DeGatto: A Sentiment Analysis Framework for E-Commerce

Aine Drelingyte, Axelle Gapin, Omar AbedelKader, Danyl Shkrebko, Elie Dina, Joseph Jreije, Quentin Herbin

Université de Lorraine, IDMC

Abstract

Sentiment analysis (SA) is a process of identifying and categorizing views and opinions expressed in the text by examining data. This paper will summarize the process of Sentiment Analysis and provide the methods and results of the most recent studies. It will introduce the SA project "DeGatto" and provide an argumentative reasoning for the chosen NLP, DL and ML models and methods. The content will describe the progress, timeline and upcoming goals of the project as well as recently made technical changes.

1 Introduction

The interest in others' opinions is thought to be as old as verbal communication itself (Mäntylä et al., 2016). The earliest examples of trying to discern others' dissent can be found in times as early as Ancient Greece or between 475 and 221 B.C.E., during the Warring States period in China when "The Art of War" is thought to be written (Mäntylä et al., 2016). "The Art of War" features a chapter regarding the recruiting and betrayal of spies, whilst the politics and democracy in Ancient Greece meant a constant interest in upholding others' opinion before the time of elections e.g., the elections of generals (Evans and Tougher, 2022). It was not until the mid-2000s that modern SA emerged, and it focused on product reviews available on the internet (Mäntylä et al., 2016).

Nowadays, SA has become a relevant research topic in many fields such as Computational Linguistics (Berend and Farkas, 2008), business (Tamrakar et al., 2020), politics (Wang et al., 2014) and more. According to research, there were approximately 7000 research papers on SA released by 2016 with the number readily growing in recent years (Mäntylä et al., 2016; Nigam et al., 2018).

This paper will focus on the SA usage in women's apparel marketing and perform a sentence-level and aspect-level analysis of an E-Commerce corpus using different machine learning algorithms and hyperparameters. The goal of this paper is to choose the most suitable models and hyperparameters for the project "DeGatto" introduced later on. The research questions that are sought to be answered are as follows: which models will perform the best on an aspect-level in comparison to a sentence-level analysis on the same corpus and which hyperparameters will provide the best results for each model.

2 Background in Sentiment Analysis

2.1 Process of Sentiment Analysis

The process of SA consists of five steps; those are data collection, text pre-processing (data cleaning), sentiment detection, sentiment classification, and output presentation (Aqlan et al., 2019). This process is visualized in Figure 1. The majority of research in this area focuses on detecting polarity in opinions, that can be classified as positive, negative, and neutral (Almashraee et al., 2016). Moreover, SA can be performed at different levels: document-level which provides an overall sentiment of a document, sentence-level which results in an overall sentence sentiment and an aspect-level when the opinion is distinguished by a polarity and a target (Behdenna et al., 2018).

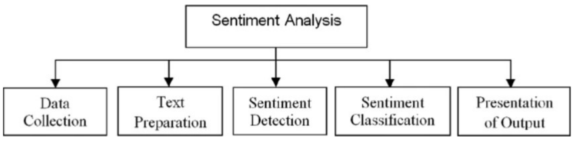

Figure 1: Sketch of sentiment classification process, from Aqlan et al., 2019.

2.2 Sentiment Classification

Sentiment classification model selection in accordance to the corpus that is being used is one of the most important steps to influence the accuracy of

SA. The multiple available sentiment classification models and approaches are summarized in Figure 2. According to Aqlan et al., 2019, the most popular models in sentiment classification presently, are Naive Bayes techniques and support vector machines (SVMs) that have been found to improve the accuracy of previous studies such as a study conducted on tweets by Gael et al., 2016.

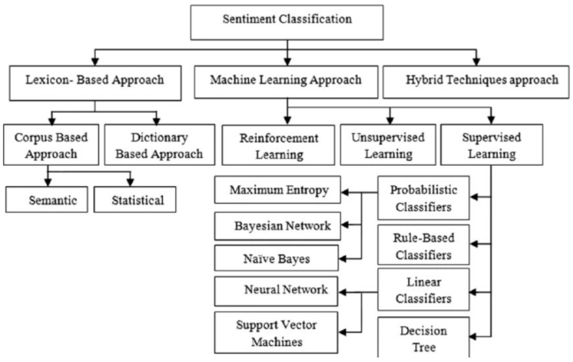

Figure 2: Sketch of sentiment classification models, from (Aqlan et al., 2019).

2.3 Recent Studies

Some recent studies have focused on developing new methods for sentiment analysis, such as using deep learning models like recurrent neural networks (RNNs) or transformer networks, while others have focused on improving existing methods or applying sentiment analysis to new domains. Sentiment analysis is a rapidly evolving field, with new methods and models being proposed all the time.

A previous research paper led in 2017 presented a deep learning model for sentiment analysis of Twitter data called "SA-LSTM", which uses a combination of convolutional neural networks (CNNs) and long short-term memory (LSTM) networks to extract features from tweets and classify them as positive, negative, or neutral. (Hassan et al., 2017) The authors compare the performance of their model to several other state-of-the-art models, such as Naive Bayes, SVM and RNNs on four publicly available Twitter datasets. They show that the SA-LSTM model outperforms the other models in terms of accuracy, F1-score and AUC-ROC. The authors also evaluate the impact of different embedding techniques (such as word2vec, Glove and Fasttext) and the different architectures and parameters of the model on the final performance. They found that pre-trained embeddings improved the overall performance and a combination of word2vec and Glove embeddings performed the best. This study's main contribution is a new model for sentiment analysis of Twitter data, SA-LSTM, that combines CNN and LSTM networks. Furthermore, the researchers evaluate the performance of different embeddings and different architectures and parameters on the final performance. Their findings are that pre-trained embeddings have a positive impact on the overall performance and that a combination of word2vec and Glove embeddings performed the best.

A few years later, in a study on e-commerce product reviews, a new model for sentiment analysis is defined. (Yang et al., 2020) The model is called SLCABG, and combines the use of a sentiment lexicon with a convolutional neural network (CNN) and an attention-based bidirectional gated recurrent unit (BiGRU) to enhance the sentiment features of the reviews and extract the main sentiment features and context features in the reviews. The authors assert that the model overcomes the shortcomings of the existing sentiment analysis models of product reviews. SLCABG model also uses an attention mechanism to weigh and classify the sentiment features. The data used to train and test the model was collected from a Chinese e-commerce website and is based on Chinese language. The data set is large and can be widely used in Chinese sentiment analysis. The experimental results show that the model can effectively improve the performance of text sentiment analysis.

In 2021, a paper describes another new method for sentiment analysis using an attention-based bidirectional convolutional neural network (CNN) and recurrent neural network (RNN) architecture called "Att-Bi-CNN-RNN". (Basiri et al., 2021) The authors used attention mechanism which allows the model to focus on different parts of the input sentence while classifying it as positive, negative or neutral. They applied this method to a Twitter dataset and compared the results with other state-of-the-art models. They found that their method achieved the best performance in terms of accuracy, precision and recall. In more detail, they use a bidirectional RNN to capture the context-sensitive information, as well as a CNN to capture the local features of the text and combine these features through an attention mechanism to make the final decision.

In a paper that is to be published in March 2023, the field of Multimodal Sentiment Analysis (MSA) in artificial intelligence and natural language processing is examined. (Gandhi et al., 2023) In the abstract, the authors explain that there is a growing demand for automating the analysis of user sentiment towards products or services and that opinions are increasingly being shared online in the form of videos as well as text. The authors state that MSA, which combines multiple modalities, has become an important research area and describes recent advancements in machine learning and deep learning that are used at various stages, including multimodal feature extraction and fusion and sentiment polarity detection, to minimize error rate and improve performance. This study will also propose several interdisciplinary applications and future research directions.

2.4 Challenges in Sentiment Analysis

As the number of studies aiming at effectual SA is increasing, target-dependent opinion mining, or in other words, aspect-level analysis is becoming a more widely studied topic. Nevertheless, the two or more features of opinionated texts render a target-dependent analysis more difficult than a sentence or a document-level analysis as the former two tend to assume that each sentence expresses a definite opinion on one target only (Szabó et al., 2016). Hence, the same methods used for sentence and aspect-level analysis may yield different results.

There are multiple reasons as to why such difficulty might be observed, some of which are word polarity and sentiment shifters (Szabó et al., 2016). Word polarity refers to words carrying a different sentiment based on the context. For instance, *The waiting time was long* and *The dress was long* might carry a different sentiment or a different sentiment weight. Whilst the first sentence expresses a negative opinion, the second one does not indicate any clear feelings towards the long dress, so here *long* might be a good or a bad thing (Szabó et al., 2016). Sentiment shifters, however, refer to a change in the target's sentiment caused by structures such as negation. For example, words like *although, but, even though* tend to shift the sentiment or the weight of the sentiment in a sentence. A good example of this would be *Although the color is nice, it does not match the product's description*. Additionally, it poses even more of a challenge considering that even though conjunctions like *not only, but also* contain negation, they do not typically change the sentiment orientation (Szabó et al., 2016).

2.5 Algorithms

Generally, considering the aforementioned problems some models such as Naive Bayes could be predicted to underperform on an aspect-level analysis as it works under the assumption that the analyzed features are independent and not related to each other (Zaidi et al., 2013). Nonetheless, some studies have successfully obtained an accuracy of 70% and more (Tamrakar et al., 2020). Such results could be partially attributed to additional techniques such as POS[1] tagging.

Despite that, according to Aqlan et al., 2019, the most popular models in sentiment classification presently are Naive Bayes techniques and Support Vector Machine (SVM) as they are efficient and rather easily implemented in comparison to some other models such as LSTM which although more complex, was found to provide good results in SA (Wibawa et al., 2019; Shinde et al., 2021; Bilen and Horasan, 2022). In addition, Naive Bayes and SVM have been found to improve the accuracy of previous studies such as a study conducted on tweets by Goel et al., 2016.

3 Project “DeGatto”

The project "DeGatto" is centered on adapting the SA in women's apparel marketing such as dresses, skirts, blouses, etc. It is an industry-focused project which aims to support E-commerce companies and customers by providing a concise analysis of their product feedback data. The final system enables the users to upload an Excel or a CSV file into the system which will then generate the following results: the percentage of negative, positive and neutral comments overall, and in four different aspects: material, size, design, and comfort. Regarding the analysis of the overall sentiment, a sentence-level analysis of the corpus is required, on the other hand, in order to provide a sentiment for each of the four aforementioned categories an aspect-level analysis is needed. Hence, the necessity of identifying the appropriate models and parameters for each level.

The corpus that the study uses was sourced from kaggle.com. Kaggle is an online community of data scientists and machine learning practitioners and contains an abundance of datasets. The corpus that was sourced is a Women's Clothing E-Commerce dataset revolving around the reviews

[1] Part of Speech (POS) tagging is a categorization of word classes, such as verbs, nouns, adverbs, and etc. (Usop et al., 2017).

written by customers and contains approximately 23 thousand sentences and 251 thousand tokens with 10 attributes of which only 4 (clothing ID, recommendation score, review text and rating) were kept[2]. The size of the annotated corpus is 2356 sentences that were randomly selected. The final annotated corpus at an aspect-level contains 672 comfort, 175 longevity, 1673 size, 889 design, and 1080 material sentences.

4 Visualisation Tool

In order to give the end-user the ability to visualise the analysis of their product feedback data, a website was created. This website allows the users to upload an excel or csv file containing product reviews. After processing this data the tool gives the option to choose between two types of graphs, a bar chart and a pie chart. The possibility to select different levels of analyses (sentence and aspect levels) is also present with the ability to download the graphs. If the visualisation method chosen is a pie chart the choice to select which sentiments are portrayed in the chart is available. To build the website, ReactJS was used for the front-end and NodeJS for the back-end, both are free and open-source JavaScript libraries. Several libraries were used to complete the website, for instance "chart.js" was the one employed to make the charts, and to implement an upload component "react-filepond" was used. To make the tool accessible on different devices and screen sizes, the decision of developing a website instead of a mobile application was made.

5 Text Preprocessing and Manual Annotation

Prior to annotation, the review text data has been cleaned by upper-casing and the removal of punctuation, duplicates and manually selected stop-words in consideration of keeping the negation words such as *but, although, even though, not*. To strengthen the anonymization, the corpus has been manually checked for location and identity clues that have been removed. For instance, *"I live in hawaii [...]"*. In this comment, the reference to location is ‘Hawaii’ and it has been nulled out or, in other words, removed from the comment.

The corpus has been annotated manually and there are several reasons for such a decision. Firstly, the four aspects at the aspect-level annotation, complex sentences, the abundance of grammar errors and abbreviations are all likely to affect an automatic annotation. Additionally, the aforementioned problems with an aspect-level annotation such as sentiment shifters and word polarity are also likely to affect the automatic annotations negatively. Furthermore, the research conducted by Atteveldt et al., 2021 on the validity of different approaches to corpus annotation in sentiment analyses, concluded that the best performance is still attained with trained human annotators.

In terms of aspect-level annotation the sentences were annotated in what was called sentence fragments that differed by target features e.g. *I like the size but the colour is ugly* contains two targets that are size and colour both of which were assigned a separate sentiment. Constructions like negation, sentiment shifters and word polarity were considered in cases when these structures focused on the same target e.g. *The length is nice but I expected it to be longer*.

Although the manual corpus annotation is a labour-intensive process, it tends to yield good results (Atteveldt et al., 2021), nevertheless, neither manual nor automated annotation is flawless, and both carry a risk of bias (Lavid, 2010). To minimize the risk of bias in the corpus, each sentence has been annotated by two annotators whose disagreement led to discussion before the final sentiment assignment. Moreover, the annotators were provided with clear annotation guidelines constructed after much consideration to ensure a uniform annotation. To evaluate the validity of the corpus, inter-annotator agreement measures were used and the Cohen’s kappa coefficient was computed separately for each aspect and for the sentence-level annotations.

The Cohen’s kappa is a statistical test that represents the degree of accuracy and corpus validity in a statistical classification (Atteveldt et al., 2021). Atteveldt et al., 2021 recommends at least 143 annotations for determining intercoder reliability, hence, the scores were determined using 150 annotations for each aspect. The kappa coefficients for the sentiment classification of the corpus are as follows: 0.82 (material), 0.86 (comfort), 0.86 (design), 0.62 (size) and 0.82 (sentence-level) which implies an almost perfect agreement on Cohen’s scale in all cases except for size which resulted in a substantial agreement. Henceforth, the corpus

[2] Click here to access the unannotated corpus used in this study.

annotation is valid to use for the future studies.

6 NLP Models and Methods

Sentiment classification model selection in accordance with the corpus that is being used is one of the most important steps to influence the accuracy of the SA. The models that will be used in this paper are the ones that were found to be the most popular amongst the kaggle studies on the same corpus and yielded the best results on the sentence-level analysis (Duran, 2021; Yasser, 2021; Anonymous, 2022; Luna, 2019). Based on the previous studies the models chosen for the project were LSTM, Support Vector Machine (SVM), Logistic Regression (LR), and the Multinomial Naive Bayes (MNB) model of which BiLSTM emerged as the one with the best results on average for Sentence-level (0.94% accuracy) (Yasser, 2021). Based on the restrictions on Naive Bayes in the aspect-level analysis that were mentioned before, we suggest that this model will yield the worst results while BiLSTM with no such foreseen restrictions will give the best results on aspect-level and sentiment-level analyses.

Because of the imbalanced dataset that we had, and to be able to minimize the bias and overfitting risk, a resampling of the sentiments was performed. Hence, when the models were tested and the hyperparameters were applied, accuracy and precision scores carried a very small difference.

6.1 Hyperparameters and test size

Hyperparameters are machine learning parameters whose values are chosen before a learning algorithm is trained. It plays an essential role in the fitting of supervised learning algorithms. Hyperparameters should be selected carefully before fitting the models to a data set, as it is computationally expensive to tune all tunable hyperparameters simultaneously, especially when it comes to large datasets (Jin, 2022). The hyperparameters that we applied to our models are listed in the results and discussion section.

In the study of Vikash Signh which was conducted on the impact of train/test sample size of the machine learning performance in cardiovascular imaging, different test-sizes were explored as the means to improve performance. In accordance to this, we have trialed several test sizes as well.

The following four subsections will provide short introductions into the models used in this study.

6.2 Naive Bayes

Naive Bayes Classifier is one of the simplest and most effective classification algorithms which helps in building fast machine learning models that can make efficient predictions (Wibawa et al., 2019). It is a probabilistic classifier, which means it predicts on the basis of the probability of an object. The Naive refers to the assumption of independence of occurrences between two features. There are different implementations of Naive Bayes when it comes to ML with Python. The one used in this paper is Multinomial Naive Bayes (MNB) which, as different research shows, is often used in sentiment classification tasks(Atanassov and Tomova, 2019).

The hyperparameter tuned was alpha, which is an additive smoothing parameter used to resolve the problem of zero probability in Naive Bayes models. It had been proven that smoothing methods are able to significantly improve the classification performance of Naive Bayes (Yuan et al., 2012).

6.3 Support Vector Machine

Support Vector Machine (SVM) is a set of supervised machine learning methods that can be used for both classification and regression (Pedregosa et al., 2012). Additionally, they are particular linear classifiers which are based on the margin maximisation principle (Adankon and Cheriet, 2009). The objective of the SVM algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points (Jalal and Ezzedine, 2020).

The hyperparameters optimized for this model were C and max-iterations. The C parameter represents the regularization parameter which serves as a degree of avoidance of misclassification. Additionally, max-iterations is the maximum number of iterations to be run for the solver to converge, in other words, how many times the parameters are updated (Pedregosa et al., 2012). The performance of the model was improved in past studies using these hyperparameters (Elgeldawi et al., 2021). Additionally, max-iterations was used to be able to overcome the problem of over-fitting.

6.4 Logistic Regression

Logistic Regression is a statistical supervised learning model that is used for both binary and multi-class classification (Farhadloo and Rolland, 2016).

The hyperparameters tuned for this model are similar to the ones used in LinearSVC. Based on the performance of different studies, tuning these specific hyperparameters produced a better accuracy (Elgeldawi et al., 2021).

6.5 Bidirectional LSTM

LSTM is a deep learning model based on Recurrent Neural Network (RNN), that has long-term memory in the form of weights, to resolve the issue of vanishing gradients in RNN. A bidirectional LSTM (BiLSTM), is a sequence processing model that consists of two LSTMs, the first taking the input in a forward direction and the other in a backwards one, which helps preserve the future and past information. The advantage of using BiLSTM is effectively increasing the amount of information available to the network. Similar to the deep learning models, BiLSTM has hidden input layers and an output layer, that each takes a specific number of neurons. Additionally, it has forward and backward layers.

Based on the different studies (Yafoz et al., 2022), (Elgeldawi et al., 2021), and trials, different numbers of input layers were implemented, with a different combination of neurons numbers, while applying the dropout where randomly selected neurons are ignored. Depending on the number of features that the dependent variable has, which is three for Sentence-Level (Positive, Negative, Neutral) and four for Aspect-Level (Positive, Neutral, Negative, No annotation), the number of neurons for the output layer has been chosen. Regarding activation and loss functions, softmax and sparse categorical crossentropy have been chosen, to be able to handle multiclass classification. While fitting the model, depending on the size of the data and previous parameters chosen, different epochs and batch sizes were implemented to be able to obtain the best accuracy.

7 Results and Discussion

After running the models on both sentence-level and aspect-level corpus annotations with default and tuned hyperparameters, different graphs and tables showing the results were created and can be found in the Appendix. The tables and the graphs represent the best accuracy, precision and F1-scores obtained in each aspect-level category and on the whole sentence-level. The overall results reflected in section 8 partially support our suggestion stated in the section of NLP models and methods. BiLSTM model perform the best in sentence-level and design aspect-level analyses. In contrast, the MNB model tends to fall short on all aspect- and sentence-level analyses.

7.1 Hyperparameter Tuning

The parameter tuning results with accuracy, F1-score and precision are presented in detail in section 8 and section 8. The following sections will summarize parameters and the best results obtained with different models in aspect and sentence-level analyses.

7.2 Multinomial Naive Bayes Parameters

To begin, the alpha parameter was tuned using MNB model aiming to identify the best performance. Although the difference is small and it is not clear if it is statistically significant, as can be seen from Figure 4 in the appendix the alpha scores to obtain the best results differed between different aspects and sentence-level analysis. The best accuracy, precision and F1-scores obtained were 0.926, 0.930 and 0.926 with the parameter alpha 1.0 in the comfort aspect-level. In Size aspect-level MNB performed the worst in all of the aspect-levels, with an accuracy, precision and F1-score of 0.730 each, with an alpha equal to 0.5. Such results could be the influence of the low IAA score introduced in section 5. Regarding the Sentence-level analysis, the best performance with MNB was obtained with an alpha equal to 0 with an accuracy, precision and F1-score of 0.838, 0.842 and 0.839 respectively.

7.3 Logistic Regression Parameters

The parameters tuned for the LR model were the c-score and max iterations. The best c-value parameter for the LR model in all aspect-level analyses appears to be the 0.1 value while in the sentence-level analysis, the best c-value is 2.0, which can be seen in the Figure 3. The max-iteration parameter made a difference only in the sentence-level analysis with the best performance with max-iterations equal to 400. The accuracy, precision and F1-scores obtained were 0.940, 0.941 and 0.939 respectively. Regarding the aspect-level we were able to obtain the best model performance with comfort as an aspect, with accuracy, precision and F1-score of 0.981, 0.981 and 0.992 using a c-value of 2. As can be seen in the Figure 3, the variation of max-iterations did not affect the performance of the model, in contrast to c-value who affected the performance.

7.4 Linear SVC Parameters

The parameters tuned for this model were the c-value and max iterations. The SVC model results were peculiar. The behavior of the model's performance heavily relied on the parameters used and appeared to be volatile. In contrast with other models, the performance of an SVC model on different aspect-levels differed greatly while using different parameters, specifically max iteration count. This peculiarity can be clearly observed on the aspect-level analyses but not on the sentence-level analysis Figure 5. The current study, however, failed to identify the reason behind such results. More studies need to be done to explore this output.

In an aspect-level analysis the model performed well with a c-value of 1.0 in all categories whilst in sentence-level analysis, the best accuracy and precision were obtained using a c-value of 1.8.

7.5 BiLSTM

The hyperparameter used for this model were the number of epochs. As can be seen in Figure 6 the performance of a BiLSTM model in both analyses steadily raises with the increase in the number of Epochs, however, a slight, insignificant decrease in some aspect-level analyses can be noticed at different epochs depending on the aspect which could be a direct result of noisy data and imperfect annotator agreement.

The BiLSTM model tends to perform the best in design aspect-level analysis with an accuracy of 0.945, a precision of 0.942 and F1-score of 0.943, and sentence-level one with an accuracy of 0.958, a precision of 0.955 and F1-score of 0.956.

7.6 Discussion

After training our models, their performance was tested on a small dataset containing comments scraped from shein.com, a corpus different than the training set. The chosen model for sentiment-level was able to achieve better performances with the long sentences instead of the short ones. Short sentences were usually classified as positive by the model, and the reason to that may be the short length of positive sentences and the lack of short negative sentences in the corpus. In contrast, the respective models chosen for aspect-level didn't face this issue, as the classification was based on specific words or short word combinations.

Whether in sentence-level or aspect-level, the worst performing model was MNB. The reason for this bad performance might be due to the model's assumption that features are independent of each other (Arar and Ayan, 2017). In other words, the combination of words are treated the same as each word alone. However, in our case the words (features) are interrelated especially for classifying neutral and negative sentiments either for sentence or aspect-level. Hence, the decrease in the performance.

The best performing model in the sentence-level was found to be BiLSTM, while in aspect level, the best performing model was found to be LinearSVC, which showed a better although unstable performance, especially, in size and material aspects.

8 Conclusion

Our study investigated performance of different models on aspect-level and sentence-level analyses with various hyper-parameter values. It summarized the performance of each model using different hyperparameters and identified the best parameters for each model in both sentence and aspect-level analyses. Additionally, the study provided a summary of the SA history and process, and reviewed current the research conducted in the recent years. The paper partially confirmed the suggestion raised in section 5 that the BiLSTM model will perform the best in all aspect-level and sentence-level analyses. The suggestion was confirmed partially, since the best performance was only seen in design aspect-level, while for the other aspects, the best performing model was LinearSVC.

References

Mathias M. Adankon and Mohamed Cheriet. 2009. *Support Vector Machine*, pages 1303–1308. Springer US, Boston, MA.

Munir Ahmad, Shabib Aftab, Syed Shah Muhammad, and Sarfraz Awan. 2017. Machine learning techniques for sentiment analysis: A review. *International Journal of Multidisciplinary Sciences and Engineering*, 8:27–32.

Mohammed Almashraee, Dagmar Monett, and Adrian Paschke. 2016. Emotion level sentiment analysis: The affective opinion evaluation.

Anonymous. 2017. *Women's E-Commerce Clothing Reviews*.

Anonymous. 2021. Natural language processing: Rnn & lstm's.

Anonymous. 2022. Eda & lstm classification on clothing reviews.

Ameen Aqlan, Dr. Manjula Bairam, and R Lakshman Naik. 2019. *A Study of Sentiment Analysis: Concepts, Techniques, and Challenges*, pages 147–162.

Omer Arar and Kürşat Ayan. 2017. A feature dependent naive bayes approach and its application to the software defect prediction problem. *Applied Soft Computing*, 59.

Atanas Atanassov and Fani Tomova. 2019. Comparision of two sentiment analysis algorythms. *Industry 4.0*, 4:216–219.

Wouter Atteveldt, Mariken van der Velden, and Mark Boukes. 2021. The validity of sentiment analysis: Comparing manual annotation, crowd-coding, dictionary approaches, and machine learning algorithms. *Communication Methods and Measures*, 15:1–20.

Mohammad Ehsan Basiri, Shahla Nemati, Moloud Abdar, Erik Cambria, and U. Rajendra Acharya. 2021. Abcdm: An attention-based bidirectional cnn-rnn deep model for sentiment analysis. *Future Generation Computer Systems*, 115:279–294.

Salima Behdenna, Fatiha Barigou, and Ghalem Belalem. 2018. Document level sentiment analysis: A survey. *EAI Endorsed Transactions on Context-aware Systems and Applications*, 4:154339.

Gábor Berend and Richárd Farkas. 2008. Opinion mining in hungarian based on textual and graphical clues. pages 408–412.

Burhan Bilen and Fahrettin Horasan. 2022. Lstm network based sentiment analysis for customer reviews. *Politeknik Dergisi*, 25(3):959 – 966.

Stephanie Chevalier. 2022. Global retail e-commerce sales 2026. *Statista*.

Kadir Duran. 2021. Nlp sentiment classification with ml and dl models.

Enas Elgeldawi, Awny Sayed, Ahmed Galal, and Alaa Zaki. 2021. Hyperparameter tuning for machine learning algorithms used for arabic sentiment analysis. *Informatics*, 8.

Richard Evans and Shaun Tougher. 2022. *Generalship in Ancient Greece, Rome and Byzantium*. Edinburgh University Press.

Mohsen Farhadloo and Erik Rolland. 2016. Fundamentals of sentiment analysis and its applications. In Witold Pedrycz and Shyi-Ming Chen, editors, *Sentiment Analysis and Ontology Engineering - An Environment of Computational Intelligence*, volume 639 of *Studies in Computational Intelligence*, pages 1–24. Springer.

Ankita Gandhi, Kinjal Adhvaryu, Soujanya Poria, Erik Cambria, and Amir Hussain. 2023. Multimodal sentiment analysis: A systematic review of history, datasets, multimodal fusion methods, applications, challenges and future directions. *Information Fusion*, 91:424–444.

Ankur Goel, Jyoti Gautam, and Sitesh Kumar. 2016. Real time sentiment analysis of tweets using naive bayes. In *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*, pages 257–261.

Anees Ul Hassan, Jamil Hussain, Musarrat Hussain, Muhammad Sadiq, and Sungyoung Lee. 2017. Sentiment analysis of social networking sites (sns) data using machine learning approach for the measurement of depression. In *2017 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 138–140.

Dziri Jalal and Tahar Ezzedine. 2020. Decision tree and support vector machine for anomaly detection in water distribution networks. In *2020 International Wireless Communications and Mobile Computing (IWCMC)*, pages 1320–1323.

Honghe Jin. 2022. Hyperparameter Importance for Machine Learning Algorithms. *arXiv e-prints*.

Julia Lavid. 2010. Towards a science of corpus annotation: a new methodological challenge for corpus linguistics. *International Journal of Translation*, 22:13–36.

Rolfo Luna. 2019. E-commerce predict recommendation.

Mika Viking Mäntylä, Daniel Graziotin, and Miikka Kuutila. 2016. The evolution of sentiment analysis - A review of research topics,venues, and top cited papers. *CoRR*, abs/1612.01556.

Sandeep Nigam, Ajit Kumar Das, and Rakesh Chandra. 2018. Machine learning based approach to sentiment analysis. In *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, pages 157–161.

Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Edouard Duchesnay, and Gilles Louppe. 2012. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12.

Omkar Shinde, Rishikesh Gawde, and Anurag Paradkar. 2021. Image caption generation methodologies. *International Research Journal of Engineering and Technology (IRJET)*, 8:3961–3966.

Martina Katalin Szabó, Veronika Vincze, Katalin Ilona Simkó, Viktor Varga, and Viktor Hangya. 2016. A Hungarian sentiment corpus manually annotated at aspect level. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2873–2878, Portorož, Slovenia. European Language Resources Association (ELRA).

Sujan Tamrakar, Bal Krishna Bal, and Rajendra Thapa. 2020. *Aspect Based Sentiment Analysis of Nepali Text Using Support Vector Machine and Naive Bayes*. Ph.D. thesis.

Eka Surya Usop, R. Rizal Isnanto, and Retno Kusumaningrum. 2017. Part of speech features for sentiment classification based on latent dirichlet allocation. In *2017 4th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*, pages 31–34.

Andrew J. Einstein Joanna X. Liang Daniel S. Berman Piotr Slomka Vikash Signh, Michael Pencina.

Yu Wang, Tom Clark, Jeffrey Staton, and Eugene Agichtein. 2014. Towards tracking political sentiment through microblog data. pages 88–93.

Aji Wibawa, Ahmad Kurniawan, Della Murti, Risky Perdana Adiperkasa, Sandika Putra, Sulton Kurniawan, and Youngga Nugraha. 2019. Naïve bayes classifier for journal quartile classification. *International Journal of Recent Contributions from Engineering, Science & IT (iJES)*, 7:91.

Ayman Yafoz, Farial Syed, Malek Mouhoub, and Lisa Fan. 2022. Analysing the sentiments in online reviews with special focus on automobile market. pages 261–267.

Li Yang, Ying Li, Jin Wang, and R. Simon Sherratt. 2020. Sentiment analysis for e-commerce product reviews in chinese based on sentiment lexicon and deep learning. *IEEE Access*, 8:23522–23530.

Omnia Yasser. 2021. Women clothing reviews classification with rnn.

Quan Yuan, Gao Cong, and Nadia Thalmann. 2012. Enhancing naive bayes with various smoothing methods for short text classification.

Nayyar Zaidi, Jesús Cerquides, Mark Carman, and Geoffrey Webb. 2013. Alleviating naive bayes attribute independence assumption by attribute weighting. *The Journal of Machine Learning Research*, 14:1947–1988.

Aziz Özmen. 2021. Nlp: Comparative rnn & dl models with detailed eda.

Appendix A

Model	Vectorization Method	Test Size	Hyper Parameters	Accuracy	Precision	F1
LR	Count Vectorizer	0.10	2.00 (C Value), 500 (Max Iterations)	0.940	0.941	0.939
	Count Vectorizer	0.25	Default: 1.00 (C Value), 100 (Max Iterations)	0.845	0.845	0.845
	TF-IDF	0.10	2.00 (C Value), 200 (Max Iterations)	0.890	0.892	0.890
	TF-IDF	0.15	Default: 1.00 (C Value), 100 (Max Iterations)	0.859	0.860	0.859
MNB	Count Vectorizer	0.20	0 (Alpha)	0.838	0.842	0.839
	Count Vectorizer	0.15	Default: 1.0 (Alpha)	0.804	0.809	0.806
	TF-IDF	0.20	0 (Alpha)	0.847	0.852	0.848
	TF-IDF	0.15	Default: 1.0 (Alpha)	0.806	0.814	0.808
SVM	Count Vectorizer	0.10	1.8 (C Value)	0.944	0.945	0.943
	Count Vectorizer	0.15	Default: 1.0 (C Value)	0.939	0.941	0.939
	TF-IDF	0.15	1.8 (C Value)	0.924	0.925	0.924
	TF-IDF	0.10	Default: 1.0 (C Value)	0.910	0.911	0.910

Table 1: Model Sentence-Level Results

Model	Vectorization Method	Test Size	Hyper Parameters	Accuracy	Precision	F1
LR	Count Vectorizer	0.10	2.00 (C Value), 400 (Max Iterations)	0.981	0.981	0.992
	Count Vectorizer	0.10	Default: 1.00 (C Value), 100 (Max Iterations)	0.976	0.976	0.987
	TF-IDF	0.10	2.00 (C Value), 400 (Max Iterations)	0.934	0.935	0.950
	TF-IDF	0.30	Default: 1.00 (C Value), 100 (Max Iterations)	0.918	0.918	0.931
MNB	Count Vectorizer	0.10	0.5 (Alpha)	0.919	0.924	0.919
	Count Vectorizer	0.30	Default: 1.0 (Alpha)	0.926	0.930	0.926
	TF-IDF	0.15	0.0 (Alpha)	0.904	0.905	0.904
	TF-IDF	0.10	Default: 1.0 (Alpha)	0.930	0.933	0.930
SVM	Count Vectorizer	0.10	0.01 (C Value)	0.977	0.977	0.988
	Count Vectorizer	0.25	Default: 1.0 (C Value)	0.986	0.983	0.999
	TF-IDF	0.1	1.5 (C Value)	0.965	0.989	0.997
	TF-IDF	0.25	Default: 1.0 (C Value)	0.969	0.969	0.980

Table 2: Model Aspect-Level Results for Comfort

Model	Vectorization Method	Test Size	Hyper Parameters	Accuracy	Precision	F1
LR	Count Vectorizer	0.15	1.50 (C Value), 400 (Max Iterations)	0.783	0.784	0.781
	Count Vectorizer	0.10	Default: 1.00 (C Value), 100 (Max Iterations)	0.823	0.823	0.822
	TF-IDF	0.15	2.00 (C Value), 400 (Max Iterations)	0.813	0.814	0.811
	TF-IDF	0.30	Default: 1.00 (C Value), 100 (Max Iterations)	0.758	0.760	0.757
MNB	Count Vectorizer	0.10	0.5 (Alpha)	0.730	0.730	0.730
	Count Vectorizer	0.15	Default: 1.0 (Alpha)	0.726	0.734	0.726
	TF-IDF	0.10	0.5 (Alpha)	0.755	0.765	0.754
	TF-IDF	0.15	Default: 1.0 (Alpha)	0.753	0.760	0.749
SVM	Count Vectorizer	0.10	1.0 (C Value)	0.847	0.847	0.844
	Count Vectorizer	0.15	Default: 1.0 (C Value)	0.930	0.931	0.930
	TF-IDF	0.10	1.8 (C Value)	0.789	0.790	0.787
	TF-IDF	0.10	Default: 1.0 (C Value)	0.788	0.791	0.786

Table 3: Model Aspect-Level Results for Size

Model	Vectorization Method	Test Size	Hyper Parameters	Accuracy	Precision	F1
LR	Count Vectorizer	0.15	1.00 (C Value), 400 (Max Iterations)	0.914	0.913	0.913
	Count Vectorizer	0.15	Default: 1.00 (C Value), 100 (Max Iterations)	0.913	0.911	0.912
	TF-IDF	0.20	2.00 (C Value), 400 (Max Iterations)	0.853	0.848	0.848
	TF-IDF	0.10	Default: 1.00 (C Value), 100 (Max Iterations)	0.864	0.860	0.860
MNB	Count Vectorizer	0.10	0.0 (Alpha)	0.896	0.898	0.895
	Count Vectorizer	0.10	Default: 1.0 (Alpha)	0.887	0.887	0.886
	TF-IDF	0.10	0. (Alpha)	0.890	0.894	0.888
	TF-IDF	0.10	Default: 1.0 (Alpha)	0.857	0.858	0.851
SVM	Count Vectorizer	0.15	0.05 (C Value)	0.930	0.932	0.930
	Count Vectorizer	0.25	Default: 1.0 (C Value)	0.929	0.930	0.928
	TF-IDF	0.20	0.6 (C Value)	0.896	0.894	0.894
	TF-IDF	0.20	Default: 1.0 (C Value)	0.896	0.894	0.894

Table 4: Model Aspect-Level Results for Design

Model	Vectorization Method	Test Size	Hyper Parameters	Accuracy	Precision	F1
LR	Count Vectorizer	0.10	0.50 (C Value), 100 (Max Iterations)	0.873	0.873	0.868
	Count Vectorizer	0.10	Default: 1.00 (C Value), 100 (Max Iterations)	0.879	0.879	0.875
	TF-IDF	0.10	0.10 (C Value), 100 (Max Iterations)	0.788	0.793	0.784
	TF-IDF	0.10	Default: 1.00 (C Value), 100 (Max Iterations)	0.788	0.793	0.784
MNB	Count Vectorizer	0.10	0.5 (Alpha)	0.844	0.840	0.841
	Count Vectorizer	0.15	Default: 1.0 (Alpha)	0.828	0.824	0.823
	TF-IDF	0.15	0.0 (Alpha)	0.835	0.832	0.833
	TF-IDF	0.10	Default: 1.0 (Alpha)	0.790	0.785	0.779
SVM	Count Vectorizer	0.10	1.80 (C Value)	0.920	0.919	0.919
	Count Vectorizer	0.10	Default: 1.0 (C Value)	0.920	0.919	0.919
	TF-IDF	0.10	0.9 (C Value)	0.868	0.868	0.865
	TF-IDF	0.10	Default: 1.0 (C Value)	0.877	0.877	0.874

Table 5: Model Aspect-Level Results for Material

Appendix B

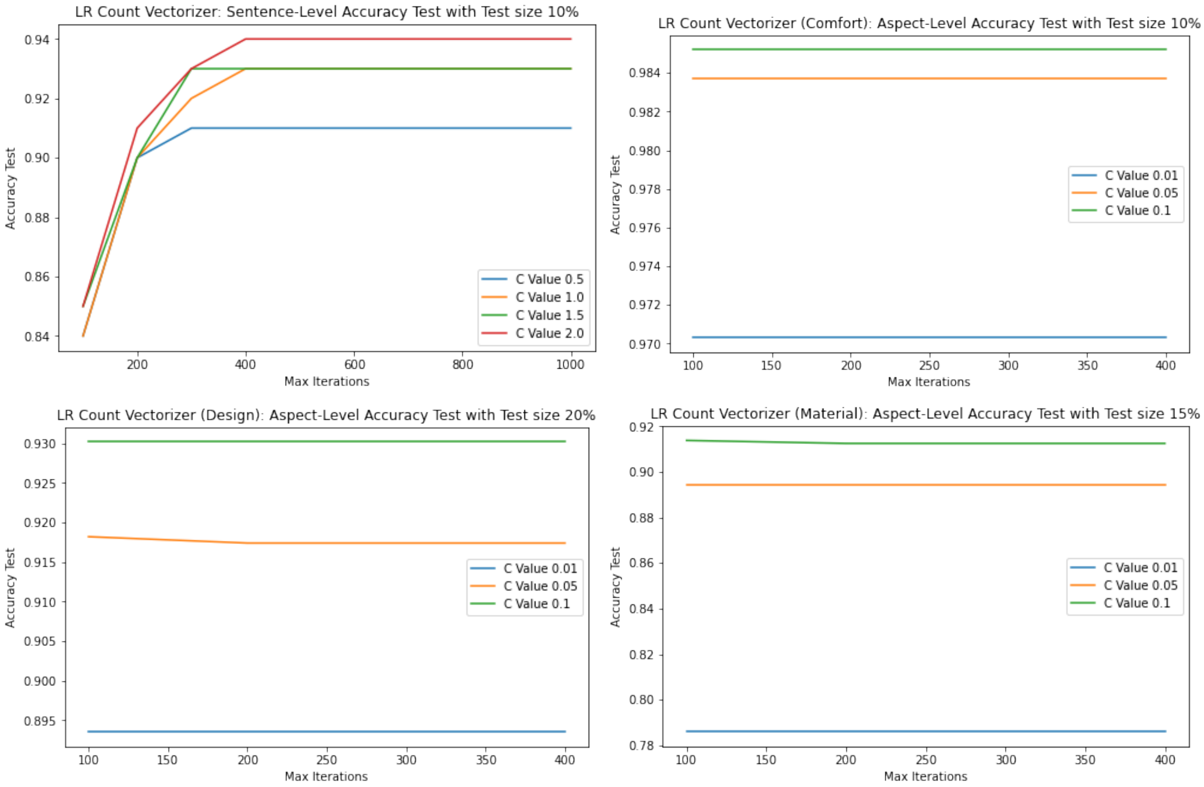

Figure 3: Logistic Regression on sentence-level different aspects data

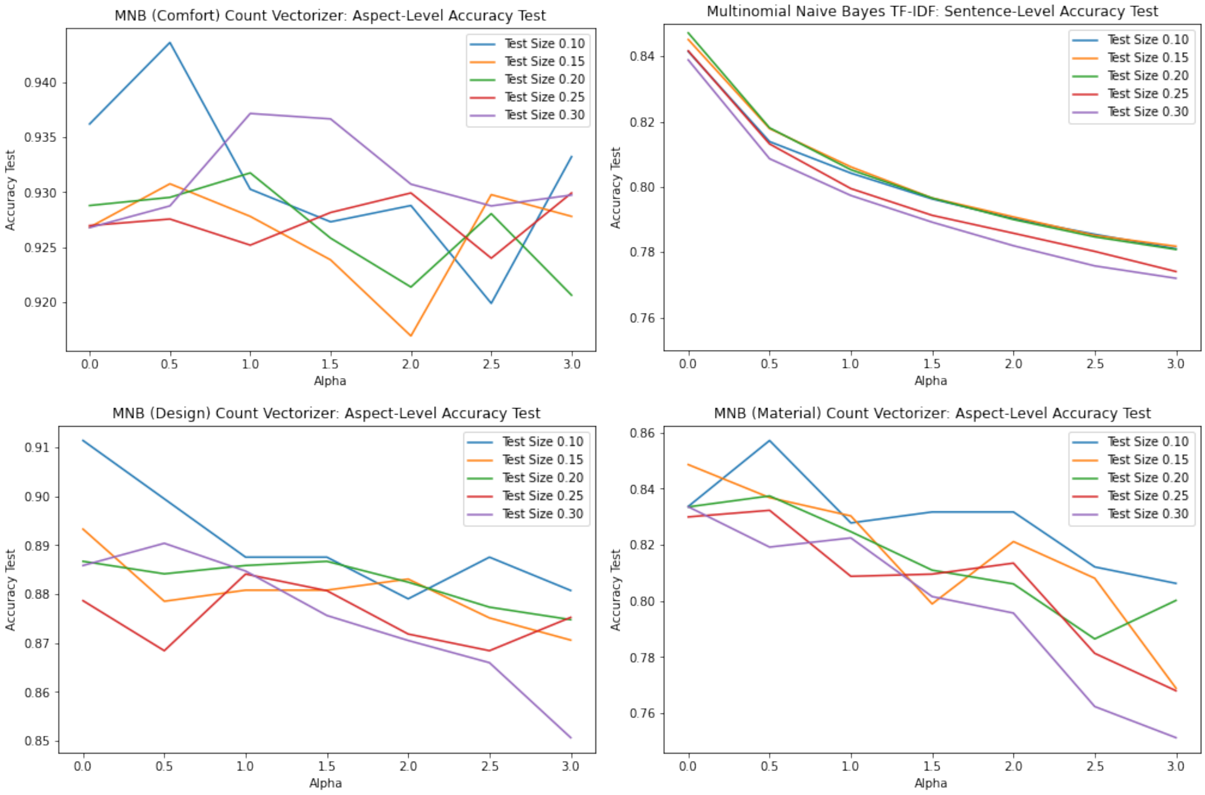

Figure 4: Multinomial Naive Bayes on sentence-level and different aspects data

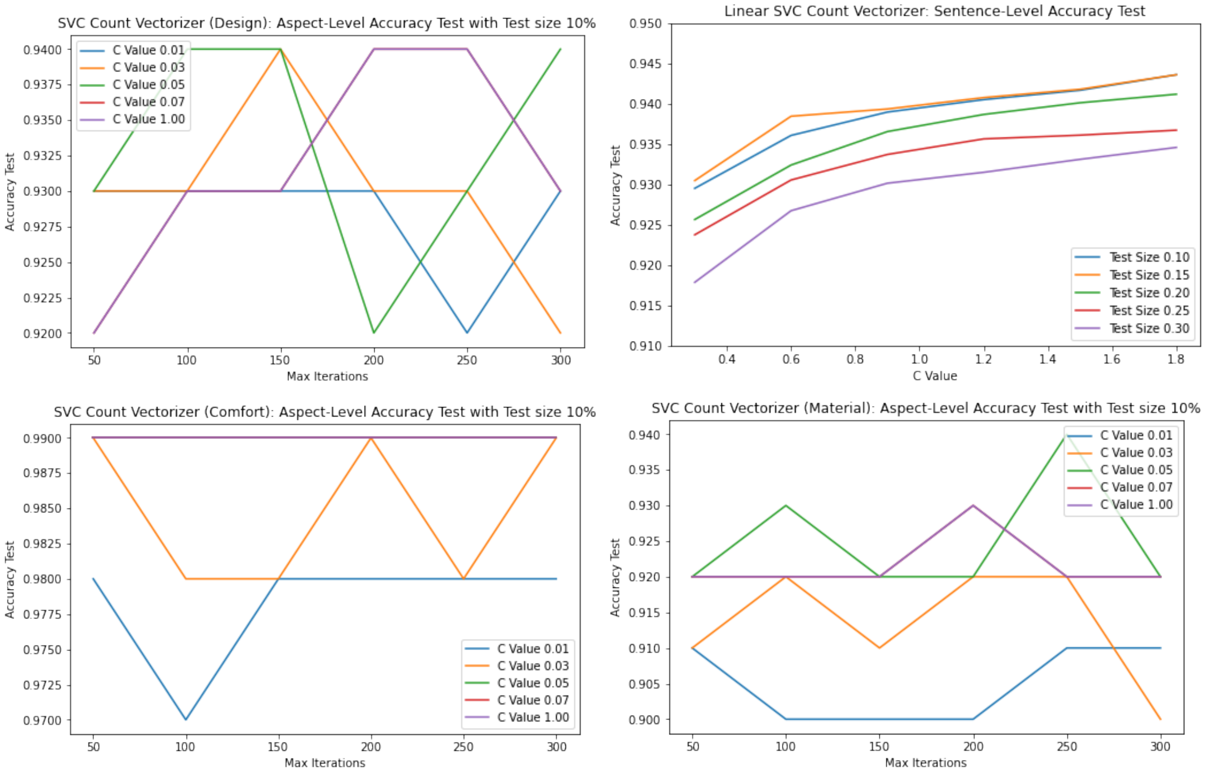

Figure 5: SVC on sentence-level and different aspects data

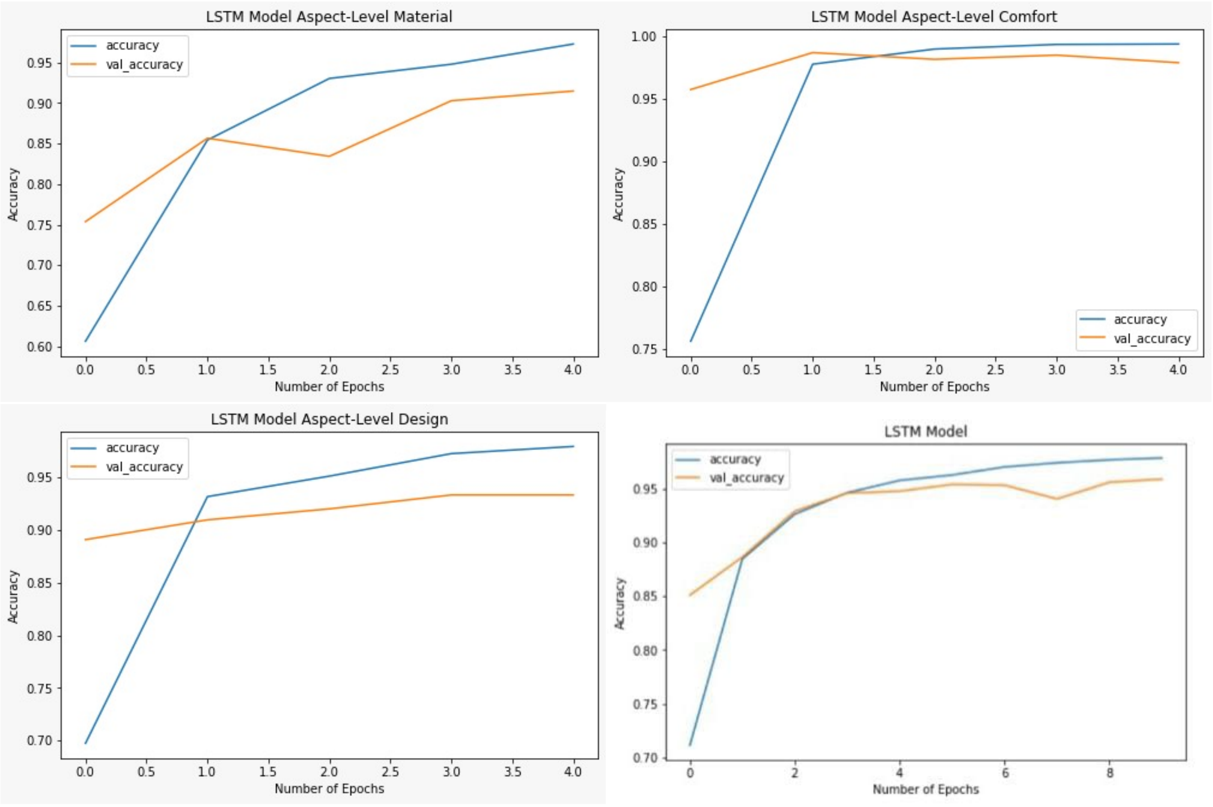

Figure 6: LSTM on sentence-level and different aspects data

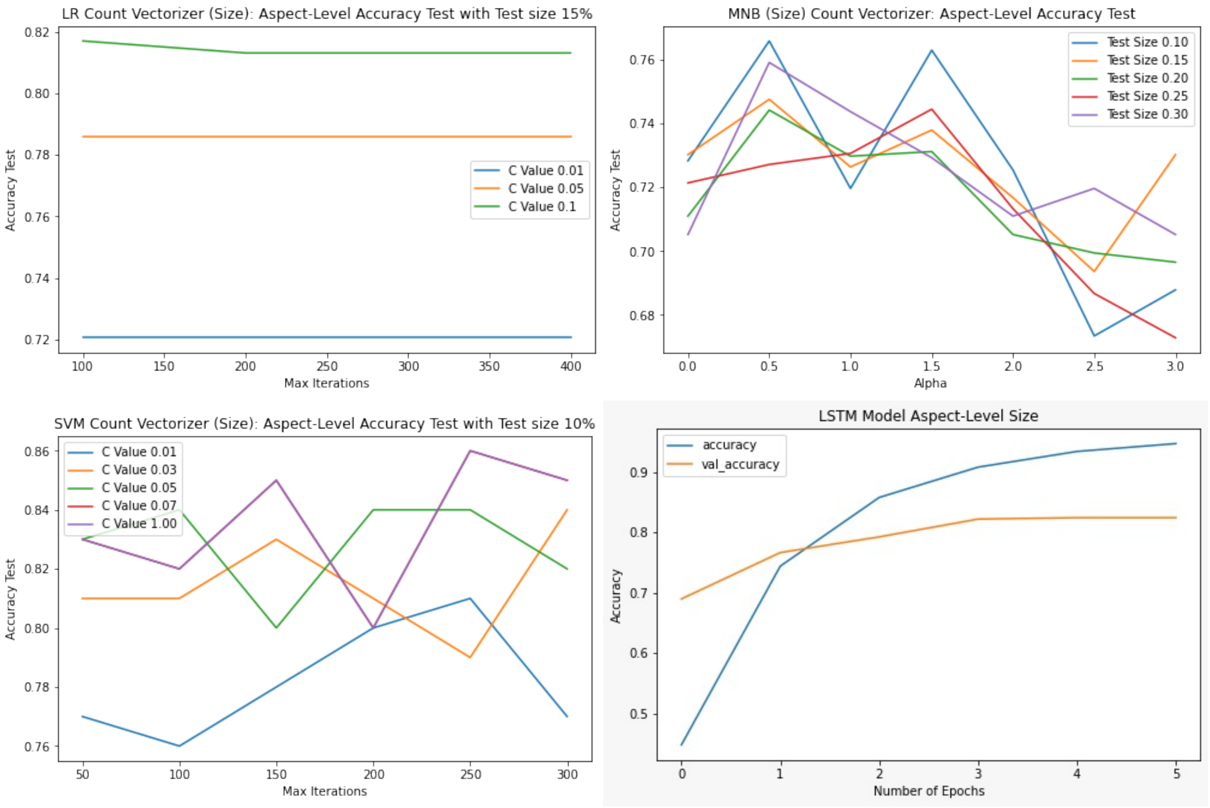

Figure 7: Size-aspect analysis with different models