DOMAIN SPECIFIC ASPECT EXTRACTION FOR PRODUCT DESIGN

A Thesis

Presented to

The Faculty of the Department of Computer Engineering

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Divyam Sobti

May 2024

The Designated Thesis Committee Approves the Thesis Titled

DOMAIN SPECIFIC ASPECT EXTRACTION FOR PRODUCT DESIGN

by

Divyam Sobti

APPROVED FOR THE DEPARTMENT OF COMPUTER ENGINEERING

SAN JOSÉ STATE UNIVERSITY

May 2024

Mahima Suresh, Ph.D.	Department of Computer Engineering
Wencen Wu, Ph.D.	Department of Computer Engineering
Vimal Viswanathan, Ph.D.	Department of Mechanical Engineering

ABSTRACT

DOMAIN SPECIFIC ASPECT EXTRACTION FOR PRODUCT DESIGN

by Divyam Sobti

As technology advances, computers become increasingly proficient at interpreting and translating human language into machine-understandable text. With the help of algorithms in natural language processing (NLP), machines can now translate textual data. These algorithms help identify and extract specific text components known as aspects. The aspects represent specific attributes or topics within textual data. For instance, an Amazon review states,“ This phone has good battery life but poor camera quality,” and attributes like ‘battery life’ and ‘camera quality’ represent aspects in the text. Aspect extraction is a pivotal process involving identifying and isolating key features or topics within text. This research aims to compare and discuss the existing aspect extraction techniques. By effectively extracting the aspects, we will help machines gain the capability to understand and analyze sentiments, thereby enhancing their ability to derive meaningful insights from diverse textual data. Aspect-based sentiment analysis (ABSA) enables the extraction of sentiments towards specific aspects of a product the user provides. For example, when a person writes a review about a restaurant, sentiment analysis can determine whether the review is positive or negative. Sentiment analysis helps us determine the polarity of that review. ABSA can separately determine the review’s sentiment towards different aspects of a restaurant, such as, food quality, ambiance, etc. We propose an approach to extract features/aspects of customer-based product reviews. Our approach is divided into two parts: first, the sentiment analysis of the reviews, which tells whether the review is positive or negative, and second, extracting the aspects from the reviews. The system then converts the consumer experience into text to help both new and seasoned merchants to improve their products. The suggested outcome will result in better consumer experience and business growth.

ACKNOWLEDGMENTS

Undertaking this thesis has been an informative and transformative endeavor, allowing me to analyze the depths of my research field and push the boundaries of academic objectives. The journey, rich in its challenges and accomplishments, would not have been possible without the support, guidance, and inspiration provided by a star of remarkable individuals to whom I am deeply indebted to Dr. Mahima Agumbe Suresh.

Dr. Mahima Agumbe Suresh, whose superior mentorship has been the guiding light of my research journey. Dr. Suresh's profound expertise, insightful feedback, and unwavering support have been instrumental in my work's formulation, ideation, and experimentation stages. His ability to foster a nurturing and intellectually stimulating research environment has dramatically enriched my academic experience and professional growth. Dr. Suresh, I extend my heartfelt gratitude for being an exceptional advisor and a paragon of academic excellence.

Lastly, I would also like to thank my family and friends in the United States and India for being my support system as I guided the challenges of my graduate studies.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

The vast data available today provides excellent insights into the value of different commercial fields. Previously, manufacturers and product designers had to rely on customers to use their products and give them feedback, which was used to improve their decision-making process and product improvement. Different methods were used to get variety in customer feedback, such as surveys, interviews, and many other methods, and that is where the reviews help us. It quickly provides a variety and volume of data, allowing us to understand our customers' needs and focus on them rather than collecting. With the sensitive economy, time is crucial for product designers, as if they spend more time on a concept, their competitors will seize the market opportunity. After facing these challenges, customer review analysis has become very important. With the help of machine learning, especially Natural Language Processing, popular research in this field has focused on sentiment analysis and aspect-based sentiment analysis to understand customer demand and improve satisfaction.

1.1 Background

Sentiment analysis is a process of understanding the emotions of the person writing the review. Sentiment analysis is very important for product designers as they get to know the pulse of the customer who is given the feedback. It needs to be more helpful as it only provides limited information about a customer, whether a customer is happy, neutral, or has opposing views on the product. However, this approach is not enough as the review consists of an extensive, expressive, and subjective nature; many narrative details will be missed customer experience if we just skim through the emotional part of the reviews, which is just like scratching the surface and leaving insights unexplored. Aspect-based sentiment analysis(ABSA) [1] addresses this limitation. Aspect extraction is finding the essential features or the vital part of the review that determines the context. It provides an approach that uses Natural Language Processing with sentiment analysis to abstract the aspects or features of the product to which customer directs their sentiment. So, the product designer gets insights into the rich quality of customer feedback. Focusing on areas like customer reviews is very important to improve the product. The decision to develop a product using online review mining is optimistic about improving its quality, which makes ABSA the right choice. The NLP-based algorithm ABSA has been extensively explored. However, the limited product-specific data makes this approach unsupervised learning. The text category used to train these machine learning models is very important and generally overlooked. For instance, customer reviews are very product-specific; the data should be domain-specific to produce better performance. Most of the work done on ABSA is to improve "state-of-the-art" performance, but not much is done on domain-specific feature selection. These works used some pre-trained language representation on generic text such as corpora, Wikipedia (dropping domain-specific), or some trained language model(reducing the volume of language information). Lastly, very little thought is given to training text (domain quality) but more to the design and working of algorithms.

1.2 Overview of NLP Techniques

Machine Learning focuses on converting complex raw data into significant features and then analyzing these features statistically to uncover patterns. In this context, raw data means raw text and is our primary source. Features selected as words in the sentence/reviews and features as phrases [2] generally get transformed into numeric data. Initially, this was done by one-hot encoding, where each word gets an index. Then, a sentence or whole document is considered a binary array, with the length being the same as a sentence or every word in a document. Each index covers one word and indicates its presence or absence.

The challenges in this approach are undeniable: the number of features generated by this is vast even after we do preprocessing steps such as lemmatization and case adjusting. Many features make it challenging to analyze the data, which can also lead to the loss of some vital information, such as covariance. Latent Dirichlet Allocation [3] was an earlier attempt to represent words with meaningful semantic features, but it had limitations [4].

Currently, most NLP methods have been influenced by the work of Mikilov et al. [5] [6], which uses word vectors or embedding. This method involves representing each word in the vocabulary as a numerical vector of a fixed size; it is because each word found in a document tends to have a similar meaning, striking a balance between capturing the quality of information and keeping the data size small. We have experimented with various techniques to generate these word vectors. Our goal is to provide insights that could be helpful for product designers and manufacturers. Further details will be discussed in Section 4

1.3 Our Work

The previous work in this field [7]–[10] has detailed the importance of domain-adaption for this task. We hypothesize that the ABSA algorithm with more domain-specific training text will improve its working and, with the help of vectors, we will have better results than our previous work [8]. Many different methods have been discovered for using ABSA in [11], [12]. We use RoBERTa and Vader as sentiment analysis and pass these through to our machine learning model. Our work can be divided into four parts:-

- Corpus search: We begin by collecting large amounts of raw text data, such as product reviews. The data we used is publicly available for academic and research use.
- Sentiment analysis: After collecting the data, the next task is to find the sentiment of each review and filter these reviews by their sentiment. This will help us identify which aspects of a product customers are happy with and which are not.
- Embedding Model Training: World embeddings are the heart and brain of our experiment; we train each corpus collected by the last step using different word and sentence embeddings.
- Aspect Extraction Process: We applied various machine learning algorithms to the vectors/embedding we got from our last steps and tried to predict aspects in these reviews. The algorithmic processes we utilized are detailed in Section 4, while Section 5 focuses on the key findings from our experiments.

The result of our work, which is discussed above, is :

- Our approach of clustering with embedding on domain-specific has shown to be effective, following Ruidan He's work in [11], [12].
- We implemented sentiment analysis with the expectation of obtaining better results. However, several aspects were neglected, which caused the result to be unsatisfactory.

The subsequent sections of this work are structured in the following manner: The following section discusses the past work done in the field of ABSA and ABSA in domain-specific fields. Section 3 consists of comprehensive information about the dataset we used. Section 4 will discuss our methods. Section 5 will cover the experiment and results in 6. Finally, Section 7 summarizes the conclusions and details about the future scope of this research in Section 8.

2 LITERATURE REVIEW

This section provides a brief overview of early methods in Aspect-based semantic analysis(ABSA) and the latest approaches used for it. ABSA uses both feature selection and pattern detection algorithms to provide multi-label classification. Research work [9] provides innovations in ABSA over the past decade.

The first step in Aspect-Based Sentiment Analysis (ABSA) is the 'aspect-term extraction,' a significant step in detecting sentiment polarity. Initially, the field leaned heavily on traditional, rule-based methodologies to solve this challenge. Techniques such as part-of-speech (POS) tagging, the construction and analysis of syntax trees, mining for frequently occurring items, and dependency parsing were used. These approaches used supervised learning algorithms, such as Support Vector Machines (SVMs) [10] and Random Forests [13], which provided a structured way to interpret and classify text data. In the initial stages of ABSA's growth, there was a strong dependence on well-established linguistic rules and patterns for the identification of aspect terms in texts. The employment of syntax trees, for example, enabled researchers to dissect sentences to reveal the structural relationships among words, thus simplifying the process of identifying the aspects under discussion. Therefore, dependency parsing provided valuable insights into the grammatical relationships between words, playing a crucial role in grasping the context in which aspects were mentioned. While these early methods proved somewhat effective, they often demanded considerable manual adjustment. They were naturally restricted by their rule-based essence, leading the domain to venture into more adaptable and advanced machine learning models. Later Wang et al. [14] discussed the incorporation of deep learning techniques using Recurrent Neural Network(RNN) [15] [16] and Long Short Term Memory(LSTM) [13]. However, experiment results showed that SVM [17] or Conditional random fields(CRF) [18] based approaches are more efficient for this task.

Furthermore, the introduction of attention mechanisms significantly changed the way models processed complex datasets, leading to revolutionary enhancements across numerous

NLP tasks. Attention mechanisms' capability to dynamically concentrate on different segments of the input data provided the creation of more nuanced and context-sensitive models. This advancement proved to be particularly revolutionary in the field of ABSA, where grasping the nuances and context surrounding aspect terms is essential for precise sentiment analysis. Vaswani et al.'s [19] introduction of the transformer architecture, which relies heavily on attention mechanisms, marked another leap forward for NLP. This architecture, devoid of recurrent layers, presented a more efficient and effective method for processing sequences, setting new benchmarks in speed and performance. Transformers quickly became the foundation for a new generation of language models tailored explicitly for ABSA tasks [20]. These models could capture the intricacies of language and sentiment with remarkable precision, largely thanks to their ability to attend to relevant parts of the text when making predictions.

The extraordinary ability of these transformers, including BERT [21], XLNet [22], GPT2, and Transformer XL [23], to accurately model language was crucial in their widespread acceptance among researchers. A particularly appealing feature of these models is their fine-tuning capacity on specific datasets.

Hence, even though supervised learning approaches have been predominant in this area, the lack of large, annotated datasets presents a significant challenge, particularly concerning aspect extraction tasks. Researchers have ventured into unsupervised learning tactics, like topic modeling, for aspect extraction and detection to tackle this. Among these are strategies that combine Latent Dirichlet Allocation (LDA) [24], word embeddings, and Brown Clusters, showing considerable potential. Taking inspiration from these techniques, some of our approaches to aspect detection make use of seed words and clustering, with the effectiveness of attention-based neural models being assessed for their utility in specific datasets.

3 DATASET

The aspect-based sentiment analysis is quite a famous problem in the field of NLP, but the datasets to solve these problems still need to be improved. With the subject of identifying the aspects or features, the review dataset should have accurate definitions, which presents a challenge due to the need for more availability of datasets. We have decided to explore the Amazon reviews dataset as part of our experiment. With the help of this dataset, we can gain insights into consumer sentiment and behavior that are not visible through conventional market analysis. For example, despite rising environmental awareness, a recent trend indicates a contradiction in the market for eco-friendly household cleaning products. We will look into Amazon reviews, which will give us an accurate representation of customer opinions and experiences to help us sort through this complexity. We intend to expand our dataset more than our previous work in [7], where we just explored eco-friendly products. This dataset will contain a wider variety of products and customer reviews. We hope to gain more insight into the customer's views and needs. Each review in the dataset will have one aspect related to it. The main objective of our experiment is aspect detection, which is a multi-label classification problem. Apesect identification/extraction are some other take of ABSA.

3.1 Aspects

Finding the aspects for each review category is crucial in Aspect-Based Sentiment Analysis (ABSA). The entire set of factors must include nearly all of the qualitative information provided in the product reviews, which are essential to consumers and product creators alike. Furthermore, every part of the aspect should be:

- Embodies a noteworthy, pertinent, and significant characteristic.
- Easily identified and describable by human annotators, ensuring straightforward undertaking and consistency.
- It should be unique with minimum overlapping of aspects to maintain clarity for easy analysis.

The selection of aspects varies significantly across different review categories, as the attributes of products like cars are different from the attributes of clothing. However, certain aspects such as “Price” or “Appearance” tend to be universally significant across various categories. However, if we look at a single product category, the specific function and marketing focus of the product can further influence relevant aspects. For example, within the automobile category, the aspects prioritized for a luxury car will differ from those for an economy or electric vehicle. High-end vehicle buyers may place less emphasis on “price” or “fuel efficiency” if the car excels in “performance,” “comfort,” and “luxury features.” On the other hand, consumers who are conscious about budget might prioritize cost-related aspects over luxury enhancements. Using unsupervised learning to analyze big sets of unannotated reviews is an efficient and practical approach. It uses vast raw data available to get useful insights without needing manual labeling of the data. This method helps identify both common and specific features of products without any prior biases.

An application of this is the Attention-Based Aspect Extraction (ABAE) model developed by He et al. This model uses a special technique to transform words into feature vectors, which are then grouped into product aspects based on their positions. Tested on SemEval datasets, the ABAE model automatically figures out important product features, helping us better understand what consumers think and want.

The study also provides observations about how these techniques work with eco-friendly products:

- Redundancy in Aspect Inference: The most noticed aspects were about the products being eco-friendly. Although this is true, it’s also expected since the data specifically focuses on eco-friendly items.
- Aspect Clusters Overlap: Sometimes, the model combines several expected aspects into bigger groups. This could mean the model is generalizing too much, and adjusting how it

identifies aspects and improving how it processes texts could lead to clearer and more specific results.

- Threshold for Aspect Identification: The model did identify some expected aspects, but setting a minimum number of similar words needed to confirm an aspect could help focus on the most relevant and clear aspects, possibly limiting them to a group of 5-8 main aspects.

These points show the strengths and areas for improvement in using advanced unsupervised learning for finding out different aspects of products and pointing out ways to make these tools better and more relevant in figuring out what aspects are important. Further, we can leverage the various word-embedding models to feed into the ABAE algorithm to enhance the quality and accuracy of inferred aspects. The experiment also highlighted the importance of semantic aggregation of aspects. This aggregation was based on the semantic similarities observed, which helps refine the aspect categories more logically and consistently.

3.1.1 Eco-Friendly Products

The review analysis, particularly within the eco-friendly dataset, underscored the subjective nature of aspect annotation, which relies on the review context, the reviewer's focus, and the intended use of the product. For example, a review praising a hair dryer for its quick drying time could be categorized under "Efficiency" or "Performance," depending on whether the product's quick action is a highlighted feature or an expected outcome.

Similarly, a review noting the strong stitches of a cloth bag could be attributed to either "Durability" or "Performance," depending on the reader's expectations. This subjectivity and the overlapping nature of aspects require a sophisticated approach to aspect categorization[8].

In response to these challenges, a revised mapping system was developed to better categorize and label aspects, reducing them from 12 to 6, 4, and 3 refined aspects, which is also discussed in [7]. This new system not only addresses the initial problems of overlap and redundancy but also aims to strike an optimal balance between specificity and generality in aspect categorization, as demonstrated with the adjusted aspect set detailed in 1 of the report.

Table 1
Review's Aspect

Original 12 Aspect	**6 Aspect**
Efficiency Performance	Performance
Aesthetics	Aesthetics
Durability Adaptability Ergonomics	Design
Ease of Reprocessing Ease of Storage Ease of Use	Ease of Use
Interference Safety	Interference
Price	Price

3.1.2 Amazon Validated Products

The Amazon-validated Products dataset [20] is similar to eco-friendly product reviews. It contains a collection of reviews from different products based on amazon.com. The reviews in this dataset have been manually labeled with aspects to enable more accurate Aspect-Based Sentiment Analysis (ABSA). The dataset is diverse, containing reviews from multiple product categories such as electronics, home goods, and more. This diversity ensures that the analysis and results can be generalized across different types of products. Unlike eco-friendly datasets, the aspects in this dataset are manually labeled by humans. This manual process involves reading each review and assigning specific aspect labels that reflect the customer's sentiment and details about the product in the review. The aspects include "Durability," "Ease of Use," "Aesthetics," "Price," "User Experience," and others depending on the product being reviewed.

By analyzing the sentiment linked to specific aspects, businesses can gain insights into what features are performing well and what areas require improvement. Maintaining a precise and accurately labeled dataset can better align product development with consumer expectations and enhance overall customer satisfaction.

4 PREPROCESSING

To prepare the data for future analysis, we must first clean it. Preprocessing is used for data cleaning, integration, and modeling. Preprocessing is the most crucial stage for the unstructured data used in our project. Ensuring the data is appropriate for the algorithm to function correctly and produce good results is essential. The preprocessing steps followed during our experiments are as follows:

4.1 Sentiment analysis

Sentiment Analysis is a part of NLP that tells the polarity of the overall context. When we look at locating and classifying viewpoints to ascertain the writer's stance on a specific subject or the text's overall contextual polarity, this mindset can be neutral, hostile, or positive. Sentiment analysis is a popular tool for market research, brand reputation management, social media monitoring, and comprehending consumer feedback. In this project, we take the preprocessed data and pass it through the model VADER or RoBERTa model to see if the sentence is positive or negative.

4.1.1 VADER:

VADER(Valence Aware Dictionary and sEntiment Reasoner) rule-based sentiment analysis is specifically trained in sentiments expressed on social media. It is particularly effective in dealing with the type of language used in social media, including internet slang, emojis, and colloquialisms. VADER provides the sentiment score by analyzing the words used in a document. VADER follows certain terms and guidelines to provide the sentiment score of a text. It evaluates each word's positivity or negativity using a valence score [25]. The valence score is a given based on observation rather than logic. The valence score for a word is in the range of [-4,+4]. So, it should be helpful to classify the customer reviews based on sentiment.

4.1.2 RoBERTa:

Robustly Optimized BERT Approach(RoBERTa) [26] is an NLP model developed by Facebook AI, which builds upon BERT (Bidirectional Encoder Representations from Transformers).

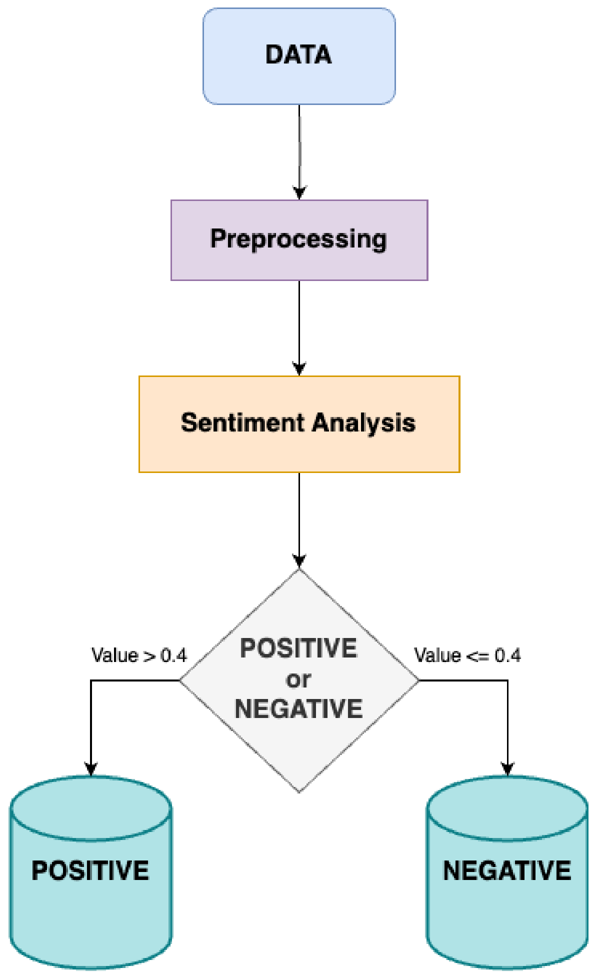

Fig. 1. Flowchart of Sentiment Analysis

It is designed to improve the training and performance of the original BERT model. Since RoBERTa has been trained on significantly more data than the BERT model, it is more adept at accurately identifying patterns within the textual datasets. We have used RoBERTa to perform sentiment analysis on our data for similar reasons, and we obtained better results than the previous VADER model.

As per the figure 1, the working of sentiment analysis can be easily seen. Initially, the data undergoes a preprocessing step that removes all the unnecessary things (more elaborated in the subsequent section). After preprocessing, the data is fed into sentiment analysis models, either

Roberta or VADER, and then these models determine whether the data is positive, neutral, or negative. The positive and neutral data are in the positive dataset, and the negative data are in the negative dataset.

4.2 Cleaning and Normalization

- We removed the unwanted columns and converted all text into lowercase characters to format the data and eliminate unnecessary information. The customer reviews columns contained user information, which we felt was unnecessary and may cause unintended bias in the model. So, we used regex to identify the user information from comments and filtered it out.
- Tokenization and Filtering are the processes of breaking a sentence up into smaller units like words or phrases. We remove stop words that don't contribute any meaning to the text, such as" is,"" an,"" the," etc. After removing stopwords, we perform tokenization on the text and convert input sentences into a concatenation of multiple smaller-sized tokens.
- Stemming: It is a technique that is used to reduce the word to its root word. Stemming chops the word based on the prefix or suffix of that word. This process would convert words such as" cooking" into" cook" and" education" into" educate." Skipping this step reduces the input data quality and affects the model accuracy. The machine learning model quality depends on the data it is trained on.

4.3 Parts-of-Speech tagging (PoS)

One of the main tasks of NLP is parts-of-speech tagging, which gives each word in the text a grammatical context. It provides the phrase structure and semantics. Tagging is essential for various NLP tasks as it provides detailed information about the word usage and syntax of the sentences. Here is how it works:- First, it will split sentences into different words; then each word is analyzed to determine the part of speech it belongs to. The second step is to decide the rule for us. We only wanted to keep just the nouns, so all the words except nouns would be dropped. The third step is to pass this through the models to determine the context of the sentences and understand which word belongs to which speech. The output will be the original text where each word is annotated with its part of speech. So, as per the rule, it filters out everything except the nouns.

4.4 Embeddings

The final preprocessing step, before performing Aspect Detection, is creating vectors from the corpora generated in above section, We researched 3 different embedding algorithms, with one having two variations, making it a 4 algorithms, being employed for this task. These algorithms are highly popular and widely used in the industry.

4.4.1 Word2Vec

Word2Vec is the most renowned algorithm and was introduced by Mikolov et al. [4]. The vectors generated using this technique capture information about the meaning of a word based on surrounding words. Word2Vec is a group of shallow, 2-layered neural networks that are trained to reconstruct the linguistic context of words. The algorithm works by initializing the vectors to random projections in a d-dimensional space, constructing the initial layer of the neural network. Each word is converted into two vectors of d dimensionality, the first one illustrating the term itself and the other representing the word's context. The models are then trained using backpropagation to produce a viable target vector and a context vector, which are then combined/averaged to produce the final word vector representation. Two variants of this algorithm are based on the loss function used during the training process. Further enhancements have also been proposed by the authors of [5], where they modify the loss function to accurately identify a few samples of "negative" words that do not belong in the context of the provided text.

- Continuous Bag-of-Words (CBOW): This variant analyzes the specific context around a word, and it tries to maximize the likelihood of predicting the missing word. Each individual word vector is averaged to create the context vector. CBOW is typically the default choice in many applications, but it has limitations when dealing with infrequent words. Since the focus lies on predicting the most probable word to fill a blank spot by examining its surroundings, frequently occurring words are favored, while rarer ones may be overlooked.

- Skip-gram (SG): In this variant of the Word2Vec algorithm, the roles of the target word and context word are reversed. Instead of predicting what is missing from a given context like before, SG aims to predict the context in which a given word is most likely to appear. Therefore, even the infrequent words stand an equal chance of consideration. It also treats each "target+context word" pair as a distinct feature instead of just averaging them out. Due to the stated reasons, SG excels with infrequent/rare words and with smaller datasets but has the drawback of longer training times.

4.4.2 Global Vectors (GloVe)

Pennington et al., through their work in [27], highlighted a drawback of the Word2Vec algorithm and proposed GloVe to overcome it. They stated that Word2Vec is unable to fully tap into the statistical insights offered by text inputs, such as word co-occurrence patterns and frequencies. The GloVe algorithm leverages the matrix factorization and a loss function centered on the ratios of co-occurrence probabilities among words to address this issue. Though the work done in [28] shows that the statistical information used by GloVe is what Word2Vec uses, Word2Vec and GloVe are really variations of the same fundamental technique [29]. GloVe is an unsupervised learning algorithm for obtaining vector representations of English words. Training is done on global data (word-word), then the result is shown in vector space in a linear structure. We used the pre-trained GloVe model in order to obtain embeddings from GloVe in our project. GloVe was chosen because of its large dataset training capability and the high quality of the embeddings generated. Unlike Word2Vec, which primarily uses local context information, GloVe combines both global (the overall statistics of the corpus) and local (individual context) statistical information of words. Consequently, a greater range of word relationships can be recorded.

4.4.3 TF-IDF

Term Frequency-Inverse Document Frequency (TF-IDF) is another popular and widely used algorithm within the domain of embeddings. TF-IDF is a numerical metric that tells us the significance of a word in a document. The TF-IDF increases the value in direct proportion to the number of times a word appears in the document. Two parts make up this: IDF (Inverse Document Frequency) is calculated by dividing the total number of documents by the total number of documents that contain the word. TF (Term Frequency) is the frequency of a word in a document. Because TF-IDF vectorization strikes a remarkable balance between simplicity and efficacy, it can determine the relative importance of various factors; it is a helpful tool for optimizing search engines, text mining, document clustering, and many other NLP and information retrieval applications. The computation of TF-IDF involves the multiplication of term frequency (TF), indicating the frequency of a word within a document, and inverse document frequency (IDF) to reduce the weight of frequently occurring words across the entire corpus. TF-IDF has found widespread application in various NLP tasks such as Information Retrieval and text mining. Given its usefulness in highlighting significant terms within documents, we use TF-IDF, in addition to GloVe, for our experiments.

5 EXPERIMENT

With almost all the work done in the previous section, 3, in this section, we discuss the unsupervised learning method and aspect extraction using unsupervised learning. Since there are many libraries available for unsupervised learning, we will be using this third party library to create clusters to detect the aspects.

The aspect detection step is done by passing the words/ sentence embeddings and the review text as input. Then, these embeddings and reviews are assigned labels according to the problem set. The next step is to pass through the dimensionality reduction to convert it into a single vector.

A cluster is created if the vectors are close to each other. In a well-trained model, the embeddings and vectors for words, phrases, and sentences that are similar to each other will be relatively close to each other .

The final step is evaluating these clusters, for which many methods are available, but we create the tables to evaluate these clusters. Each table will be created based on embedding, the cluster to be formed, and which cluster method to be used. This table will show us how each cluster distinguishes the reviews based on these aspects. This approach will help us understand the different characteristics of each cluster and how they relate to specific parts of the reviews.

5.1 K-Nearest Neighbors (KNN)

The K-Nearest Neighbors (KNN) algorithm with Natural Language Processing (NLP) is an interesting approach. KNN is a machine learning algorithm that is mainly used for classification tasks. It is a straightforward, easy-to-understand, and adaptable algorithm that does not require much training. It classifies the points according to the distance from its neighbors. We use embeddings as the input, and based on the value of K, the number of neighbors to be considered, we classify each point into one of the classes. We classify each cluster, identify the points that are closest to each of them, and assign them to that cluster. We use KNN because it does not need to be trained and has a simple design. For this experiment, we tried it with 4,6 and 12 clusters.

5.2 K-Means Clustering

The K-mean clustering algorithm divides the n points into k parts, with each point belonging to a single cluster, with the nearest point being the cluster's mean. The mean is calculated by averaging the distance between each point in the cluster. K-means is very similar to KNN, as in KNN, when k is 1, KNN behaves like the k-means algorithm. However, KNN is a supervised learning algorithm, whereas K-means is unsupervised learning.

5.3 K-Medoids Clustering

The K-Medoids clustering algorithm is quite similar to the K-means algorithm. Both K-Means and K-Medoids divide the datasets and attempt to cluster the data by minimizing the distance between each point and center of its cluster and assigning labels to each point. K-medoids use actual points as the center, which allows a greater understanding of a cluster in comparison to K-means, where the center may or may not be an actual point. The K-medoids use arbitrary dissimilarity measures to find the pairwise distance between 2 points. We use K-medoids to compare it with K-means.

5.4 Hierarchical Clustering Algorithms

Popular hierarchical clustering algorithms such as DBSCAN [30] and OPTICS [31] were used. They create clusters based on the naturally occurring dense regions. No input regarding the number of clusters and size is given; the algorithm will assemble all the points in/under a certain distance/threshold. Like k-means, each point is assigned to a cluster depending on its position. We hope that these cluster methods will result in better results. One constraint is that they cannot handle the large dimension dataset, so we trained them on 50d and 100d datasets.

5.5 Gaussian Mixture Models (GMMs)

Gaussian Mixture Models (GMM) is a type of soft clustering method that assumes the data is generated from a mixture of several Gaussian distributions with unknown parameters. Unlike k-means, GMMs do not require their sizes, but the number of clusters is defined. Instead, GMMs identify clusters based on the likelihood that data points belong to one of several Gaussian distributions. The parameters of a Gaussian mixture model can be estimated using the expectation-maximization (EM) algorithm, in which we iteratively assign data points to clusters and update the clusters' parameters. GMMs can identify complex cluster shapes since each component can have its covariance structure, enabling the model to capture elongated or rotated clusters that k-means could miss.

GMM is suitable for capturing arbitrary-shaped clusters in scenarios where the data distribution is expected to be multi-modal. In natural language processing, GMMs can be used to cluster word embeddings, grouping words with similar meanings or in similar contexts. High-dimensional embeddings, such as those obtained from deep learning models, can be reduced to lower dimensions using techniques like PCA to alleviate computational demands before applying GMM.

5.6 Gaussian Hyperbolic Mixture Model

Gaussian Hyperbolic Mixture Model (GHMM) [32] is a technique deployed in clustering and statistical modeling when the data shows heavy tails and skewness. These models integrate the Gaussian mixture models (GMMs) adaptability with the capacity to represent data that is not properly captured by the normal distribution due to the inclusion of the hyperbolic distribution component. The GHMM can be divided into two parts:- Gaussian hyperbolic distribution and Mixture models. The Gaussian hyperbolic distribution is a dynamic distribution system capable of accurately representing the modeling asymmetries and heavier tails than the normal distribution. Mixture models are a statistical method for picturing the existence of several subgroups within a larger population without dividing the data into separate subsets for each subgroup. Every element of a mixture model represents a distinct subgroup. GHMM is the idea of Gaussian mixture models using Gaussian hyperbolic distributions instead of Gaussian distributions, which allows the model to effectively capture more complex data structures.

5.7 Latent Dirichlet Allocation(LDA)

In the context of natural language processing (NLP), “LDA” stands for Latent Dirichlet Allocation [24]. It is used for topic modeling, which is the process of identifying topics or themes present in a collection of documents. Imagine we have a pile of articles, and we want to know what they are all about without reading each one. LDA can help with this. LDA works by assuming that each document in the corpus is a mixture of a small number of topics and that each word in the document is attributable to one of the document’s topics. The model then aims to learn the distribution of words in each topic and the distribution of topics in each document. By applying LDA to a corpus of text documents, one can uncover the underlying structure of the documents in terms of the topics they cover. LDA helps with grouping documents together, summing them up quickly, or finding more documents on the same topic. Overall, LDA is a widely used and powerful tool for understanding and analyzing large collections of text data.

Figure 2 shows the architecture of the whole process, initializing by fetching the data from Amazon and passing through the preprocessing layer. Data goes through sentiment analysis to check if the data is positive or negative. Following that, we generate the data embeddings and train our model to identify the aspects. In order to verify the functionality of our models, we send the data to processing and sentiment analysis. Subsequently, we extract characteristics from the data and obtain them as our output. The left side of figure 2 depicts our training method, in which the aspects are being fed into the model for training purposes so that it can identify the aspects properly when we test. The right side of the figure shows the testing part.

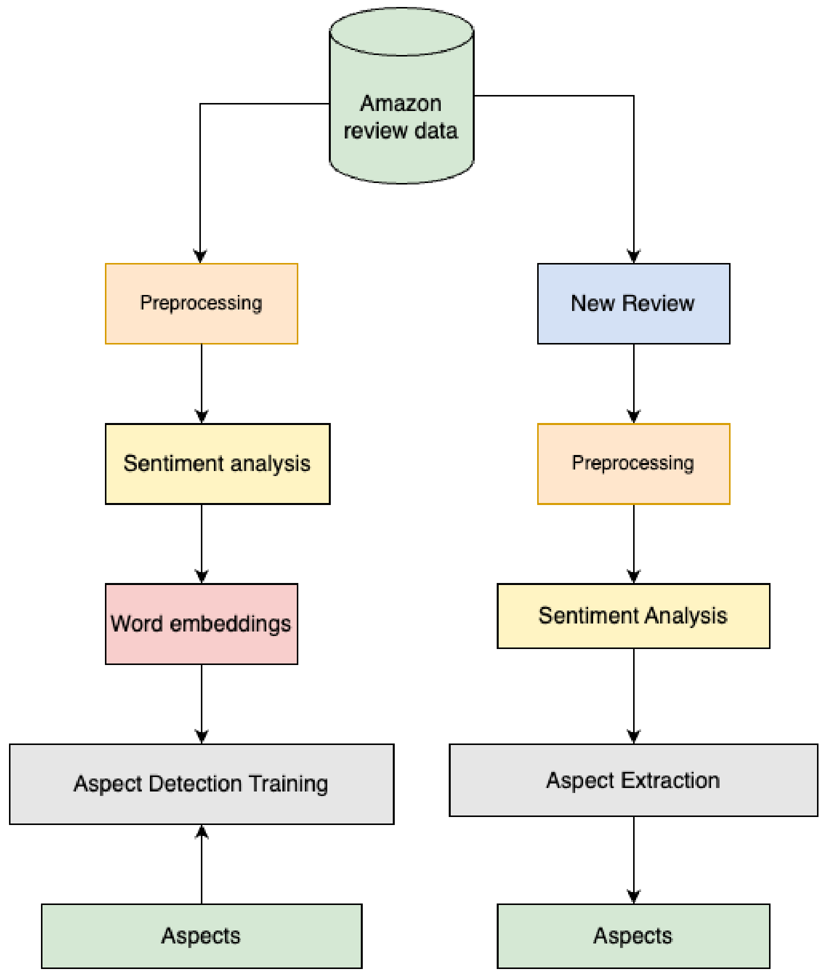

Fig. 2. Block Diagram Depicting the approach

6 RESULTS

6.1 Overview

After extensive experimentation, our result had much variance across each experiment. The aspects extraction process had:-

- Two datasets
- Three embedding models
- All clustering methods(x6)
- All number aspects method(x4)

This section will discuss the results obtained from all these combinations. The corresponding results for the k-means Clustering and the Mixture models are relegated to the Appendices, respectively.

Using sentiment analysis could have produced better results as we consider that people generally write reviews when they are dissatisfied, or the product is not to their liking. So, our hypothesis was that the number of negative reviews would be more than positive reviews, which, in this case, would give us more insight into each customer's needs. However, when we did the sentimental analysis, we got around 10% reviews as negative, and the rest were positive. This result rejected our hypothesis, so if sentiment analysis is to be used in this experiment, the negative reviews should have been above 40% of the total reviews, which might have had a better result.

6.1.1 Overview of Performance Trends

6.1.1.1 Across Aspect Detectors: The following trends are noteworthy and generally hold across all experiments.

- The Mixture models Outperform the k-means.
- The hierarchical clustering methods (DBSCAN, OPTICS) fail as they assign the entire cluster to a single or couple of cluster clusters.
- The K-medoids do not perform well, but they can detect several aspects accurately.

Refer to section 6.4 for more information about the result.

6.1.1.2 Across Number Cluster: Cluster 6 performs better in all cases but in some cases, such as when using TFIDF with means, the performance of 12 clusters outperforms 6. For GHMM on GloVE, cluster 4 has the best values among them.

Refer to section 6.4 for more information about the result.

6.1.1.3 Across Word Embeddings: it is tough to determine which embedding algorithm performs the best, but we have observed some trends, such as

- TFIDF vectors had the worst performance
- GloVe embeddings usually outperform any other embedding algorithm

Refer to 6.3 for complete results.

6.2 Analysing Performing on Sentiment Analysis Embedding Models

The model's performance concerning sentiment analysis differs significantly from the model without sentiment analysis. The clusters formed using sentiment analysis can be seen in figure 3. Representing the result in the 2-D dimension is hard. We used tSNE for the dimensionality reduction method. To reduce the dimension from 100d to 2d, we used tSNE(T-distributed Stochastic Neighbor Embedding.), which is used to reduce the dimension. tSNE converts the similarities between data points into joint probabilities and aims to minimize the Kullback-Leibler divergence between the joint probabilities of the low-dimensional embedding and the high-dimensional data [33]. tSNE will generate its own axis to label the data tSNE Dimesion 1 and 2 as shown in figure 3.

We can see how the cluster is forming and how hard it is to distinguish each cluster. The main use of sentiment analysis is to check if the negative reviews can provide some specific insight as to why people do not like a product. Another reason we thought sentiment analysis could provide us with some help as general human text is because people will mostly write reviews if the product does not work as per their expectations. However, as seen in the figure, more positive and negative reviews reject our hypothesis.

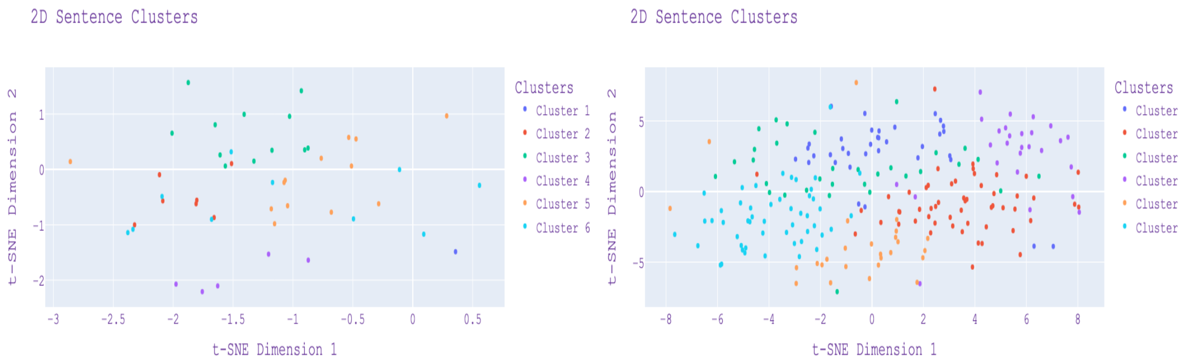

Fig. 3. Sentiment Analysis Clustering

6.3 Analysing Performance on Word Embedding Models

Examining the outcomes of various word embedding algorithms on particular tasks is a fascinating investigative objective. Each algorithm has distinct advantages and disadvantages, which makes it challenging to choose the superior method. Although one may excel in a specific task, one may struggle in other areas.

Outperforming models apply the GloVe Embedding for each of the three word-embedding techniques. In both the Amazon Validation and eco-friendly goods datasets, Word2Vec performs noticeably worse. However, TF-IDF performs poorly on the Amazon Validation dataset but excels in the eco-friendly issue set. These observations illustrate notable departures from the usual, needing additional research into the underlying causes.

With the exception of the previously noted anomalies, performance trends are relatively consistent throughout embedding techniques. Even though the product categories covered in the Amazon Validation Product and eco-friendly product jobs are very different, there are differences in performance when the cluster settings are adjusted.

6.4 Results with Clustering Models

This section will discuss the analysis of results based on cluster number. We tested three, four, six, and twelve cluster numbers to optimize the clustering approach for aspect extraction from reviews. The number of clusters was predefined to significantly impact the model's capacity to identify and classify different features in the data.

Applying 3 and 4 clusters resulted in data that showed a merger of thematic content, with various features combined into a single segment. This could be due to various regions, but the primary reason is fewer clusters than aspects. Raising the number of clusters to 12 was expected to produce a finer analysis, which it did for some cases. Mostly, using 12 with Tf-IDF, clustered most of the data in 2,3 clusters. On the other hand, the best illustration of aspects was obtained by applying six clusters as seen in figure 4 and figure 5 using tSNE to reduce the dimensions. The model performed flawlessly, with each cluster exhibiting a distinct theme grouping. This compromise allowed for a thorough granularity without being disorganized and instructive without reductive.

The analysis findings obtained after employing six clusters are displayed in figure 6 and figure 7. The distribution of items within each cluster is visible to us. Furthermore, we may infer from figure 6 that GloVe and GMM produce the most various classifications across the number of characteristics. This indicates that combining GloVe embeddings with a Gaussian Mixture Model clustering algorithm produces a rich and detailed data segmentation. This is because GloVe captures the feature context well, and GMM models the data points as mixtures of multiple Gaussian distributions. This method appears especially useful in differentiating between features, like performance, when many pieces congregate in a specific cluster.

In summary, the arrangement of six clusters proved to be the most effective in striking a balance between discernibility and detail, resulting in the most logical and valuable classification of features for this specific corpus.

6.5 Aspect Extraction Models that Failure to Perform

Throughout the experiment, very few aspect-based extraction models produced good results. The cause for lousy performance could differ from model to model. We will discuss the reason for this performance through our findings.

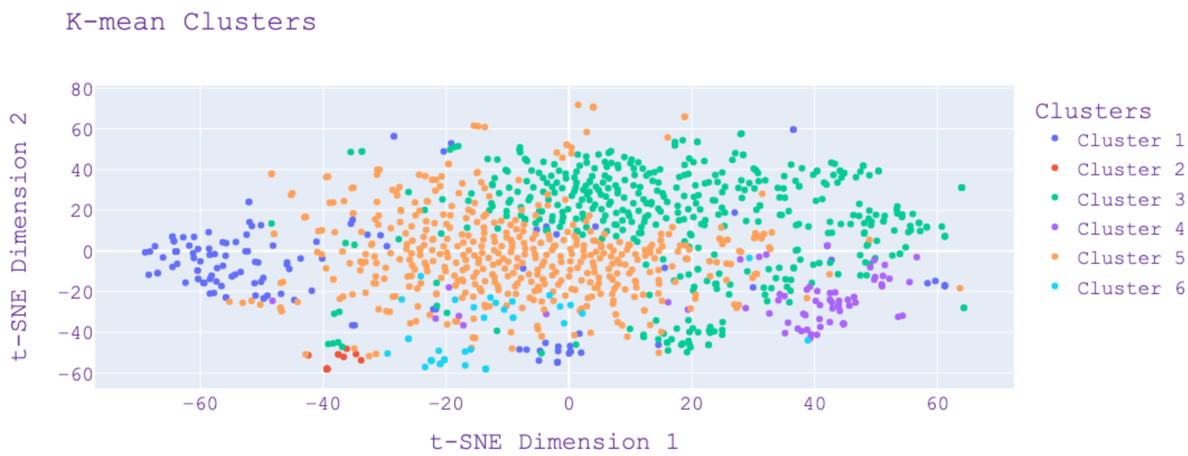

Fig. 4. Kmean cluster with 6 clusters

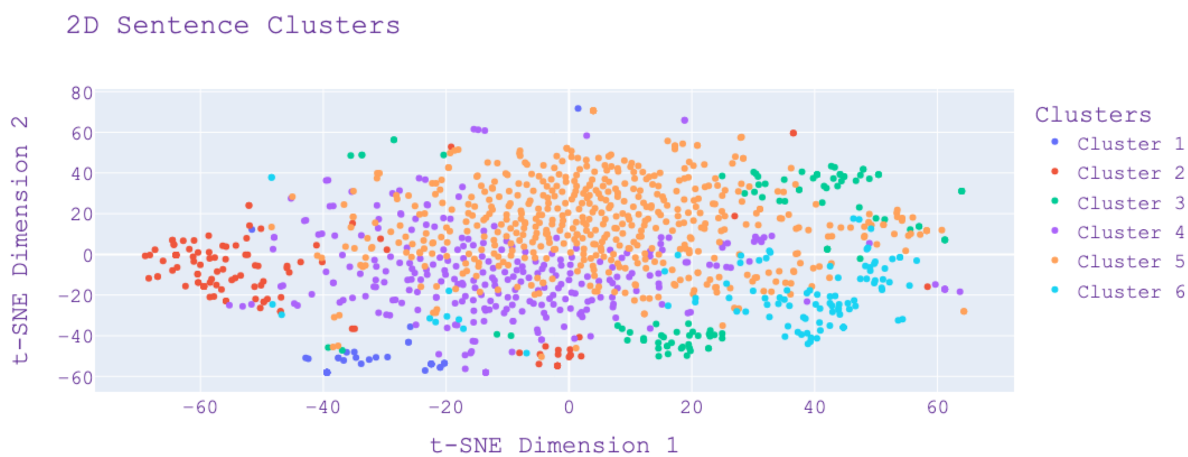

Fig. 5. GMM with 6 clusters

6.5.1 The Poor Performance of k-Means Detectors

We chose the k-Means algorithm for its simplicity and purported effectiveness, but it failed to deliver satisfactory results in aspect extraction through word clustering. We explored two variants: kmeans++ and cosine similarity distance. Both variants underperformed,

	0	1	2	3	4	5
adaptability	1	0	24	0	50	4
durability	2	21	83	1	91	19
ease use	54	0	18	0	28	0
ergonomics	29	0	13	2	27	3
interference	1	0	19	6	52	2
performance	2	1	135	4	92	11
use efficiency	4	0	23	0	23	0
aesthetics	5	0	45	56	38	2
ease reprocessing	34	0	16	1	42	1
ease storage	2	0	3	0	7	0
price	6	0	58	0	10	0
safety	2	0	1	0	7	0

	0	1	2	3	4	5
adaptability	23	46	1	0	7	2
durability	88	73	2	32	19	3
ease use	18	33	1	0	2	46
ergonomics	17	23	15	0	6	13
interference	23	46	9	0	2	0
performance	110	87	6	1	19	22
use efficiency	24	23	0	0	0	3
aesthetics	32	31	74	0	3	6
ease reprocessing	17	40	2	0	1	34
ease storage	4	6	0	0	1	1
price	60	5	1	0	0	8
safety	4	4	0	0	0	2

Fig. 6. GloVe embedding with K-mean on the left and GMM on the right, Rows are the actual aspects and columns are the cluster predicted

	0	1	2	3	4	5
Adaptability	1	3	9	0	65	1
Durability	9	2	6	29	164	7
Ease of Use	1	0	1	0	98	0
Ergonomics	1	3	0	0	56	14
Interference	3	1	1	0	72	3
Performance	4	24	36	1	177	3
Use Efficiency	6	0	0	0	44	0
Aesthetics	1	10	9	0	110	16
Ease of Reprocessing	3	0	1	0	89	1
Ease of Storage	0	0	1	0	11	0
Price	1	12	11	0	50	0
Safety	0	0	0	0	10	0

	0	1	2	3	4	5
Adaptability	1	7	0	0	70	1
Durability	0	5	29	26	157	0
Ease of Use	1	0	0	0	48	51
Ergonomics	3	0	0	0	70	1
Interference	0	1	0	0	79	0
Performance	3	35	1	1	205	0
Use Efficiency	0	0	0	0	48	2
Aesthetics	4	8	0	0	130	4
Ease of Reprocessing	0	1	0	0	64	29
Ease of Storage	0	1	0	0	10	1
Price	0	11	0	0	62	1
Safety	0	0	0	0	10	0

Fig. 7. TF-IDF embedding with K-mean on the left and GMM on the right, Rows are the actual aspects and columns are the cluster predicted

reflecting poor aspect extraction capabilities in reviews. Models often mistakenly map multiple aspects to a single cluster, leading to significant precision issues due to its properties. These findings suggest a fundamental problem with the k-means approach, particularly in handling asymmetrical data clusters and complex cluster hierarchies.

6.5.2 *The Failure of DBSCAN and OPTICS Implementations*

Our experiments with DBSCAN and OPTICS, hierarchical clustering algorithms for detecting the shape, size, and structure based on the dense regions in feature spaces, also experienced some problems. Primarily, these models tended to group most word vectors into a single cluster or overlooked them as noise, and despite their potential for deep, insightful aspect extraction and the ability to discover new, domain-specific aspects, the practical limitations—such as immense computational demands, hampered their effectiveness.

Both experiments highlight the challenges in applying traditional clustering algorithms to aspect extraction in text reviews. The issues arise from the algorithms' assumptions about data symmetry and cluster distribution, which do not hold in complex, real-world text data.

6.6 Key Takeaways and Distilling Our Observations

- **Challenges with several clusters**: While building models with large clusters performs better, the performance will depend on other factors, like the embedding method and clustering algorithm.
- **Limitations of k-Means Clustering**: k-Means clustering struggles to form meaningful clusters around aspects. Although cosine similarity initialization shows some improvements over kmeans++ initialization, it highlights the challenge of catching aspect-related words in embeddings.
- **Failures of Hierarchical Clustering**: Hierarchical clustering methods like DBSCAN and OPTICS could have performed more effectively but failed to cluster the embedding spaces. Points to their limitations in handling the specific complexities of word embeddings.
- **Impact of Hyperparameter Settings**: The overall performance of clustering methods suggests that incorrect hyperparameter settings could hinder their success.
- **Influence of Domain Semantics**: The initial hypothesis is that the training text's domain semantics significantly influence models' performance in topic-sensitive tasks using word embeddings.

- **TF-IDF Embedding in GHMM**: THE GHMM could not produce results with TF-IDF vectors. After analyzing, we concluded that it works with word vectors rather than sentence vectors, or TF-IDF has many dimensions, which it could not support.

7 FUTURE WORK

In this study, we experimented with integrating word embedding and clustering methodologies with substantial promise for product design enhancement. While this provided us with good results, other word embedding techniques such as GTE-Small and E5-Small can be used to build our clusters. We can also explore other clustering techniques, such as Birch and Fuzzy C-Mean, on the embeddings produced to see if they yield better results. The dataset used in this study is domain-specific so that this dataset can be combined with other publicly available review datasets. This will help increase the training dataset's size and diversity, making the model robust to new and unseen data. New datasets provide additional features that could provide deeper insights and improve model performance.

Our research focused on implementing various clustering algorithms; unsupervised neural network models can also be explored in the future. Autoencoders, Restricted Boltzmann Machines, and GANs are examples of deep networks that can generate clusters in an unsupervised setting. A critical future focus will be refining our algorithms to determine the optimal number of clusters, allowing for improved data segmentation and more insightful analysis. The elbow method, silhouette score, and Davies–Bouldin Index are some methods that can be used to accomplish this task.

To support the growth of our application, we plan to conduct scalability tests using Apache Spark to ensure that our models can handle increased data volumes effectively. We also plan to create an automated pipeline using a cloud-based system such as AWS for quick and easy training and testing our model whenever the new data arrives. Using it will ensure that the model is scalable and easy to implement. Using the AWS pipeline and distributed environments such as Spark will also help optimize the model performance by reducing the training time when new and more enormous datasets are used.

Finally, we also plan to perform an ethical review to identify and mitigate any potential biases in the model, ensuring fair and unbiased outcomes.

8 CONCLUSIONS

Our analysis has demonstrated that integrating word embedding and clustering methodologies holds significant promise for product design enhancement. Applying these techniques to the complex realm of product development enables an understanding of customer feedback and market trends.

Among the various models evaluated, it has become evident that combining GloVe embeddings with fuzzy clustering methods, specifically mixture models, outperforms its counterparts. This pairing has proven adept at capturing the semantic structure within the customer review. It has emerged as the most effective tool for aligning product features with customer expectations.

It is essential to acknowledge that the current conclusions are drawn from a dataset confined to a single domain—Amazon product reviews—which is the limitation of our scope. Our ongoing and future research is expanding to include multiple domains. Merging multiple domains will test the model's versatility and refine its accuracy by exposing it to a broader range of expressions and sentiment complexities.

In conclusion, while our findings are promising and suggest a new horizon for product design through advanced NLP techniques, we remain committed to validating these results across a broader spectrum of domains. Our dedication to testing and domain expansion is critical to unlocking the full potential of these methodologies in product development and beyond.

Literature Cited

[1] B. Liu, "Sentiment analysis and opinion mining," 2012.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[3] D. Mimno, H. M. Wallach, E. Talley, M. Leenders, and A. McCallum, "Optimizing semantic coherence in topic models," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '11*, (USA), pp. 262–272, Association for Computational Linguistics, 2011.

[4] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[5] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems*, pp. 3111–3119, 2013.

[6] R. He, W. S. Lee, H. T. Ng, and D. Dahlmeier, "Exploiting document knowledge for aspect-level sentiment classification," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, (Melbourne, Australia), pp. 579–585, Association for Computational Linguistics, 2018.

[7] A. Mokadam and M. A. Suresh, "Dossad - leveraging domain specific semantics in aspect detection for product design," in *2020 IEEE 14th International Conference on Semantic Computing (ICSC)*, pp. 249–252, 2020.

[8] A. Rietzler, S. Stabinger, P. Opitz, and S. Engl, "Adapt or get left behind: Domain adaptation through bert language model finetuning for aspect-target sentiment classification," in *Proceedings of the 12th Language Resources and Evaluation Conference*, (Marseille, France), pp. 4933–4941, European Language Resources Association, 2020.

[9] V. Raunak, V. Gupta, and F. Metze, "Effective dimensionality reduction for word embeddings," in *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, (Florence, Italy), pp. 235–243, Association for Computational Linguistics, 2019.

[10] D. K. Gupta and A. Ekbal, "Iitp: Supervised machine learning for aspect based sentiment analysis," in *Proceedings of the 8th International Workshop on Semantic Evaluation*

(SemEval 2014), (Dublin, Ireland), pp. 319–323, Association for Computational Linguistics, 2014.

[11] R. He, W. S. Lee, H. T. Ng, and D. Dahlmeier, "An unsupervised neural attention model for aspect extraction," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (R. Barzilay and M.-Y. Kan, eds.), (Vancouver, Canada), pp. 388–397, Association for Computational Linguistics, July 2017.

[12] H. T. Nguyen and M. Le Nguyen, "Effective attention networks for aspect-level sentiment classification," in *2018 10th International Conference on Knowledge and Systems Engineering (KSE)*, pp. 25–30, 2018.

[13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, pp. 1735–1780, 1997.

[14] B. Wang and M. Liu, "Deep learning for aspect-based sentiment analysis," 2016. Citation details need to be updated once the source details are available.

[15] D. Tang, B. Qin, X. Feng, and T. Liu, "Effective lstms for target-dependent sentiment classification," in *Proceedings of the International Conference on Computational Linguistics (COLING)*, 2016.

[16] M. Zhang, Y. Zhang, and D.-T. Vo, "Gated neural networks for targeted sentiment analysis," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2016.

[17] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory, COLT '92*, 1992.

[18] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of the Eighteenth International Conference on Machine Learning, ICML '01*, (San Francisco, CA, USA), pp. 282–289, Morgan Kaufmann Publishers Inc., 2001.

[19] K. Schouten and F. Frasincar, "Survey on aspect-level sentiment analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, pp. 813–830, 2016.

[20] M. K. S. Vadla, M. A. Suresh, and V. K. Viswanathan, "Enhancing product design through ai-driven sentiment analysis of amazon reviews using bert," *Algorithms*, vol. 17, no. 2, p. 59, 2024.

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

[22] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *Advances in neural information processing systems*, vol. 32, 2019.

[23] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-xl: Attentive language models beyond a fixed-length context," *arXiv preprint arXiv:1901.02860*, 2019.

[24] H. Jelodar, Y. Wang, C. Yuan, *et al.*, "Latent dirichlet allocation (lda) and topic modeling: Models, applications, a survey," *Multimedia Tools and Applications*, vol. 78, pp. 15169–15211, 2019.

[25] C. J. Hutto and E. Gilbert, "Vader: A parsimonious rule-based model for sentiment analysis of social media text," *ICWSM*, vol. 8, no. 1, pp. 216–225, 2014.

[26] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *ArXiv*, 2019.

[27] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP 2014)*, vol. 12, 2014.

[28] O. Levy and Y. Goldberg, "Neural word embedding as implicit matrix factorization," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, NIPS'14*, (Cambridge, MA, USA), pp. 2177–2185, MIT Press, 2014.

[29] S. Arora, Y. Li, Y. Liang, T. Ma, and A. Risteski, "A latent variable model approach to pmi-based word embeddings," *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 385–399, 2016.

[30] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD'96*, pp. 226–231, AAAI Press, 1996.

[31] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "Optics: Ordering points to identify the clustering structure," in *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data, SIGMOD '99*, (New York, NY, USA), pp. 49–60, Association for Computing Machinery, 1999.

[32] C. Tortora, R. P. Browne, A. ElSherbiny, B. C. Franczak, and P. D. McNicholas, "Model-based clustering, classification, and discriminant analysis using the generalized hyperbolic distribution: Mixghd r package," *Journal of Statistical Software*, vol. 98, no. 3, pp. 1–24, 2021.

[33] L. J. P. van der Maaten and G. E. Hinton, "Visualizing high-dimensional data using t-sne," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

Appendix A

K-MEAN

This section shows the result which we did on all the embedding for Kmeans. 8-13 shows LDA results on each of the embedding models.14 and 15 shows the result of 12 cluster models.

```
Cluster 0: well, good, durable, sturdy, comfortable
Cluster 1: great, work, price, looks, works
Cluster 2: easy, use, clean, install, wash
```

Fig. 8. LDA for word2vec 3 cluster

```
Cluster 0: like, clock, shower, use, timer
Cluster 1: away, bottom, strap, guitar, feels
Cluster 2: durable
```

Fig. 9. LDA for TFIDF

```
Cluster 0: concentrator, beginning, lip, smells, steel
Cluster 1: functional, directly, thread, brown, smudge
Cluster 2: corner, interchangable, obstruction, turning, unusable
```

Fig. 10. LDA for GloVe

```
Cluster 0: easy, clean, convenient, handy, little
Cluster 1: settings, hair, perfect, types, lots
Cluster 2: stay, also, diffuser, heating, element
Cluster 3: durable, fragile, flimsy
Cluster 4: great, work, good, desk, daily
Cluster 5: attractive, great, hair, pretty, wobbly
```

Fig. 11. LDA for word2vec 6 cluster

```
Component 0: handle, comfort, evident, upon, playability
Component 1: great, work, looks, quality, works
Component 2: good, quality, product, looks, fabric
Component 3: price, great, good, reasonable, quality
Component 4: easy, use, clean, install, wash
Component 5: well, durable, sturdy, comfortable, nice
```

Fig. 12. LDA for Tf-IDF 6 cluster

```
Component 0: holders, load, value, ice, clips
Component 1: irish, sensitive, shoulder, gave, drink
Component 2: toys, hassle, connections, scuffs, bin
Component 3: boiled, jet, trunk, slower, note
Component 4: straps, much, wo, copper, seems
Component 5: office, long, tears, fail, machine
```

Fig. 13. LDA for GloVe 6 cluster

	0	1	2	3	4	5	6	7	8	9	10	11
adaptability	0	6	38	26	0	0	0	1	6	1	0	1
durability	0	7	49	84	0	21	0	5	30	1	11	9
ease use	0	4	26	21	0	0	0	1	1	46	0	1
ergonomics	21	10	16	13	0	0	0	0	8	2	1	3
interference	0	15	39	19	0	0	0	0	0	0	0	7
performance	1	27	64	54	1	1	6	50	3	0	28	10
use efficiency	0	5	19	22	0	0	0	1	0	3	0	0
aesthetics	0	38	22	25	37	0	0	6	2	5	6	5
ease reprocessing	0	5	39	19	0	0	0	0	0	30	0	1
ease storage	0	2	3	4	0	0	0	1	1	1	0	0
price	0	5	5	19	0	0	0	1	1	2	41	0
safety	0	0	4	3	0	0	0	0	0	2	0	1

Fig. 14. Evaluation result for 12 cluster for Glove

	0	1	2	3	4	5	6	7	8	9	10	11
Adaptability	0	6	4	0	1	2	0	0	65	1	0	0
Durability	0	5	1	29	4	15	26	1	136	0	0	0
Ease of Use	0	0	0	0	1	2	0	0	50	47	0	0
Ergonomics	1	0	3	0	3	1	0	11	27	1	27	0
Interference	5	1	1	0	3	2	0	0	68	0	0	0
Performance	0	35	23	1	9	27	1	0	148	0	1	0
Use Efficiency	0	0	0	0	5	0	0	0	44	1	0	0
Aesthetics	0	8	10	0	27	1	0	0	81	4	0	15
Ease of Reprocessing	0	1	0	0	2	7	0	0	55	29	0	0
Ease of Storage	0	1	0	0	0	0	0	0	10	1	0	0
Price	0	11	11	0	1	3	0	0	48	0	0	0
Safety	0	0	0	0	0	0	0	0	10	0	0	0

Fig. 15. Evaluation result for 12 cluster for Tf-IDF

Appendix B

MIXTURE MODELS

This section shows the result which we did on all the embedding for Mixture Models. 18-23 shows LDA results on each of the embedding models.16 and 17 shows the result of 12 cluster models.

	0	1	2	3	4	5	6	7	8	9	10	11
adaptability	0	1	2	30	0	0	2	43	0	1	0	0
durability	0	8	12	48	0	21	6	103	7	1	11	0
ease use	0	0	1	19	0	0	0	31	0	49	0	0
ergonomics	0	2	1	13	7	0	0	25	0	3	0	23
interference	0	0	1	28	7	0	0	44	0	0	0	0
performance	7	39	47	58	3	1	2	81	0	0	6	1
use efficiency	0	0	2	10	0	0	0	29	5	4	0	0
aesthetics	4	6	17	16	50	0	0	47	1	5	0	0
ease reprocessing	0	0	5	31	1	0	1	23	3	30	0	0
ease storage	0	0	1	2	0	0	1	7	0	1	0	0
price	35	4	2	4	0	0	1	27	0	1	0	0
safety	0	0	0	1	0	0	0	7	0	2	0	0

Fig. 16. Evaluation result for 12 cluster for GloVe

	0	1	2	3	4	5	6	7	8	9	10	11
Adaptability	1	1	2	7	54	0	0	6	0	1	3	4
Durability	0	0	19	7	153	0	0	12	1	15	2	8
Ease of Use	1	48	1	0	44	0	0	1	0	0	0	5
Ergonomics	1	1	1	0	29	27	0	12	1	0	2	0
Interference	3	0	3	1	53	0	0	8	1	3	1	7
Performance	0	1	25	32	125	1	0	8	25	1	14	13
Use Efficiency	1	2	0	0	44	0	0	0	0	0	0	3
Aesthetics	1	4	1	8	95	0	1	25	1	0	9	1
Ease of Reprocessing	0	29	7	1	53	0	0	3	0	0	0	1
Ease of Storage	2	1	0	1	7	0	0	0	0	0	0	1
Price	0	0	3	11	40	0	0	2	5	0	10	3
Safety	0	0	0	0	9	0	0	0	0	0	0	1

Fig. 17. Evaluation result for 12 cluster for Tf-IDF

```
Component 0: easy, well, good, use, clean
Component 1: great, price, work, looks, quality
Component 2: sturdy, durable, absorbent, made, well
```

Fig. 18. LDA for word2vec 3 cluster

```
Cluster 0: easy, clean, convenient, use
Cluster 1: lots, settings, without, frizz, dryer
Cluster 2: stay, also, diffuser, heating, element
```

Fig. 19. LDA for Tf-IDF 3 cluster

```
Component 0: potting, basically, traveled, friendly, keyboard
Component 1: cheery, responsive, highly, aesthetically, refund
Component 2: waits, times, juice, aware, solution
```

Fig. 20. LDA for Glove 3 cluster

```
Cluster 0: directly, functional, beginning, pulled, controls
Cluster 1: interchangable, builds, construction, rinsing, husband
Cluster 2: corner, concentrator, rigid, fabrics, cheaply
Cluster 3: lip, steel, smells, dislodged, cloudy
Cluster 4: ok, savings, erased, cheap, smudge
Cluster 5: obstruction, unlikley, falling, started, fits
```

Fig. 21. LDA for word2vec 6 cluster

```
Cluster 0: colors, looks, gorgeous, true, picture
Cluster 1: durable
Cluster 2: well, still, level, works, one
Cluster 3: sturdy, perfectly, cupholders, fit, easy
Cluster 4: like, clock, use, shower, timer
Cluster 5: easy, use, clean, handle, excellent
```

Fig. 22. LDA for Tf-IDF 6 cluster

```
Cluster 0: nice, compact, size, fan, weight
Cluster 1: comfortable, hold, super, carry, strap
Cluster 2: great, price, work, looks, works
Cluster 3: use, easy, install, difficult, fabrics
Cluster 4: good, well, durable, sturdy, quality
Cluster 5: easy, clean, wash, super, install
```

Fig. 23. LDA for GloVe 6 cluster