**Text

Description automatically generated**

**CS4022 Social Media Mining**

**Individual Project 2:**

***Amazon Software Review Mining***

|  |  |
| --- | --- |
| **By: Tan Ching Fhen** | **Mat ID: U1920787D** |
|  |  |
|  |  |
|  |  |
|  |  |

**1 Introduction:**

The Amazon Review Data [1] is a rich source of review data for companies to perform social media mining. Utilizing social media mining, we can extract many insightful information that can be used to enhance business processes. In this project, I have decided to perform social media mining on *software* reviews because many of the largest companies in the world like Microsoft are driven by software.

Using many social media mining techniques such as topic modelling (Section 3.1), word frequency mining (Section 3.2), sentiment categorization (Section 3.5), review semantics (Section 3.3), keyword extraction (Section 3.3), and information retrieval (Section 3.6), I hope to extract insightful information from *software* reviews, explore better techniques, and uncover potential areas for social media mining in future.

All plots and figures are illustrated in the appendix.

**2.1 Data Description:**

In the Amazon Review Data, under the *small* datasets section, I downloaded the *software* reviewssection as well the meta-data. In each sample, the fields that I utilized are “overall”, “summary”, “reviewText”, “asin” and “brand”. The “overall” field refers to the rating given to the product which ranges from 1 to 5, “asin” field is an identifier, “summary” field is the summary of the review and “reviewText” is the full review. I decided to include the “summary” field because it is much more concise, which I hypothesize that it will lead to better features for sentiment categorization. In Figure 17 and 18, I show a sample of the raw data.

**2.2 Preprocessing:**

With that being said, I will now describe the basic preprocessing steps that I performed to make the data suitable for my social media mining techniques.

First, I merged the review (Figure 17) and metadata (Figure 18) by the “asin” field. I assigned, reviews with ratings 1-2 a negative sentiment, reviews with rating 3 a neutral sentiment and reviews with ratings 4-5 a positive sentiment. Some of these samples are displayed in Figure 16. Furthermore, I balanced the response variable, “sentiment” (Figure 19). Introducing a neutral sentiment and balancing the response classes will make the sentiment categorization task more difficult but retains a lot of information that can tell us, “Why a customer gave neutral feedback?”, and “How can we improve our business processes so that customer satisfaction can improve from neutral to positive?”.

Next, I performed all basic processing steps such as duplicate removal, lower-casing, stop word removal, punctuation removal, lemmatization (*textstem* R package [10]) and encoding of the response variable by the mapping {“NEU”:0, “POS”:1, “NEG”: -1}. Note that depending on the type of mining technique used, the processing step may be different to suit that task. For instance, I performed lemmatization, stop word removal and punctuation removal for *TFIDF* mining (Section 3.2), but not for embedding generation (Section 3.3) because sentence semantics need to be retained for *Transformer* models – these language models inherently understand punctuations, grammar, and meanings of words.

In total, the processed dataset has 4440 samples.

**3 Social Media Mining Techniques:**

**3.1 Topic Modelling:**

Topic modelling is an unsupervised task of assigning hidden topics to textual documents. There are many topic models available, each with their strengths and weaknesses. Perhaps the most state-of-the-art and handy of those is *Top2vec* (R package[2]), which has been found to produce topic words that are much more informative (Figure 25) as found by Dimo Angelov [3]. In addition, in contrast to Latent Dirichlet Allocation (LDA), *top2vec* does not require the number of topics to be specified beforehand. Thus, I will utilize *top2vec* [2] over LDA.

*Top2vec* uses Doc2vec to produce word and review embeddings, then UMAP (a dimensionality reduction technique) is performed, followed by clustering using HDBSCAN. These clusters produced are the hidden topics and the topic words are the word embeddings that are closest to each topic centroid. With that said, in addition to assigning each review the most similar topic, I also computed the dot product similarity to each topic using the embeddings (Figure 26) – this is similar with the topic distributions generated by LDA. This produces much more informative features for sentiment categorization models later (Section 3.5) – having a degree of association to topics is more informative than simple yes or no association.

There were 36 topics found in total, and Figures 8-13 show the topic words for 6 of these topics – the larger the word, the more relevant the word is to that topic. Figure 8-10 are the most important topics while Figure 11-13 are the least important topics (I will describe topic importance later in Section 3.5). The most important topic, topic 35, have topic words like “excellent”. Looking at the distribution of sentiments by topic (Figure 24), we can observe that topic 35 has mostly, positive sentiments. On the other hand, topic 6 (Figure 9), has topic words like “junk” and it has mostly negative sentiments (Figure 24). This means that, the more unbalanced the sentiment distributions for the topic, the more informative the topic (in predicting sentiment) and the corresponding topic words.

With this knowledge, we can also find out, for a particular review, product, or brand, what is its most similar topic (by finding the topic with the highest similarity score in Figure 26)? What is the sentiment distribution towards that topic (Figure 24)? What (topic) words are usually used to describe that review, product, or brand (Figures 8-10)? For instance, for sample 49 (Figure 22) about a Microsoft product, its most similar topic is topic 6, which has many negative topic words (Figure 9). For sample 828 (Figure 23) about an Apple product, its most similar topic is topic 35, which has positive sentiments and associated words (Figure 8).

**3.2 TFIDF Mining:**

Term Frequency and Inverse Document Frequency (TFIDF) is a numerical statistic that reflects how important a word is in the review corpus. I computed TFIDF features using the text2vec R package [4]. However, TFIDF features are very sparse, and high in dimensionality. This makes learning more difficult especially for certain models due to the curse of dimensionality. Thus, I try to reduce this effect by taking only the most frequent topic words found in Section 3.1, which tend to be more informative and higher in variance (this reduced the total number of TFIDF features by over 10000).

Nonetheless, we can still observe that TFIDF are poor predictors of sentiment (Figure 2) with very low feature importance (I will describe feature importance in Section 3.5). Figure 3 shows some of the most important TFIDF features like “easy” and “great”, while Figure 4 shows some of the least important TFIDF features.

Since TFIDF features are generally poorer predictors (Figure 2), I only used unigrams, because more n-grams likely would not make significant contributions.

**3.3 Review Embeddings:**

Semantic embeddings are vectors that represent meanings. Specifically, I will use contextualized embeddings as additional features for my sentiment categorization model. Contextualized embeddings are dynamic; depending on context, the same words can be assigned different embeddings. This accounts for situations where the same words have multiple meanings.

To generate contextualized embeddings, I made use of *Transformers* (R package [5]), that are state-of-the-art language models that utilize self-attention mechanism. Specifically, I generated embeddings for the “summary” field and “reviewText” field separately. The reason being that the summaries are generally more concise and less noisy, thus I hypothesize that they would produce better embeddings for sentiment categorization in Section 3.5.

Because the “reviewText” field tend to be very long and noisy, I used Rapid Automatic Keyword Extraction (RAKE) [6] to extract important keywords. These keywords are passed through the *Transformer* models to generate another set of keyword embeddings. The motivation here is to reduce the impact of noise and hopefully improve embedding quality.

In figure 3, we can observe that the summary embeddings are most important in predicting sentiment i.e., informative, while the review and keyword embeddings are less so (possibly because the keywords generated by RAKE are still quite noisy, with a lot of redundant keywords). This validates my hypothesis that, a summary of a review is a better source of raw data compared to a full review.

**3.4 Additional Feature Engineering:**

In this section, I will describe some additional features that I generated manually, which may help improve sentiment categorization.

First, I extract the number of words and number of uppercase words (I group these features as “other\_features” in Figure 2). The motivation is that the number of uppercase words or words may have correlation to the sentiment of the review.

Next, I extract the number of punctuations. Punctuations such as exclamation marks may correlate with customer sentiment, thus it’s useful to retain such information. In figure 5, we can observe the feature importance for punctuation; commas, full stops and exclamation appear to be somewhat informative.

Lastly, I also utilized sentiment lexicons. Specifically, I use WKWSCI Sentiment Lexicon Version 1.1 [7]. For every review, I acquired the sentiment scores for all the words that were also in the sentiment lexicon, then computed the summation and normalized by the length of the review. This means that if the total sentiment lexicon score is positive, there are more positive sentiment words than negative sentiment words in the review. In addition, for reasons stated before, I computed the total sentiment lexicon scores for “summary” and “reviewText” field separately. Intriguingly, sentiment lexicon scores were the best features in predicting sentiment category, far more so than all other features (Figure 1 and 2).

From Figure 2, we can observe that both these features are effective in predicting sentiment. This shows that if we have some domain knowledge or some kind pre-constructed information (such as sentiment lexicons), it is effective to leverage them.

**3.5 Sentiment Categorization:**

This chapter will describe the models trained for sentiment categorization and how I acquired the feature importance scores.

There are three sentiment categories, POS, NEG, and NEU, which are equally distributed (Figure 19) as described in Section 2.2. I split this dataset into training and test sets by the ratio 7:3.

I trained 5 types of models namely: lightgbm [8], random forests, extra trees, naïve bayes, and support vector machines (SVM trained for multi-class classification using one-to-one scheme) [9]. For all models, I used the default hyperparameters for fair comparison.

In total there were 2223 input features, consisting of “sentiment\_lexicon” features, “other\_features”, “topic\_similarity” features, embedding features, punctuation features and TFIDF features (Figure 2). However, due to training time concerns, I only used the top 300 features with the highest importance scores. These feature importance scores (Figures 1-7, 14) were derived by training lightgbm [8] – the higher the feature importance, the better it is in predicting sentiment.

For SVM, I also trained a reduced model, “SVM-reduced”, that only uses the best features, “sentiment\_lexicon\_score” features. I hypothesized that the large number of features might have created a lot of noise which tends to cause issues for SVM learning.

Figure 15 shows the accuracy, precision, recall and f1 scores. We can observe that lightgbm, performed the best. Thus, I performed further hyperparameter tuning using grid search (lightgbm-tuned). Also, we can observe that “SVM-reduced” performs better than SVM even though it uses only 2 features (compared to 300).

From the confusion matrix and statistics (Figure 27 and 28), we can observe that neutral sentiment (NEU) is more difficult to predict – the sensitivity and accuracy is much lower for this class. To give an example, figure 29 shows a sample that is neutral, but the best model predicts a negative sentiment – which is understandable because the word “leery” has a negative connotation. On the other hand, figure 30 shows a sample that is neutral, but the best model predicts a positive sentiment – which is also understandable because of the words “good” and “easy”.

**3.6 Information Retrieval**

Lastly, this chapter will describe how I utilized the *top2vec* model and embeddings (Section 3.1) to perform some interesting semantic retrieval tasks. For instance, given a *new* review, what is the most similar topic? Given some keywords, what is the most similar review? When the number of reviews and topics get very large, these tasks will have some utility, thus is worth exploring.

Given a *new* review that was not found in the dataset in Section 2.2, we can efficiently assign it to an existing topic without having to retrain the topic model by doing a semantic retrieval. In detail, the *top2vec* model first assigns the unseen review, an embedding **r** of size 1 x 300. I performed matrix multiplication with the existing topic embedding matrix **TT** of size 300 x 36 to produce a vector of similarity scores. The most similar topic for the review shown below was topic 33 (Figure 32), which happens to be about video editing:

I am a total amateur when it comes to editing but needed to try something for editing the photos I take for my business. This software was a quick and easy download. Once downloaded, I was able to start using it pretty quickly, the learning curve was one that I could understand without too much trial and error. I take many photos of different items in the same location, so when I need to edit, it's generally the same thing which needs editing, this software is perfect for that. Now I can fix multiple photos at one time, saving a lot of time! The price is great if you're not sure about the program, not a huge expense like many others out there. For me, and my level of expertise, or lack of it, this program fits the bill!

Next, given a short query, “useful tool to compress file”, I can retrieve a similar *review* from the dataset using the same steps. First, compute the query embedding **q** using the *top2vec* model and perform matrix multiplication with the review matrix **RT** of size 300 x 4440. This produces a vector of scores, and the most similar review happens to be one about win-zip (Figure 31).

**4 Discussion:**

This section will describe some further learning points, and insights found that can be used to improve business processes and mining tasks in future.

From figure 2, we can observe that TFIDF features are least useful in predicting sentiment likely because of the sparsity and high dimensionality. The number words and uppercased words (“other\_features”), performed surprisingly well. This shows that, applying our domain knowledge to manually extract useful features can be quite effective. “Topic\_similarity” features (Figure 26) also performed quite well. Sentiment lexicon features are surprisingly effective as features for sentiment categorization. With that said, future mining tasks can focus on the best features found and reduce development time.

Furthermore, based on my findings (Figure 1 and 2), we can observe that the summaries of reviews reveal the most information of customer sentiment and satisfaction. Therefore, when collecting customer reviews in future, it will be effective to ask for a short summary of the review if not already done so.

In addition, based on my experiments in Section 3.1 and 3.6, we can observe the effectiveness of topic modelling using *top2vec*. Using the embeddings generated, we can perform interesting tasks like semantic information retrieval like I’ve shown. In addition, we can extract many informative topic words as I’ve shown in Figures 8-13.

Finally, I have shown that we can use topic modelling in conjunction with sentiment categorization to improve business goals. Let’s say we want to find out how we can improve customer satisfaction from neutral to positive or negative to neutral, we could do the following: First, observe the sentiment distribution of topics (Figure 24), we find that topic 6 has a high proportion of negative sentiment while topic 23 has a high distribution of neutral sentiment. Next, bring out the topic words from those topic (Examples in Figures 8-13) – this will summarize the key areas to improve our software product. Finally, bring these insights to the relevant teams, where they can focus on solving those problems that customers are facing.

**5 Conclusion:**

To conclude, I’ve made use of many social media mining techniques namely: topic modelling, sentiment categorization, TFIDF, sentiment lexicons, contextualized embeddings, information retrieval, and feature engineering.

In future work, we can perform more in-depth analysis of the topics and topic words. We could develop a dashboard that summarizes the topic words associated to each brand, making analysis easier. Furthermore, we could automatically or manually assign sentiments to the topic words to create a *software* *domain specific sentiment lexicon*; for topics like 6 and 35 that have very unbalanced sentiment distribution (Figure 24), we can reliably assign the most relevant topic words like “junk”, “crash” and “reinstall” (Figure 9) with high negative sentiment score, while “excellent”, “paint-shop” and “exposure” can be assigned high positive sentiment score. Since we learnt earlier that sentiment lexicons make very good features, this can have some utility.

**6 Appendix:**

Here, I append all plots and figures generated from R packages such as *ggplot*, *wordcloud2*, and *caret*.

Chart, bar chart

Description automatically generated

Figure - Top 20 Most Important Features

Chart

Description automatically generated

Figure - Mean Feature Importance by Feature Types

Chart, funnel chart

Description automatically generated

Figure - Top 20 Most Important TFIDF word features

Chart

Description automatically generated

Figure - Top 10 Least Important Features

Chart

Description automatically generated

Figure - Feature Importance of Punctuations

Chart, funnel chart

Description automatically generated

Figure - Topic Similarity Feature Importance

Chart

Description automatically generated

Figure - Feature Importance - Sentiment Lexicon Scores and Engineered Features

Text

Description automatically generated

Figure - Topic 35 Word cloudText

Description automatically generated

Figure - Topic 6 Word cloud

Text

Description automatically generated

Figure - Topic 12 Word Cloud

Text

Description automatically generated

Figure - Topic 2 Word Cloud

Text

Description automatically generated

Figure - Topic 4 Word Cloud

Text

Description automatically generated

Figure - Topic 16 Word Cloud

Graphical user interface

Description automatically generated with low confidence

Figure - Top 300 Features

Chart, bar chart

Description automatically generated

Figure - Performance Comparison on Test Set

A picture containing graphical user interface

Description automatically generated

Figure - Sentiment Categorization Dataset

Text

Description automatically generated with medium confidence

Figure - Raw Review Sample 1 (Extraneous Fields Removed)



Figure - Raw Metadata Sample 1 (extraneous fields removed)

Chart, bar chart

Description automatically generated

Figure - Response Class Distribution

Chart, bar chart

Description automatically generated

Figure - Sentiment Distribution (Top 10 most frequent brands)

Chart

Description automatically generated

Figure - Distribution of Topics

A picture containing text

Description automatically generated

Figure - Processed Sample 49 (Most Similar Topic: 6)

Text

Description automatically generated

Figure - Processed Sample 828 (Most similar topic: 35)

Chart

Description automatically generated

Figure - Sentiment Distribution of Topics

Chart, line chart

Description automatically generated

Figure - Informativeness of Topic Words (Figure taken from Dimo Angelov [3])

Graphical user interface

Description automatically generated

Figure - Topic Similarity Features – similarity score of each review to each of the 36 topics

Table

Description automatically generated with medium confidence

Figure – Caret Confusion Matrix – LightGBM predictions

Table

Description automatically generated

Figure - Caret Confusion Matrix Statistics - LightGBM predictions

Text

Description automatically generated

Figure - Sample 1378 - Predicted Negative

A picture containing text

Description automatically generated

Figure - Sample 1621 - Predicted Positive

Text

Description automatically generated with low confidence

Figure - Top result - querying documents

Text

Description automatically generated

Figure 32 - predicting topic for new review

**7 References:**

Here, I append all the resources I utilized in this social media mining project.

[1] Data Source: [Amazon review data (nijianmo.github.io)](https://nijianmo.github.io/amazon/index.html)

[2] R package for top2vec: [R: Distributed Representations of Topics (r-project.org)](https://search.r-project.org/CRAN/refmans/doc2vec/html/top2vec.html)

[3] Top2vec research paper: [[2008.09470] Top2Vec: Distributed Representations of Topics (arxiv.org)](https://arxiv.org/abs/2008.09470)

[4] R package for generating TFIDFs: [Term Frequency and Inverse Document Frequency (tf-idf) Using Tidy Data Principles (r-project.org)](https://cran.r-project.org/web/packages/tidytext/vignettes/tf_idf.html)

[5] R package for generates embeddings with Transformers: [RStudio AI Blog: State-of-the-art NLP models from R](https://blogs.rstudio.com/ai/posts/2020-07-30-state-of-the-art-nlp-models-from-r/)

[6] R package for keyword extraction: [R: Keyword identification using Rapid Automatic Keyword... (r-project.org)](https://search.r-project.org/CRAN/refmans/udpipe/html/keywords_rake.html)

[7] Download of sentiment lexicons: [WKWSCI Sentiment Lexicon v1.1 available for download | Chris Khoo (ntu.edu.sg)](https://blogs.ntu.edu.sg/chriskhoo/2017/07/wkwsci-sentiment-lexicon-v1-1-available-for-download/)

[8] R package for training LightGBM [Light Gradient Boosting Machine • lightgbm](https://lightgbm.readthedocs.io/en/latest/R/index.html)

[9] R package for training random forests, SVMs, Naïve bayes models: [train function - RDocumentation](https://www.rdocumentation.org/packages/caret/versions/4.47/topics/train)

[10] R package for lemmatization: [lemmatize\_words: Lemmatize a Vector of Words in textstem: Tools for Stemming and Lemmatizing Text (rdrr.io)](https://rdrr.io/cran/textstem/man/lemmatize_words.html)

**END**