A Comprehensive Evaluation of GloVe, Word2Vec, and Custom Embeddings for Sentiment Analysis of Amazon Fine Food Reviews

Divya Naika

CSIT697-Master's Project
Montclair State University

Abstract

Sentiment analysis is a critical task in natural language processing, providing valuable insights into public opinion and emotional trends. This project focuses on the Amazon Fine Food Reviews dataset, leveraging advanced text representation techniques and machine learning models to predict consumer sentiment[1][4]. Our methodology includes extensive preprocessing to clean and normalize the reviews, followed by feature extraction using pre-trained GloVe embeddings, Word2Vec embeddings, and custom domain-specific embeddings[15]. These embeddings serve as input to a variety of models, including simple averaging-based approaches and more advanced architectures such as LSTMs.We implement various model architectures, including basic averaging of embeddings and advanced LSTM-based deep learning networks, to analyze sentiment[].

A comprehensive comparative analysis of the models is performed using metrics such as accuracy, precision, recall, and F1-score to assess their effectiveness. The findings demonstrate the advantages of domain-specific embeddings for improving sentiment classification accuracy, while pre-trained embeddings prove versatile for general use. This research contributes to the field by providing actionable insights into embedding selection and performance optimization, with potential applications in product feedback analysis, recommendation systems, and customer relationship management[5].

1. Introduction

Sentiment analysis has emerged as a crucial tool in natural language processing (NLP), enabling organizations to understand public sentiment and make data-driven decisions. It involves the classification of text data to determine the underlying sentiment, typically categorized as positive, negative, or neutral[6]. With the explosion of user-generated content on platforms such as e-commerce websites, the ability to process and analyze large-scale textual data efficiently has become increasingly important[2].

This research focuses on the Amazon Fine Food Reviews dataset, which comprises thousands of consumer reviews offering insights into customer satisfaction and product feedback. Analyzing this dataset poses challenges due to the complexity and variability of natural language, including factors such as sarcasm, context, and sentiment intensity. Traditional approaches, such as bag-of-words or TF-IDF, often fail to capture these nuances[8].

To overcome these challenges, this study employs advanced word embedding techniques, including pre-trained GloVe and Word2Vec embeddings, as well as custom embeddings tailored to the dataset. These methods encode text into dense numerical vectors that retain semantic relationships and contextual information. The embeddings are integrated into various machine learning models, ranging from simple averaging to complex LSTM architectures, to perform sentiment classification.

The study aims to evaluate the effectiveness of these embedding techniques and identify the optimal approach for domain-specific sentiment analysis. By benchmarking performance using standard evaluation metrics such as accuracy, precision, recall, and F1-score, this research provides actionable insights for both academic and industrial applications[9]. The findings are expected to benefit industries seeking to leverage consumer feedback for product improvement, customer engagement, and strategic decision-making.

.

2. Literature Survey

The study of sentiment analysis has seen significant advancements in recent years, with various approaches and techniques emerging to address the challenges of extracting subjective information from textual data. This section reviews the key contributions and methodologies relevant to the current research, focusing on embedding techniques and machine learning models for sentiment analysis[4].

Sentiment analysis is an essential task in NLP, extensively used to analyze public opinion, customer feedback, and social media content. Early works in sentiment analysis utilized rule-based approaches and traditional feature extraction methods, such as bag-of-words (BoW) and term frequency-inverse document frequency (TF-IDF) (Pang & Lee, 2008). While effective for small-scale datasets, these methods struggled to capture semantic relationships and contextual nuances, leading to the development of advanced representation techniques[11].

The introduction of word embeddings revolutionized NLP by providing dense, distributed representations of words in a vector space[12].

Word2Vec (Mikolov et al., 2013): Word2Vec introduced the concepts of Continuous Bag of Words (CBOW) and Skip-Gram models, which learn word embeddings by predicting context words. It demonstrated superior performance in capturing syntactic and semantic relationships compared to traditional methods.

GloVe (Pennington et al., 2014): GloVe embeddings improved upon Word2Vec by leveraging global co-occurrence statistics, offering more robust representations for words with infrequent appearances. These embeddings have been widely adopted in sentiment analysis tasks due to their scalability and pre-trained availability.

Custom Embeddings: Domain-specific embeddings have gained traction as they offer improved performance for datasets with unique vocabulary or context (Tang et al., 2014). By training embeddings directly on the dataset, models can better capture domain-relevant semantics[12].

With the rise of deep learning, models such as Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks have become popular for text classification tasks. **LSTM Networks** (Hochreiter & Schmidhuber, 1997): LSTMs address the issue of long-term dependency in sequential data, making them well-suited for sentiment analysis tasks where word order plays a critical role. Studies have shown that combining LSTM networks with word embeddings significantly enhances sentiment classification accuracy (Tang et al., 2015)[13].

Ensemble Methods: The use of ensemble techniques, such as voting classifiers or stacking, has been explored to combine the strengths of different models. These methods often lead to improved robustness and generalization in sentiment prediction tasks (Opitz & Maclin, 1999)[14].

Research comparing the effectiveness of different embeddings has highlighted the trade-offs between pre-trained and custom embeddings. Pre-trained embeddings, such as Word2Vec and GloVe, provide generalizable representations suitable for a wide range of tasks. However, custom embeddings trained on domain-specific data can outperform pre-trained models in niche applications by capturing the unique semantics of the dataset (Zhou et al., 2018). Studies emphasize the importance of balancing generalizability and specificity when selecting embeddings for sentiment analysis.

The practical applications of sentiment analysis are vast, spanning domains such as marketing, customer relationship management, and public policy analysis. Businesses leverage sentiment analysis to monitor brand perception, improve product recommendations, and enhance customer satisfaction. Researchers have demonstrated that embedding techniques and deep learning models can significantly improve the accuracy of sentiment analysis systems in these applications (Medhat et al., 2014)[16].

Despite advancements, challenges remain in handling sarcasm, negation, and domain adaptation in sentiment analysis. Furthermore, while contextualized embeddings like BERT (Devlin et al., 2019) have emerged as state-of-the-art methods, their high computational requirements limit their accessibility for many real-world applications. This study addresses these gaps by comparing traditional and domain-specific embeddings, offering a balanced approach to improving sentiment classification without relying on heavy computational resources.

3. Data Description

3.1 Dataset Overview

The data set used for this project is Amazon fine food reviews dataset. This dataset contains Amazon reviews on exquisite meals. The information includes all 500,000 reviews up to October 2012 and spans more than ten years. Reviews include plain text, user and product information, ratings, and reviews. Reviews from all other Amazon categories are also included. Review data is available from October 1999 to October 2012.

Shape of data: (568454, 10)

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Summary	Text
462205	462206	B000CR00FQ	A21U03PVX582A9	Sue	0	0	1	1294444800	Burnt taste and shelf life short or used up	I was excited about buying these in bulk at th...
58432	58433	B001ULH7P4	AJB5P7GVR0MT8	Harry the Horse	0	1	2	1318809600	I eat these when I want to punish myself.	The Asian Pear nails you with an overwhelming ...
347383	347384	B000BF3AGU	A3E0ORO3Q8F6AU	JC	2	2	5	1324252800	Great Product	Vegetarian for many years. This product is gr...
522330	522331	B0030V9HV4	AIXLS4JSYH9DX	Jamie Hussey "Huss"	0	0	5	1311552000	my dogs are not that fond of the tablets but I...	I have a 9 year old a 6 year old and a 1 year ...
334021	334022	B001JSWJHE	A3318SY7C730S5	Evster	0	0	5	1350432000	Great product	Just got our package in the mail and it not on...

3.2 Dataset Composition

The reviews dataset contains a total of 568,454 reviews, which were submitted between October 1999 and October 2012. It includes feedback from 256,059 unique users, who reviewed 74,258 different products. Notably, there are 260 users who have submitted more than 50 reviews each.The data set include several key columns for each review:

- **Id**: Unique identifier for each review.
- **ProductId**: Identifier for the product being reviewed.
- **UserId**: Identifier for the user who submitted the review.
- **ProfileName**: The name of the user submitting the review.
- **HelpfulnessNumerator**: The number of users who found the review helpful.
- **HelpfulnessDenominator**: The total number of users who rated the helpfulness of the review.
- **Score**: The rating score given by the user (typically 1-5 stars).
- **Time**: Timestamp indicating when the review was submitted.
- **Summary**: A summary or title of the review.
- **Text**: The actual textual content of the review.

3.3 Data Processing and Balancing

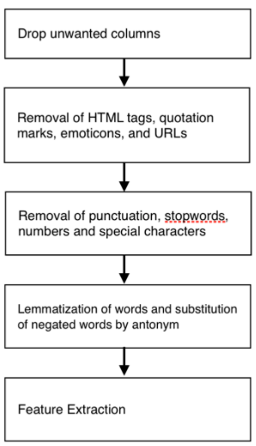
The preprocessing involves cleaning text by removing punctuation, special characters, and converting text to lowercase. The data is then tokenized into individual words, followed by the removal of stopwords to eliminate common, insignificant terms. Lemmatization is applied to convert words to their root forms, ensuring consistency. Numerical data is standardized or removed as necessary, while overly short or long reviews are managed by setting length thresholds. Finally, the cleaned and processed text is vectorized using techniques like TF-IDF or embeddings (e.g., GloVe, Word2Vec) to prepare it for machine learning models.

3.4 Acknowledgements and Resources

Julian McAuley from the University of California, San Diego, who made the dataset publicly available and can be accessed at: **Amazon Fine Food Reviews Dataset on Kaggle**

4. Data Cleaning and Preprocessing

Effective data cleaning and preprocessing phase is crucial in transforming raw text data into a clean, organized format suitable for building robust sentiment analysis models. This section elaborates on the cleaning and preprocessing steps taken in the project, highlighting the importance and implementation of each.



4.1 Initial Data Loading

Imported essential Python libraries for data manipulation and analysis. Read the dataset file (usually a .csv) into a pandas DataFrame for easy manipulation. Inspected the dataset to understand its structure, columns, and sample data. Identifed and handled missing or null values in the dataset. Selected the columns relevant for sentiment analysis (e.g., Text for reviews and Score for sentiment). Analyzed sentiments as positive, negative, or neutral, map numerical scores to sentiment categories

4.2 Preprocessing Steps

Selected the columns relevant for sentiment analysis (e.g., Text for reviews and Score for sentiment). Analyzed sentiments as positive, negative, or neutral, map numerical scores to sentiment categories

4.3 Data Cleaning

Data cleaning is the initial and a critical step in the preprocessing pipeline, focusing on removing irrelevant and noisy components from the text that could adversely affect the performance of machine learning models.

4.3.1 Case Normalization:

- Standardized the text by converting all characters to lowercase, ensuring that "Good" and "good" are treated the same

4.3.2 URLs and HTML Tags Removal:

- Eliminated any web links and HTML elements, as these do not carry sentimental value and can introduce noise into the text data.

4.3.3 Special Characters and User Mentions Removal:

- Cleaned the text by removing unnecessary characters like punctuation, digits, or symbols that don't contribute to sentiment.

4.3.4 Tokenization

- Splits the text into individual words (tokens) for further processing.

4.3.5 Stop Word Removal

- Eliminated common words (like "the," "is," "and") that do not carry significant meaning in sentiment analysis.

4.3.6 Lemmatization

- Reduced words to their root forms (e.g., "running" to "run"), ensuring consistency.

4.3.7 Joining Tokens Back into Sentences

- Converted the list of tokens back into a single string, as some models or vectorizers expect raw text input.

4.3.8 Vectorization

- Converts the processed text into numerical representations using methods like TF-IDF, GloVe, or Word2Vec.

4.3.9 Removing Very Short or Long Reviews

- Filters out reviews that are too short (e.g., one word) or excessively long, as they might add noise to the analysis

4.3.10 Final Dataset Preparation

When the dataset was pictorially represented, it was discovered that more than 70% of the dataset contains less than 60 words. So, 60 was set as an upper limit for a review to be considered in the training data i.e.. all reviews with length greater than 60 words were rejected by the model. Then the dataset was split into TRAIN-VALIDATION-TEST in the ratio of 60:20:20.

Then the dataset was Tokenized i.e. words were converted to corresponding indices. The reviews with length less than 60 were padded with zeroes so as to have a consistent length of 60 for all reviews.

4.3.11 Data Storage

The cleaned and preprocessed tweets are stored in a new CSV file, ensuring they are ready for the next stages of the project, which involve feature extraction and model training.

4.4 What is Embedding ?

Embedding is the process of transforming tokens into dense, continuous, and fixed-size **vector representations**. The purpose of embedding is to convert the tokens into numerical representations that capture semantic meaning. Embedding maps discrete tokens into a continuous vector space where words with similar meanings are represented by vectors that are close to each other.

Example Workflow in NLP:

1. **Tokenization**:
   - Text: "I love NLP."
   - Tokenize: ["I", "love", "NLP"]
2. **Embedding**:
   - Embedding for each token
   - "I" → [0.1, 0.2, 0.3]
   - "love" → [0.4, 0.5, 0.6]
   - "NLP" → [0.7, 0.8, 0.9]

4.5 Methods for Embedding

Embedding can be done on mentioned ways

a) **Pre-trained Word Embeddings**:
Embeddings like **Word2Vec**, **GloVe**, or **FastText** provide a pre-trained mapping from words to vectors. These embeddings are learned on large corpora and capture semantic and syntactic relationships between words.

b) **Trainable Embeddings**:
In neural networks, embeddings can also be learned during training (e.g., **Embedding Layer** in Keras). information about the specific document's context.

This study bridges gaps by comparing traditional and domain-specific embeddings, providing an in-depth analysis of sentiment classification methods specifically designed for the domain specific (food e-commerce domain), which could lead to more accurate and meaningful results compared to generic pretrained models.
This value is high for a term that has high term frequency (in the given document) and low document frequency (across the corpus).

5. Explanation of Models and Methods Employed in Sentiment Analysis

Our study employs advanced natural language processing (NLP) techniques and machine learning models to predict consumer sentiment. We explore multiple approaches to text representation and sentiment classification:
Multiple approaches used:

Model 1: Custom Embedding
Model 2: Word2Vec Embedding
Model 3: GloVe Embedding + Averaging
Model 4: GloVe Embedding + Advanced Architecture

After preprocessing, tokenization and padding of data custom embedding layer is implemented in model architecture.

5.1 Custom Embedding

- **Overview and Implementation:** custom embedding layer is implemented as part of an LSTM architecture for sentiment analysis on the Amazon Fine Food Reviews dataset.

- **Model architecture:**
  Embedding layer: This is the custom embedding layer that learns word representations directly from the dataset
  LSTM layer(s): Process the sequential data, capturing long-term dependencies
  Dropout layer(s): For regularization to prevent overfitting
  Dense layer(s): For final classification, typically with a sigmoid activation for binary sentiment classification
  Training: The model is trained on the preprocessed review texts, learning to classify sentiment as positive or negative based on the review scores

5.2 Word2Vec embedding

- **Overview and implementation:**
  Model is trained on the preprocessed review texts using the gensim library. The trained Word2Vec embeddings are used to initialize a Keras Embedding layer . This Embedding layer is incorporated into an LSTM architecture for sentiment classification.

- **Model Architecture:**
  Embedding layer: Initialized with Word2Vec embeddings trained on the dataset
  LSTM layer(s): Processes sequential data
  Dropout layer(s): For regularization
  Dense layer(s): For final classification with sigmoid activation

5.3 GloVe Embedding + Averaging

- **Overview and implementation:**
  Leverages pre-trained word representations while simplifying the input by averaging word embeddings for each review. It can be effective for capturing overall sentiment without the need for complex sequential processing.
  - Load pre-trained GloVe embeddings
  - Create an embedding matrix using the GloVe embeddings for words in the dataset's vocabulary
    For each review:
  - Tokenize the text into individual words
  - Look up the GloVe embedding vector for each word
  - Average the embedding vectors of all words in the review
- **Model architecture:**
  Use the averaged embedding vectors as input features for a machine learning model, typically involving:
  Dense layer(s) for classification
  Sigmoid activation for binary sentiment classification (positive/negative)

5.3 GloVe Embedding + Advanced Architecture

- **Overview and implementation:**
- leverages pre-trained word representations directly in a more complex model, allowing the network to capture sequential dependencies in the text for sentiment classification.
  - Load pre-trained GloVe embeddings.
  - Create an embedding matrix using the GloVe embeddings for words in the dataset's vocabulary
  - Initialize a Keras Embedding layer with the pre-trained GloVe weights.
  - Incorporate the Embedding layer into an LSTM architecture:

- **Model Architecture:**
  Embedding layer: Initialized with pre-trained GloVe weights

LSTM layer(s): Processes sequential data
Dropout layer(s): For regularization
Dense layer(s): For final classification with sigmoid activation

6. Experiments & Observations

6.1 Model Evaluation Metrics

To assess the performance of embedding models the metrics used are

- **Accuracy:**
  Training Accuracy
  Validation Accuracy
  Validation Loss (Best Epoch)
- **Loss and Validation Accuracy Over Epochs**

6.2 Performance: Accuracy

6.2.1 Custom Embedding Model :

Training Accuracy - 99.14%
Validation Accuracy - 94.86%
Test Accuracy - 94.92%

6.2.2 Word2Vec Embedding Model

Training Accuracy - 93.21%
Validation Accuracy - 93.72%
Test Accuracy - 93.89%

6.2.3 GloVe Embedding + Averaging

Training Accuracy - 90.27%
Validation Accuracy - 89.42%
Test Accuracy - 89.38%

6.2.4 GloVe Embedding + Advanced Architecture

Training Accuracy - 92.92%
Validation Accuracy - 93.38%
Test Accuracy - 93.13%

6.3 Loss and Validation Accuracy Over Epochs

6.3.1 Custom Embedding + Word2Vec

Epoch	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy
1	0.0582	97.98%	0.1730	94.79%
2	0.0484	98.31%	0.2052	94.82%
3	0.0432	98.52%	0.1909	94.97%
4	0.0399	98.62%	0.2072	94.73%
5	0.0356	98.76%	0.2341	94.85%
10	0.0250	99.14%	0.2368	94.86%

6.3.2 GloVe Embedding Model

Epoch	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy
1	0.2729	88.81%	0.2068	91.86%
2	0.2259	90.96%	0.1970	92.36%
3	0.2107	91.60%	0.1880	92.73%
4	0.2018	92.00%	0.1825	92.83%
5	0.1955	92.28%	0.1818	92.88%
10	0.1802	92.92%	0.1713	93.38%

7. Observations:

1. **GloVe Embedding Model**:
   - Achieves good validation accuracy (93.38% by epoch 10).
   - The loss steadily decreases, showing consistent training.
2. **Custom Embedding + Word2Vec Model**:
   - High training accuracy (99.14%) but some overfitting observed as validation accuracy stabilizes around 94.92%.
   - Slight fluctuation in validation loss across epochs.

8. Discussion

8.1 Strengths of Each Embedding Approach:

a) *GloVe Embeddings:*

- Pre-trained on a large corpus, making it effective for general language understanding.
- Captures global statistical co-occurrence relationships between words.
- Performs well on tasks with sufficient training data and less domain specificity.

Strength: Quick to implement, requires less computational power.

b) Custom Embeddings (Word2Vec):

- Tailored to the specific dataset, enabling better performance for

domain-specific tasks like sentiment analysis on fine food reviews.
- Captures context and semantics of words unique to the dataset.

Strength: Superior performance on niche datasets due to focused vocabulary learning.

8.2 Weaknesses of Each Embedding Approach:

a) ***GloVe Embeddings:***
- Limited adaptability to domain-specific terms or phrases not covered in the pre-trained corpus.
- Does not account for context-dependent word meanings (e.g., polysemy).

b) ***Custom Embeddings (Word2Vec):***
- Requires significant computational resources and time for training on large datasets.
- Performance is highly dependent on the quality and quantity of training data.
- Can overfit on smaller datasets, reducing generalization ability.

9. Conclusions & Roadmap

9.1 Conclusions

- This study demonstrates the efficacy of different embedding techniques—GloVe and custom Word2Vec embeddings—for sentiment analysis on the Amazon Fine Food Reviews dataset. Through comprehensive preprocessing, feature extraction, and model training, the findings highlight key trade-offs between pre-trained and custom embeddings.
- The results indicate that while GloVe embeddings are effective for general sentiment analysis tasks, custom embeddings excel in capturing domain-specific semantics, yielding higher accuracy on niche datasets. However, the choice of embeddings and models depends on resource availability, dataset size, and the specific application. This research contributes to the understanding of embedding selection for sentiment analysis and provides practical frameworks for real-world applications in e-commerce, customer feedback analysis, and public sentiment tracking.

9.2 Roadmap for Future Work

- Future research can focus on incorporating **contextual embeddings** like BERT and RoBERTa to capture nuanced word meanings, enhancing the accuracy of sentiment analysis. Addressing **dataset biases** by balancing sentiment classes and expanding to multilingual analysis using cross-lingual embeddings (e.g., mBERT) can make sentiment analysis systems more robust and globally applicable.Another direction involves developing **real-time sentiment analysis systems** for processing live data streams, such as social media posts. Optimized pipelines and interpretability tools like SHAP can ensure faster inference and greater transparency. Additionally, exploring **fine-grained sentiment analysis** for sentiment intensity and leveraging transfer learning for domain-specific applications will further expand the applicability of sentiment analysis in fields like healthcare, finance, and public policy.

References

1. A.Pak and P. Paroubek. „Twitter as a Corpus for Sentiment Analysis and Opinion Mining". In Proceedings of the Seventh Conference on International Language Resources and Evaluation, 2010, pp.1320-1326

2. J. McAuley, "Amazon review data", *UCSD*.
3. Po-Wei Liang, Bi-Ru Dai, “Opinion Mining on Social MediaData", IEEE 14th International Conference on Mobile Data Management,Milan, Italy, June 3 - 6, 2013, pp 91-96, ISBN: 978-1-494673-6068-5, http://doi.ieeecomputersociety.org/10.1109/MDM.2013.
4. T. U. Hague, N. N. Saber and F. M. Shah, "Sentiment analysis on large scale Amazon product reviews", *2018 IEEE International Conference on Innovative Research and Development (ICIRD)*, 2018.
5. Verma, S. Sentiment analysis of public services for smart society: Literature review and future research directions. *Gov. Inf. Q.* **2022**, *39*, 101708. [**Google Scholar**] [**CrossRef**]
6. A. Almjawel, S. Bayoumi, D. Alshehri, S. Alzahrani and M. Alotaibi, "Sentiment Analysis and Visualization of Amazon Books’ Reviews", *2019 2nd International Conference on Computer Applications & Information Security (ICCAIS)*, 2019.
7. J. Kamps, M. Marx, R. J. Mokken, and M. De Rijke, “Using wordnet to measure semantic orientations of adjectives,” 2004.
8. R. Xia, C. Zong, and S. Li, “Ensemble of feature sets and classification algorithms for sentiment classification,” Information Sciences: an International Journal, vol. 181, no. 6, pp. 1138–1152, 2011.
9. Li Rao,” Sentiment Analysis of English Text with Multilevel Features”, January 2022
10. Bollegala, D., Weir, D., & Carroll, J.. Cross-Domain SentimentClassification using a Sentiment Sensitive Thesaurus. Knowledge andData Engineering, IEEE Transactions on, 25(8), 1719-1731,2013
11. Al-Otaibi, S.; Alnassar, A.; Alshahrani, A.; Al-Mubarak, A.; Albugami, S.; Almutiri, N.; Albugami, A. Customer satisfaction measurement using sentiment analysis. *Int. J. Adv. Comput. Sci. Appl.* **2018**, *9*. [**Google Scholar**] [**CrossRef**]
12. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *NAACL-HLT*.
13. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv:1301.3781*.
14. Pang, B., & Lee, L. (2008). Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval*.
15. Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global Vectors for Word Representation. *EMNLP*.
16. ang, D., Qin, B., & Liu, T. (2015). Document Modeling with Gated Recurrent Neural Network for Sentiment Classification. *EMNLP*.
17. Zhou, P., Shi, W., Tian, J., et al. (2018). Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification. *ACL*.
18. Ali, H.; Hashmi, E.; Yayilgan Yildirim, S.; Shaikh, S. Analyzing amazon products sentiment: A comparative study of machine and deep learning, and transformer-based techniques. *Electronics* **2024**, *13*, 1305. [**Google Scholar**] [**CrossRef**]