following bar chart (Figure 10) simply depicts the F1 score gap between baseline and advanced models.

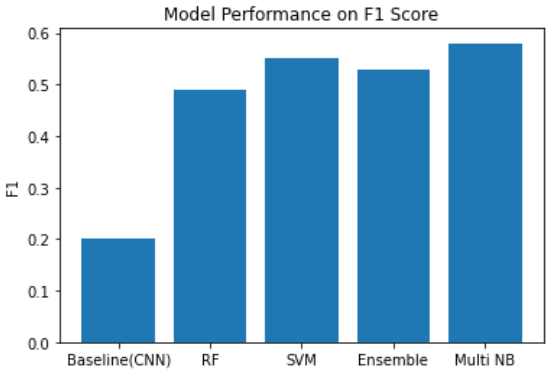

Figure 10: Model Performance Bar Chart

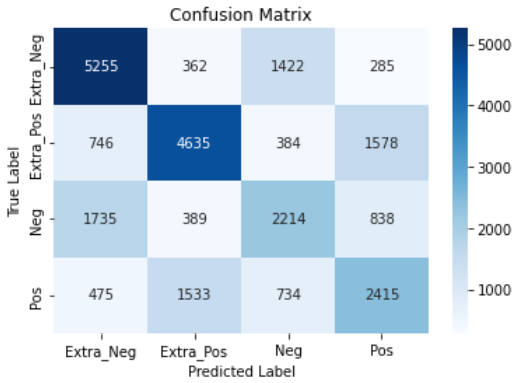

Figure 11: Confusion Matrix on Multinomial NB

In the confusion matrix (Figure 11), one of the potential reasons for the mistaked labeling is that sentences with strong sentimental tags like "Extra_" typically contain more sentimental-relating words, therefore becoming easily distinguishable. The other reason probably is the lack of corpus within "Pos/Neg" compared with "Extra_Pos/Neg", in which the parameter cannot be fully trained and tuned according to the feature inputs.

5.2 Aspect Segmentation Task Result and Conclusions

F1 Score

	Single aspect label output		Multi-aspect label output				
Embedding method	TF-IDF						**Word2vec word embed**
Model	**Cosine Similarity Baseline 1**	**Cosine Similarity**	**Dummy Classifier (Baseline 2)**	**Support Vector Machine**	**Random Forest**	**K-nearest neighbor**	
Raw	0.17						
Prolonged		0.26	0.41	0.71	0.72	0.73	0.73
Raw + top	0.27						
Prolonged +top		0.34	0.49	0.76	0.80	0.81	0.82

Note:

1. Raw indicate raw aspect sentence, which does one-to-one cosine similarity classification. This data modeling strategy doesn't apply to machine learning algorithms.
2. Prolonged indicate prolonged movie aspect sentences
3. Top refers to aspect sentences generated with top frequent words from train data set

In the movie aspect segmentation stage, we implemented control variable experiments by several models on one version of training and testing data, but with different versions of movie aspect datasets as classification reference, which finally output single and multi-class labels. Overall, based on the same data, models that output multi-labels perform significantly better than those which only output single labels, this indicates cosine similarity as a classification tool in multi-class classification may have suffered from some restrictions as mentioned previously, in comparison with machine learning models.

On the data level, model classification using prolonged sentences performs better than single short aspect sentences. To be specific, the longer the target documents are, the wider token coverage they can achieve, and the more real movie-aspect-related word distributions they can mimic. In other words, in situations like aspect segmentation, where explicit words about movie aspects only cover a small portion of individual target documents, while they are still the most important shared feature basis for classification, prolonging and combining relevant movie aspect sentences are important. Also, control variable experiments implemented on the movie aspect dataset, which was generated based on top frequent words from train datasets, enable better classification performance on both cosine similarity and machine learning classification approach. This indicates the importance of classification target data's quality in terms of how many similar aspects keywords they should contain. In brief, the larger the size and more focused content of target documents, the more likely a higher performance classification strategy can be.

On the feature engineering level, using the same machine learning model on word2vec performs better than learning on TF-IDF. For TF-IDF as a feature, a problem has been discussed previously during the experiment that targets document's quality does affect TF-IDF values a lot, and may result in classification based on non-major words. Besides, TF-IDF is based on the bag-of-words model, which means it can't capture the position in context and co-occurrences in different documents. Even though Word2Vec has an advantage over TF-IDF in this situation as it considers concepts, it still suffers from an Out-of-vocabulary issue. Although we tried to select limited words as major features based on frequency, words may potentially suffer from

imbalanced data distribution representing each movie aspect class. But in general, TF-IDF is not bad for this movie dataset, but in comparison, word2vec is a better modeling choice, as explicit descriptions or words about a certain movie aspect in film reviews are not so frequent, which makes interword and contextual information important.

On the model level, in the range of multi-label output, all models perform significantly better than the dummy strategy. In terms of advanced models, both Random Forest Classifier and KNN model performs better than the Support Vector Machine, which indicates higher dimensional feature vectors in this movie review datasets are not well linear separable. Thus ensemble and neighbor models are better choices for this movie dataset.

In conclusion, by identifying movie review aspects and analyzing viewers' sentiments together, understanding of this movie review dataset is advanced.

6. Limitations and Future Work

One limitation is the evaluation of performance using ChatGPT. Even though human annotations greatly help determine the performance of the ChatGPT, we still can't get the exact accuracy of ChatGPT in performing the two tasks since annotators make errors in annotations. In the future, we will have more annotators annotating more sentences and aspects so as to minimize the error as much as possible.

In the sentiment classification, the features combing word frequency, n-grams, and TF-IDF are limited to the token level. Higher architecture like phrase structure or semantic analysis might detect the sentiment more accurately on a sentence-level or paragraph-level and enhance the overall performance tremendously.

As discussed in the previous section, word2vec embedding method suffers from feature selection on imbalanced word distribution issues. Thus, for the aspect segmentation part in this paper, enhanced feature engineering becomes a focus in our future works. We will use other ways like principal components analysis to filter out unnecessary embedding features and set up more reliable thresholds for OOV words exclusion. Also, we will try to visualize top frequency words' distribution among each movie aspect, and see if balanced feature selection can lead to better word embedding-based aspect classification.

What's more, we will continue with this project by testing sentiment classification based on datasets being pre-classified by aspect segmentation. The reason is that sentiment classification based on general movie reviews may not be so accurate, because each movie review contains different topics about different movie aspects, and thus sentimental phrases may be different when talking about different movie aspects. Classifying sentiment based on sentences from similar topics may lead to feature vectors more concentrative and robust. By implementing this future stage, we hope movie review sentiment prediction accuracy can be enhanced by aspect classification.

7. References

Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., & Amodei, D. (2020). Language Models are Few-Shot Learners. arXiv preprint arXiv:2005.14165.

Christiano, P., Leike, J., Brown, T. B., Martic, M., Legg, S., & Amodei, D. (2023). Deep reinforcement learning from human preferences. arXiv preprint arXiv:1706.03741.

Dehak, N., Dehak, R., Glass, J., Reynolds, D. and Kenny, P. (2010). Cosine similarity scoring without score normalization techniques. In Proceedings of Odyssey Speaker and Language Recognition Workshop.

Dumais, S., Platt, J., Heckerman, D. and Sahami, M. (1998). Inductive learning algorithms and representations for text categorization. In Proceedings of the seventh international conference on Information and knowledge management, pages 148-155. ACM.

Han, E. H. and Karypis, G. (2000). Centroid-based document classification: Analysis and experimental results. Principles of Data Mining and Knowledge Discovery, pages116-123

Haque, M. U., Dharmadasa, I., Sworna, Z. T., Rajapakse, R. N., & Ahmad, H. (2022). "I think this is the most disruptive technology": Exploring Sentiments of ChatGPT Early Adopters using Twitter Data. arXiv preprint arXiv:2212.05856.

Kaitlyn Zhou, Kawin Ethayarajh, Dallas Card, and Dan Jurafsky. (2022). Problems with Cosine as a Measure of Embedding Similarity for High Frequency Words. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 401–423, Dublin, Ireland. Association for Computational Linguistics.

Kim, Y. (2014). Convolutional neural networks for sentence classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 1746-1751).

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011). 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). Portland, Oregon, USA: Association for Computational Linguistics. Retrieved from http://www.aclweb.org/anthology/P11-1015

Masood, S. Z., Khan, M. A., Basalamah, A., & Al-Sarem, M. (2018). Sentiment Analysis of Online Reviews Using TF-IDF and Machine Learning Techniques. In 2018 International

Conference on Advances in Computing, Communications and Informatics (ICACCI) (pp. 570-576). Bangalore, India: IEEE. DOI: 10.1109/ICACCI.2018.8554692.

Mishra, T., Yadav, R., & Singh, V. (2016). Sentiment analysis of movie reviews using different classifiers and feature extraction methods. In 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (pp. 1214-1219). IEEE.

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up?: Sentiment classification using machine learning techniques. Proceedings of the ACL-02 conference on Empirical methods in natural language processing, 79-86.

S. Z. Masood, M. A. Khan, A. Basalamah, and M. Al-Sarem, "Sentiment Analysis of Online Reviews Using TF-IDF and Machine Learning Techniques," in 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Bangalore, India, Sep. 2018, pp. 570-576, doi: 10.1109/ICACCI.2018.8554692.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., & Potts, C. (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). https://nlp.stanford.edu/~socherr/EMNLP2013_RNTN.pdf

Soucy, P. and Mineau, G. W. (2005, July). Beyond TFIDF weighting for text categorization in the vector space model. In International Joint Conference on Artificial Intelligence, Vol. 19, pages 1130-1135. Lawrence Erlbaum Associates Ltd

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. arXiv preprint arXiv:1310.4546.

Wang, Z., Xie, Q., Ding, Z., Feng, Y., & Xia, R. (2023). Is ChatGPT a Good Sentiment Analyzer? A Preliminary Study. arXiv preprint arXiv:2304.04339.