International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Examining the Issues Associated with Linguistic Variety and Sentiment Expression in Several Languages as well as the Application of Deep Learning Techniques to Cross-Lingual Sentiment Analysis

Kesarla Venkata Sai Mounish ,Sindhe Sai Sunith , Syed Raiyan Ali , Sai Charan S

Madanapalle Institute of Technology & Science
SRM Easwari Engineering College
Email: kvsmounish@gmail.com, sunithsindhe@gmail.com, syedraiyanalioff@gmail.com, saicharan1173@gmail.com

ABSTRACT:

As communication transcends linguistic boundaries in our globalized world, understanding sentiment expressed in diverse languages becomes crucial for effective information processing and decision-making. This study investigates the challenges associated with linguistic variety and sentiment expression across several languages, and explores the application of deep learning techniques to facilitate cross-lingual sentiment analysis.

The first part of the study delves into the intricate nuances of linguistic diversity, analyzing how sentiments are articulated in different languages, considering cultural and contextual variations. This exploration aims to identify the unique challenges posed by linguistic variety, including idiomatic expressions, cultural references, and language-specific sentiment cues.

The second phase of the research focuses on the application of state-of-the-art deep learning techniques for cross-lingual sentiment analysis. Leveraging neural network architectures and natural language processing methodologies, we aim to develop a robust model capable of understanding and classifying sentiment across multiple languages. The study evaluates the effectiveness of the model in handling the inherent complexities of diverse linguistic structures and expressions.

Through empirical analysis and experimentation, this research contributes insights into the adaptability and generalizability of deep learning models for sentiment analysis in a cross-lingual context. The findings not only enhance our understanding of sentiment expression in different languages but also provide practical implications for the development of more accurate and versatile cross-lingual sentiment analysis tools.

In conclusion, this study bridges the gap between linguistic diversity and sentiment analysis by addressing the challenges posed by various languages and proposing innovative solutions through the application of deep learning techniques. The outcomes of this research have implications for improving the efficiency of sentiment analysis in multilingual settings, thereby advancing our ability to comprehend and respond to sentiments expressed in a globalized and linguistically diverse digital landscape.

Keywords: Sentiment Analysis, Machine Learning, Deep Learning, Multi-Lingual, Cross-Lingual, Natural Language Processing, Web Data Mining, Text Mining

1. Introduction:

In today's interconnected global society, the rapid growth of digital communication platforms has facilitated the exchange of information and opinions across linguistic and cultural boundaries. As a result, understanding sentiment expressed in various languages has become increasingly crucial for businesses, policymakers, and researchers alike. However, linguistic diversity poses a significant challenge to sentiment analysis, as expressions of sentiment may vary widely across different languages and cultural contexts.

The complexity of linguistic diversity is compounded by the fact that sentiment analysis models trained in one language may not perform well when applied to another due to variations in syntax, semantics, and cultural nuances. Additionally, the scarcity of labeled datasets in certain languages further exacerbates the challenges associated with accurate sentiment analysis on a global scale.

1.1 Objectives:

The primary objectives of this study are:

1. **Explore the Issues of Linguistic Variety:** Investigate the challenges posed by linguistic variety in sentiment analysis, including differences in linguistic structures, idiomatic expressions, and cultural nuances across multiple languages.
2. **Examine Sentiment Expression Across Languages:** Analyze how sentiment is expressed in diverse languages, considering the impact of linguistic and cultural factors on the interpretation of positive, negative, and neutral sentiments.
3. **Investigate Cross-Lingual Sentiment Analysis Techniques:** Explore existing methodologies and techniques for cross-lingual sentiment analysis, with a focus on the application of deep learning models. Evaluate the effectiveness of these techniques in overcoming language barriers and improving sentiment analysis accuracy across diverse languages.
4. **Assess the Robustness of Deep Learning Models:** Evaluate the performance and robustness of deep learning models, such as recurrent neural networks (RNNs) and transformer-based models, in handling cross-lingual sentiment analysis tasks. Examine how well these models generalize across different languages and identify potential areas for improvement.
5. **Propose Recommendations for Practical Applications:** Based on the findings, provide recommendations for the practical application of cross-lingual sentiment analysis in real-world scenarios. Address potential challenges and suggest strategies for enhancing the accuracy and reliability of sentiment analysis across linguistic and cultural boundaries.

2. Preliminary Concepts of Sentiment Analysis

Sentiment Analysis involves the exploration of a significant volume of unstructured data to discern opinions and sentiments expressed toward a specific topic or object. It can be viewed as a sub-field within text processing.

Any text subject to analysis can fall into one of two categories: factual, expressing facts and information, or opinionated, conveying views and opinions. Identifying such opinionated statements is crucial for tasks related to opinion mining. This identification process is known as Subjectivity Classification, a prerequisite for Sentiment Analysis. A subjective statement expresses emotion or feeling, which may or may not be an opinion, although the majority of opinionated statements are generally subjective. Even objective statements may carry opinions at times.

Once an opinionated statement is identified, Sentiment Analysis is conducted to classify the opinion as positive or negative. A more in-depth analysis is required to determine the specific type of opinion expressed in the statement. Therefore, subjectivity classification and sentiment classification are complementary processes.

2.1 Tasks within Sentiment Analysis

Comprehending and identifying opinions expressed in texts is a highly intricate task with multiple facets. The typical structure of a sentiment analysis task outlines five essential components for a document: Opinion, Features, Object, Opinion Holder, and Time of Expression.

1) The first step involves determining the target of the opinion, i.e., the object on which the opinion is expressed. Opinions can encompass the entire object or focus on a specific component, commonly referred to as a feature or aspect. A document may include viewpoints toward multiple features of the same object, the object as a whole, and even multiple objects. Some documents may involve a comparative analysis of objects based on shared features.
2) Subsequently, the type of opinion must be identified, a process known as polarity identification. This classification task broadly categorizes opinions as positive or negative. A more detailed classification can identify specific emotions associated with the opinion, such as joy, anger, sadness, surprise, etc.
3) Certain situations necessitate determining the opinion holder, i.e., the person expressing the opinion. While it is typically assumed that a document contains the opinion of a single individual throughout, multiple people may express their opinions, which may or may not align.
4) Additionally, determining the time at which the opinion was expressed becomes essential in specific cases.
5) The identification of features significantly influences the accuracy of polarity detection. Features are critical factors when seeking sentiments, with some texts explicitly highlighting them. For example, a statement like "The picture clarity of this TV set is excellent" explicitly mentions the view about the "clarity" feature of the object "TV set." These features are termed explicit features. Implicit features, on the other hand, may not be mentioned directly but are inferred from the context, making their analysis more challenging.

The crux of Sentiment Analysis lies in finding the polarity of an expressed opinion. For this purpose, words are identified as "positive" or "negative." Positive terms include "good," "excellent," "satisfactory," etc., while negative terms encompass "poor," "disgusting," "bad," etc. Detecting the presence of such terms provides insights into the polarity of a statement. However, this task is not always straightforward, as opinions can be expressed in various forms. Statements may present a clear, direct opinion or express an effect of an object on another without using explicit descriptive words. Comparative opinions may also be conveyed through the analysis of multiple objects based on certain common features, indicating a preference based on the comparison.

2.2 Text Sources for Sentiment Analysis

The text suitable for sentiment analysis can be sourced from a diverse array of platforms. Some of the most prevalent sources include:

1) **Opinion Blogs –** Blogging has become a widespread activity covering a variety of topics such as gadgets, current affairs, political issues, travel spots, etc. Analyzing blogs provides insights into opinions on respective content.
2) **E-commerce Reviews –** Users on popular e-commerce sites like Amazon, Flipkart, eBay, etc., often provide feedback and reviews, usually accompanied by star ratings. These opinions are valuable for prospective buyers seeking information before making a purchase.
3) **Social Media Content and Media Sharing –** With approximately 2.5 billion active users on social media platforms worldwide, data generated on platforms like Facebook, Twitter, LinkedIn, Instagram, YouTube, SoundCloud, etc., is abundant and serves as a prime source for information regarding users' opinions and preferences.
4) **Communication –** Platforms like SMS and WhatsApp also serve as rich sources of information that can be mined for insights and opinions.

3. Techniques for Sentiment Analysis

Sentiment Analysis employs various techniques to analyze and interpret the sentiment expressed in textual data. Some common techniques include:

1. Machine Learning Algorithms: Supervised Learning: Using labeled datasets, machine learning models like Support Vector Machines (SVM), Naive Bayes, and Decision Trees are trained to classify text into positive, negative, or neutral sentiments. Deep Learning: Neural networks, especially recurrent neural networks (RNNs) and transformers like BERT and GPT, have shown effectiveness in capturing complex contextual information for sentiment analysis.

2. Hybrid Approaches: Integrating multiple techniques, such as combining lexicon-based methods with machine learning models, to leverage the strengths of different approaches and enhance overall sentiment analysis accuracy.

3. Transfer Learning: Pre-training models on large datasets for general language understanding and then fine-tuning them on sentiment-specific tasks. This approach leverages knowledge gained from diverse contexts to improve sentiment analysis performance.

4. Word Embeddings: Representing words as vectors in a continuous vector space, where semantically similar words are closer. Techniques like Word2Vec and GloVe help capture semantic relationships and improve sentiment analysis accuracy.

5. Emotion Analysis: Extending sentiment analysis to include emotion detection. This involves identifying and categorizing emotions expressed in text, providing a more comprehensive understanding of the emotional tone.

3.1 Evaluation Metrics for Performance Measurement

The assessment of sentiment analysis techniques relies on four widely used metrics: Precision, Recall, Accuracy, and F-measure.

$$Precision = \frac{RP}{RP+WP} \tag{1}$$

$$Recall = \frac{RP}{RP+WN} \tag{2}$$

$$Accuracy = \frac{RP+RN}{RP+RN+WP+WN} \tag{3}$$

$$F - Measure = \frac{(2*Precision*Recall)}{(Precision+Recall)} \tag{4}$$

Here, RP represents Right Positives, WP denotes Wrong Positives, RN signifies Right Negatives, and WN stands for Wrong Negatives. The terms are illustrated in a confusion matrix, as depicted in Table 1.

4. Multi-Lingual/ Cross-Lingual Sentiment Analysis

This paper surveys and explores various techniques and methodologies employed in Multi-Lingual and Cross-Lingual Sentiment Analysis.

4.1 Multi-lingual and Cross-lingual Sentiment Analysis using Machine Learning

This paper investigates the application of Machine Learning Techniques in Multi-lingual and Cross-lingual Sentiment Analysis, focusing on the development and evaluation of models capable of effectively classifying sentiments in texts written in different languages. We explore supervised learning approaches, leveraging algorithms such as Support Vector Machines, Naive Bayes, and Neural Networks, and unsupervised methods, including clustering and topic modeling. The study emphasizes feature representation, including word embeddings and deep learning-based contextual embeddings, to capture nuanced linguistic features for sentiment classification. Transfer learning techniques, utilizing pre-trained models on large multilingual corpora, are also discussed to enhance model performance across languages. Evaluation metrics tailored for cross-lingual sentiment analysis are employed to assess the effectiveness of the proposed machine learning models.

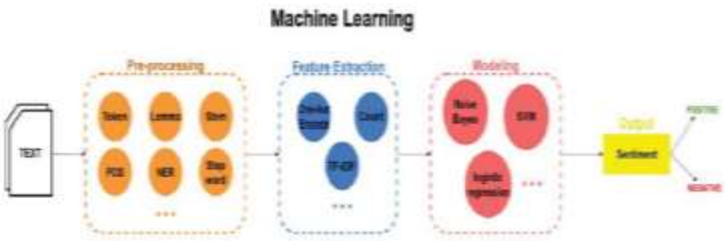

Figure 1: Sentiment Analysis using Machine Learning

5. Deep Learning for sentiment analysis

Deep Learning for sentiment analysis involves the utilization of advanced neural network architectures to comprehend and classify sentiments in textual data. Here's an overview of the key aspects:

1. Neural Network Architectures:

1. Recurrent Neural Networks (RNNs): These are capable of capturing sequential dependencies in text data, making them suitable for sentiment analysis tasks where context matters.
2. Long Short-Term Memory (LSTM): A type of RNN designed to address the vanishing gradient problem, making it effective in learning long-term dependencies.
3. Gated Recurrent Unit (GRU): Another variant of RNN that simplifies the architecture compared to LSTM while maintaining competitive performance.

2. Word Embeddings: Word2Vec, GloVe, and FastText: Pre-trained word embeddings capture semantic relationships between words, allowing models to understand contextual nuances and improve sentiment analysis accuracy.

3. Convolutional Neural Networks (CNNs): Applying CNNs to sentiment analysis involves using convolutional layers to extract relevant features from textual data. This is particularly effective in capturing local patterns and expressions.

4. Attention Mechanisms: Transformer Models: Attention mechanisms, as seen in transformer models like BERT, GPT, and RoBERTa, have demonstrated state-of-the-art performance in various NLP tasks, including sentiment analysis. These models capture global context and dependencies effectively.

5. Deep Transfer Learning: Knowledge Transfer: Applying knowledge learned from sentiment analysis tasks in one language to improve performance in another, especially when labeled data in the target language is limited.

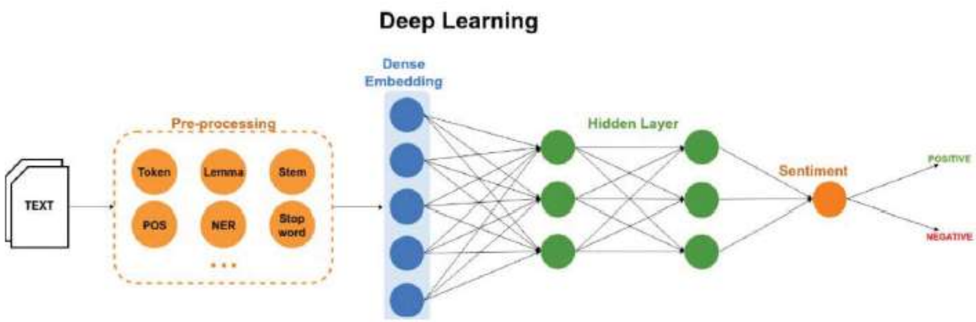

Figure 2: Sentiment Analysis using Deep Learning

6. Related Work

This study aims to explore diverse techniques and methodologies in sentiment analysis, providing a foundation for prospective empirical research. Notably, recent advancements in sentiment analysis have been driven by the integration of deep learning models such as Deep Neural Networks (DNN), Convolutional Neural Networks (CNN), and Recurrent Neural Networks (RNN), contributing to enhanced efficiency in sentiment analysis tasks. In this section, state-of-the-art techniques in sentiment analysis based on deep learning are scrutinized.

Since 2015, a notable trend has emerged in scholarly exploration. Tang et al. introduced deep learning-based algorithms encompassing aspects like learning word embeddings, sentiment categorization, and opinion extraction. Zhang and Zheng delved into the application of machine learning for sentiment analysis, employing Part-of-Speech (POS) as a textual feature and TF-IDF for word weight computation. Sharef et al. Investigated the advantages of large-scale data methodologies in sentiment analysis. Recent studies are referenced, where deep learning techniques, specifically Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM), are reviewed and compared within the realm of sentiment analysis problems.

Beyond these foundational studies, other research has extended the application of sentiment analysis based on deep learning to various domains. These include banking, weather-related tweets, travel advisories, recommender systems for cloud services, and movie reviews. For instance, in, Word2Vec was employed to automatically extract text characteristics from diverse data sources, translating user information and weather knowledge into word embeddings. Papers such as adopt similar methodologies, combining topic modeling with sentiment analysis on customer-generated social media data to unveil product development prospects.

Jeong et al. showcased sentiment analysis as a tool for real-time monitoring and analysis of evolving customer demands in rapidly developing product scenarios. Pham et al. analyzed travel evaluations, discerning opinions on criteria like value, room, location, and cleanliness. Further applications include the application of polarity-based sentiment deep learning to tweets. These authors detailed their utilization of deep learning models to enhance the accuracy of sentiment assessments, predominantly focusing on content posted in English, with a handful addressing tweets in other languages such as Spanish, Thai, and Persian. Previous research has explored tweets using various models of polarity-based sentiment deep learning, including DNN, CNN, and hybrid techniques.

6.1 Data Collection

For conducting sentiment analysis, researchers can either create custom datasets or utilize existing databases. Generating a new dataset enables the use of data specifically relevant to the analyzed issue, ensuring compliance with privacy regulations. This thesis aimed to amass data and public opinions regarding a furniture shop from various social media platforms, employing web scraping to construct datasets from Twitter, Reddit, and several consumer forum websites containing reviews of the furniture store's products.

	review
0	"My partner and I visited Lakeside IKEA and wa...
1	"What a horrible experience after collecting I...
2	"Same as many others have found, disgraceful c...
3	"HONESTLY I WISH I COULD WRITE -5 STARS BECAUS...
4	"Well! I have approached Ikea a few times and ...

Figure 3: This is the sample Dataset after the web scraping one of the social media channel

6.2 Datasets

Diverse datasets were compiled from various platforms, covering different aspects of the topic to facilitate a comprehensive range of experiments. The datasets include:

1) **Twitter Dataset:** The primary dataset, comprising nearly 1.2 million tweets discussing various opinions about the furniture store. It contains fields such as 'user id,' 'date,' 'tweet URL,' and 'text' containing the primary review.

Table 1: Twitter Dataset

Metrics	TF-IDF			word2vec		
	CNN	DNN	RNN	CNN	DNN	RNN
Accuracy	0.7563	0.7548	0.5432	0.8001	0.7702	0.815
Recall	0.7321	0.7423	0.7623	0.8012	0.7865	0.8241
Precision	0.7366	0.748	0.7635	0.8023	0.7845	0.8269
F Score	0.7542	0.764	0.6412	0.8074	0.7888	0.818
AUC	0.754	0.746	0.7557	0.8006	0.7875	0.8214

2) **Reddit Dataset:** Gathered from the social networking site Reddit using a search string related to the furniture store, this dataset comprises approximately 2200 samples.

Table 2: Reddit Dataset

Metrics	TF-IDF			word2vec		
	CNN	DNN	RNN	CNN	DNN	RNN
Accuracy	0.7124	0.7542	0.5062	0.7541	0.7325	0.7247
Recall	0.7244	0.7321	0.6231	0.8102	0.7294	0.7384
Precision	0.7144	0.7655	0.5541	0.7321	0.7325	0.7221
F Score	0.7144	0.7452	0.5210	0.7622	0.7322	0.7215
AUC	0.7211	0.7451	0.501	0.7514	0.7358	0.7214

3) **Sitejabber Dataset:** Consisting of customer comments about the product, this dataset includes 9890 samples with the same fields as the Twitter Dataset.

Table 3: SiteJabber Dataset

Metrics	TF-IDF			word2vec		
	CNN	DNN	RNN	CNN	DNN	RNN
Accuracy	0.6651	0.6924	0.5421	0.7142	‘0.7012	0.7567
Recall	0.6678	0.6932	0.8421	0.7241	0.7004	0.8014
Precision	0.6623	0.7012	0.4327	0.7123	0.7023	0.7423
F Score	0.678	0.6978	0.5874	0.7145	0.7088	0.7784
AUC	0.6642	0.6933	0.5023	0.7414	0.7023	0.762

4) **Reviewsio Dataset:** A collection of reviews about the store, containing about 23100 samples.

Table 4: Review IO Dataset

Metrics	TF-IDF			word2vec		
	CNN	DNN	RNN	CNN	DNN	RNN
Accuracy	0.8122	0.8412	0.5594	0.8547	0.835	0.864
Recall	0.7954	0.8321	0.4512	0.8324	0.8365	0.8547
Precision	0.8255	0.8411	0.6021	0.8542	0.8369	0.8632
F Score	0.8011	0.8423	0.4523	0.8471	0.8214	0.874
AUC	08114	0.8544	0.5513	0.8541	0.8331	0.8641

5) **Consumer Affairs Dataset:** With the most extensive history of reviews, this dataset includes around 51000 samples.

Table 5: Consumer Affairs Dataset

Metrics	TF-IDF			word2vec		
	CNN	DNN	RNN	CNN	DNN	RNN
Accuracy	0.7921	0.8423	0.5741	0.8142	0.8014	0.8563
Recall	0.7412	0.8654	0.5541	0.8214	0.7714	0.8741
Precision	0.8241	0.8365	0.5847	0.8102	0.8001	0.8475
F Score	0.7845	0.8425	0.5632	0.8102	0.7821	0.8541
AUC	0.795	0.8475	0.5741	0.8147	0.7956	0.8547

6.3 Experiment Setup

1) "Review Author Name Link" represents the user's handle name.

2) "Review Title Link Href" is the unique link to that specific comment.

3) "Review Title" denotes the title of the review.

4) "Review Date" provides the date of the review.

5) "Review Text" contains the actual text of the review.

review_author_name_link	review_title_link href	ReviewTitle	review_date	review_text
LincolnÂ A.	https://www.sitejabber.com	Good quality	March 7th, 2021	My purchase was a reading desk. They
RichÂ B.	https://www.sitejabber.com	I like its minimalism in	August 2nd, 2020	I bought some furniture for the childre
ClaireÂ B.	https://www.sitejabber.com	Don't give Ikea your cc	May 11th, 2021	Not only have Ikea failed to deliver m,
SophiaÂ M.	https://www.sitejabber.com	This brand is very pop	November 28th, 2020	This brand is very popular now. I really
AJÂ W.	https://www.sitejabber.com	You get what you pay	June 9th, 2020	Some good, some bad. Honestly it's a
NancyÂ N.	https://www.sitejabber.com	Horrible service and pr	October 14th, 2020	We ordered a shelf from IKEA in vaugh
KimÂ B.	https://www.sitejabber.com	IKEA is good, but the c	August 26th, 2021	This review is more about the compan
AlberteÂ O.	https://www.sitejabber.com	Excellent	June 1st, 2020	Nothing to say except the words of gr
LaurieÂ B.	https://www.sitejabber.com	Beautiful displays but t	April 18th, 2019	I visited the Oak Creek store in Wiscon
maeÂ q.	https://www.sitejabber.com	Ikea delivery	November 2nd, 2021	My first time using delivery service fro
GioiaÂ L.	https://www.sitejabber.com	Amazing shop	September 14th, 2021	This is am amazing shop. There are so
AnnaÂ D.	https://www.sitejabber.com	Doesn't get any worse	July 15th, 2020	I've been trying to get in touch with

Figure 4: Sample of Sitejabber Dataset

6.4 Data Preparation

To align with the scope of our experiment, the dataset needs to be labeled for classification. As mentioned in the thesis, the BERT transformer was utilized for labeling the data into positive and negative sentiments based on the generated polarity for each tweet.

1) Download and extract the dataset, then explore the directory structure.

2) Choose the BERT model to load from TensorFlow Hub for fine-tuning. Various BERT models are available, and for this experiment, the `bert-base-multilingual-uncased-sentiment` model was employed.

Using this BERT model, the reviews were classified into a scale of 1-5, as depicted in Figure 4.4, which was further translated into positive and negative sentiments.

```
In [5]: tokenizer = AutoTokenizer.from_pretrained('nlptown/bert-base-multilingual-uncased-sentiment')
        model = AutoModelForSequenceClassification.from_pretrained('nlptown/bert-base-multilingual-uncased-sentiment')
```

Figure 5: BERT Model

	review	sentiment
0	"My partner and I visited Lakeside IKEA and wa...	2
1	"What a horrible experience after collecting I...	1
2	"Same as many others have found, disgraceful c...	1
3	"HONESTLY I WISH I COULD WRITE -5 STARS BECAUS...	1
4	"Well I have approached Ikea a few times and ...	1
5	"I don't know where to begin from! Worse worse...	1
6	"I would gone 0 stars, but I have to give 1 st...	1
7	"Wow, where to start with my utter disdain and...	1
8	"I have been waiting since before Christmas fo...	2
9	"I'm an old Ikea customer going back to 1994, ...	1
10	"HORRIBLE EXPERIENCE! Do not order your kitche...	1
11	"Ikea Exeter, please see today's offer for Pax...	5
12	"Delivery costs are ridiculous, stuff you woul...	1
13	"I'm soooooooooo extremely disappointed with I...	1
14	"I am writing here to comment on your company'...	1
15	"So they let you place an order - then turn ar...	1

Figure 6: A sample of the labeled dataset generated

6.5 Training, Validation, and Test Sets

The datasets were randomly split into three subsets: training, validation, and testing, maintaining an equal percentage of positive and negative comments (70:10:20). This stratified split ensures balanced representation of each class using the `train_test_split()` function from scikit-learn. The training subset is used to train the model, the validation subset helps determine when to stop training early, and the testing subset assesses the model's performance.

Table 6: for the sizes of each subset.

Subset	Number of Tweets
Training Set	861000
Validation Set	123000
Testing Set Set	246000

6.6 Data Cleaning

Text cleaning, a crucial preprocessing step, involves removing irrelevant words or components to enhance sentiment analysis effectiveness. Common components include white space, punctuation, and stop words. Text cleaning steps include:

1) Tokenization: Separating the phrase into words.
2) Lowercasing: Converting words to lowercase to ensure consistency.
3) Stop Words: Eliminating widely used but non-discriminatory terms.
4) Stemming: Transforming words into their basic form.
5) Lemmatization: Reducing words to existing terms in the language.

6.7 Word Embedding

Machine learning algorithms require numerical inputs, and word embeddings facilitate this process by translating words into vectors. Word2Vec, a popular embedding technique, is employed alongside TF-IDF. Unsupervised learning is conducted using artificial neural networks to train the Word2Vec model, creating word vectors that indicate semantic proximity.

6.8 TF-IDF (Term Frequency-Inverse Document Frequency)

TF-IDF measures the mathematical importance of document words. It involves vectorization, similar to One-Hot Encoding, where the word's value is TF-IDF instead of 1. The TF-IDF value is computed by multiplying term frequency (TF) and inverse document frequency (IDF). The scikit-learn package's vectorizer class is utilized for TF-IDF.

The formula for computing TF-IDF for a term (t) of a document (d) in a document set is:

$$\text{tf-idf}(t, d) = \text{tf}(t, d) \times \text{idf}(t)$$

where $\text{idf}(t) = \log \left(\frac{n}{\text{df}(t)} \right) + 1$, n is the total number of documents, and $\text{df}(t)$ is the document frequency of term t.

7. Implementation

For each dataset, a tailored processing approach was employed to facilitate model construction. For example, in the Twitter dataset, irrelevant columns such as "id," "date," "query," "username" were removed, and the class label was transformed into positive and negative values.

After cleansing the datasets, sentences were tokenized into individual words and lemmatized to return them to their base form. Two approaches, word embedding and TF-IDF, were used to convert feature vectors into word vectors. Both types of feature vectors served as inputs to the deep learning algorithms evaluated in this research, including Convolutional Neural Networks (CNNs), Deep Neural Networks (DNNs), and Recurrent Neural Networks (RNNs). Separate models were constructed for each type of vector.

Many traditional models rely on well-known features like bag-of-words, n-grams, and TF-IDF, overlooking semantic similarity between words. In contrast, deep learning models in natural language processing require word embedding findings as input features. Word2Vec was used to train initial vectors based on the provided datasets.

As mentioned earlier, k-fold cross-validation with k equal to ten was employed to assess the effectiveness of various embeddings. The neural network architecture includes an embedding layer that understands the embedding for all terms in the training datasets. In this instance, the vocabulary size is 17,000, and the maximum length is 40 characters, resulting in a 40 by 300 matrix.

The initial 1D CNN layer consists of filters with a kernel size of 3. Sixty-four filters are defined, allowing 64 distinct features to be trained on the initial layer. The output is a 40 by 64 neuron matrix, which is fed into the next layer.

Another 32 distinct filters are defined for training on the subsequent layer. Similar to the primary layer, the resulting matrix has dimensions of 40 by 32. A max-pooling layer is employed after a CNN layer to reduce output complexity and prevent data overfitting. In this instance, a level of three is chosen, resulting in a matrix size of 13 by 32.

The third and fourth 1D Convolutional Neural Network layers produce a 13 × 16 matrix and a 13 × 8 matrix. An average pooling layer is then utilized to prevent overfitting, using the average value instead of the highest number. The size of the output matrix is 1 by 8 neurons. The final layer is a fully connected layer with sigmoid activation, reducing the 8-dimensional vector to 1 for prediction ("positive" or "negative").

8. Results

Experiments were conducted with DNN, CNN, and RNN models using TF-IDF feature extraction and word embedding to assess their effectiveness. The following section presents the results of these experiments. The code's parameters were consistently set for each experiment, and performance metrics such as Area under Curve, F-score, and Accuracy were employed.

The initial dataset analyzed was the Twitter dataset, wherein content was categorized as either positive or negative. Given the larger volume of tweets in this dataset, subsets with varying proportions of the initial data were used to explore model performance. The images below depict performance metrics based on the percentage of tweets processed.

Figure 7: Accuracy values of the models with TF-IDF and Word Embedding](image-link)

The first performance metric, Accuracy, is illustrated in Figure 7. Both TF-IDF and Word Embedding exhibit similar patterns across varying data proportions, with CNN showing an exponential increase in accuracy with TF-IDF in the last 10%. DNN achieved a score between 0.75 and 0.8 for both TF-IDF and Word Embedding. RNN with TF-IDF performed the least, with an average accuracy of around 0.55, while Word Embedding with RNN surpassed others with an accuracy of over 0.8.

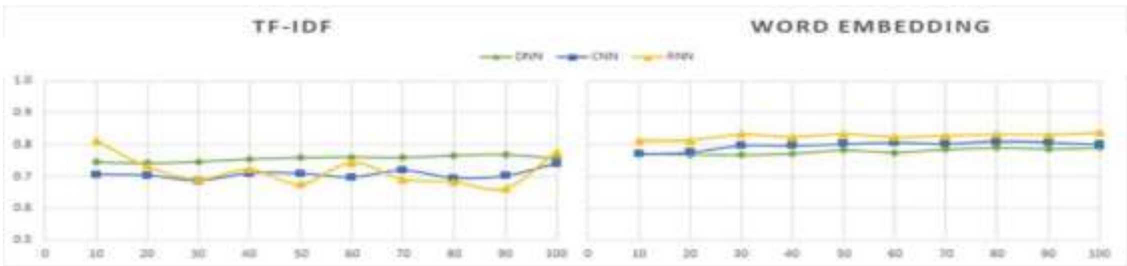

Figure 8: Recall values of the models with TF-IDF and Word Embedding](image-link)

Both TF-IDF and Word Embedding resulted in DNN having a Recall Value between 0.75 and 0.8. CNN performed slightly better with Word Embedding, having a Recall Value closer to 0.8 compared to TF-IDF's 0.7. RNN with Word Embedding was the best performer with a value above 0.8.

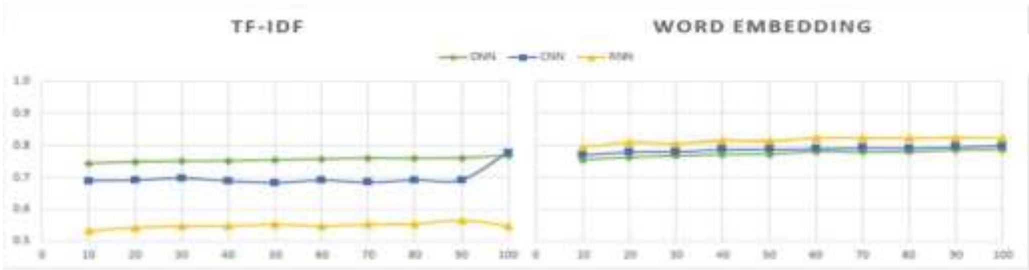

Figure 9: Precision values of the models with TF-IDF and Word Embedding](image-link)

Figure 9 shows consistent precision performance around 0.8 for Word Embedding with CNN, RNN, and DNN. TF-IDF matched this value with DNN but significantly underperformed with RNN, scoring only 0.5. CNN demonstrated decent performance, with precision ranging from 0.7 to 0.8.

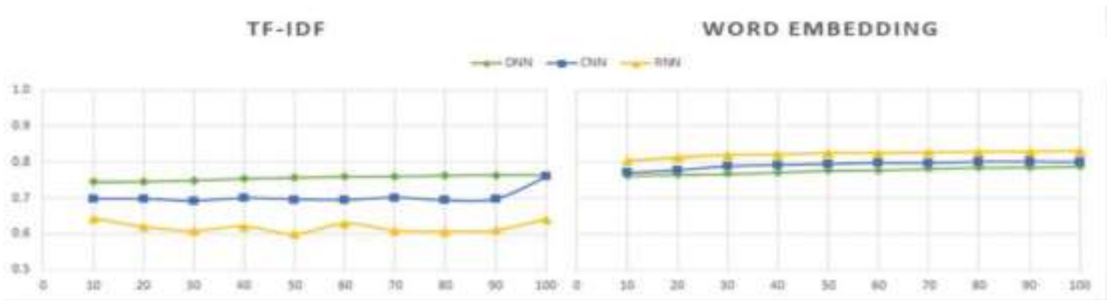

Figure 10: F-Score values of the models with TF-IDF and Word Embedding](image-link)

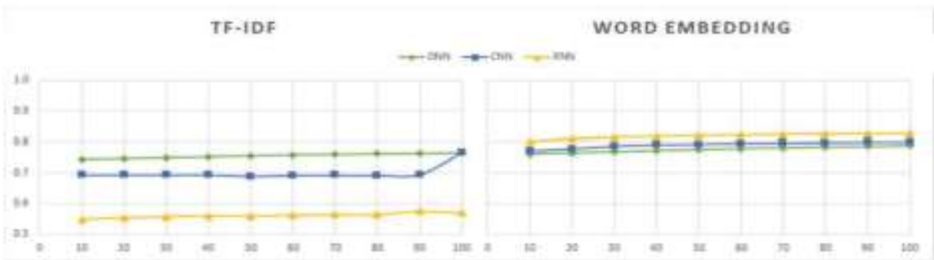

Figure 10 : AUC values of the models with TF-IDF and Word Embedding](image-link)

Figures 10 display similar patterns for F-Score and AUC. Word Embedding consistently achieved an average value of around 0.8, with RNN outperforming CNN and DNN. TF-IDF scored highest with DNN at 0.75, followed closely by CNN with 0.7. RNN displayed lower performance, with an AUC value of 0.55 and an F-Score of 0.6.

9. Conclusion

This research has systematically reviewed the existing literature on multilingual sentiment analysis, identifying prominent languages with dedicated corpora and detailing the techniques utilized along with their respective contributions and accuracy rates. Various mechanisms, including popular approaches like machine translation, have been observed in addressing sentiment analysis challenges. The ongoing efforts to create comprehensive corpora in numerous languages have made noteworthy advancements.

Despite the significant strides made in certain languages, the realm of multilingual sentiment analysis remains relatively unexplored, presenting ample opportunities for future investigations. Several languages have yet to be thoroughly examined, offering rich prospects for further research initiatives. While current methodologies exhibit above-average accuracy, there is room for improvement through the exploration of enhanced techniques and more efficient methods. Considerations such as speed, reliability, and the handling of homonyms and homographs in both source and target languages merit attention. Therefore, this area remains an open field for extensive future research, holding implications for businesses, scientific domains, and user awareness.

In this investigation, we expound on the foundational principles of deep learning models and associated methodologies applied to sentiment analysis of social network data. Our approach involved preprocessing input data using TF-IDF and word embedding techniques before in putting it into deep learning models. We explored DNN, CNN, and RNN architectures, integrating them with TF-IDF and word embedding for sentiment analysis. A series of experiments were conducted, assessing the performance of these models on diverse datasets encompassing tweets and reviews across various subjects.

Furthermore, we reviewed pertinent literature in the field. The insights gained from our experiments and the existing studies offer a comprehensive perspective on the application of deep learning models to sentiment analysis and their integration with text preprocessing methodologies.

Following a review of relevant literature, CNN, DNN, and hybrid techniques emerged as the most prevalent models for sentiment analysis. An additional observation from the study was that commonly used approaches, such as RNN and CNN, are individually evaluated across different datasets in these papers, lacking a comparative analysis. Moreover, a majority of articles present outcomes solely in terms of accuracy, neglecting considerations for processing time.

References

[1]. K. Jain and S. Kaushal, "A Comparative Study of Machine Learning and Deep Learning Techniques for Sentiment Analysis," 2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 2018, pp. 483-487, doi: 10.1109/ICRITO.2018.8748793.

[2]. Sohangir, S.; Wang, D.; Pomeranets, A.; Khoshgoftaar, T.M. Big Data: Deep Learning for financial sentiment analysis. J. Big Data 2018, 5, 3

[3]. Ain, Q.T.; Ali, M.; Riaz, A.; Noureen, A.; Kamran, M.; Hayat, B.; Rehman, A. Sentiment analysis using deep learning techniques: A review. Int. J. Adv. Comput. Sci. Appl. 2017, 8, 424.

[4]. [4] Jangid, H.; Singhal, S.; Shah, R.R.; Zimmermann, R. Aspect-Based Financial Sentiment Analysis using Deep Learning. In Proceedings of the Companion of the The Web Conference 2018 on The Web Conference, Lyon, France, 23–27 April 2018; pp. 1961–1966

[5]. H. A. Shehu et al., "Deep Sentiment Analysis: A Case Study on Stemmed Turkish Twitter Data," in IEEE Access, vol. 9, pp. 56836-56854, 2021, doi: 10.1109/ACCESS. 2021.3071393.

[6]. Kraus, M.; Feuerriegel, S. Sentiment analysis based on rhetorical structure theory: Learning deep neural networks from discourse trees. Expert Syst. Appl. 2019, 118, 65–79.

[7]. Singhal, P.; Bhattacharyya, P. Sentiment Analysis and Deep Learning: A Survey; Center for Indian Language Technology, Indian Institute of Technology: Bombay, Indian, 2016.

[8]. Erenel, Z.; Adegboye, O.R.; Kusetogullari, H. A New Feature Selection Scheme for Emotion Recognition from Text. Appl. Sci. 2020, 10, 5351. https://doi.org/10.3390/app10155351

[9]. Abid, F.; Alam, M.; Yasir, M.; Li, C.J. Sentiment analysis through recurrent variants latterly on convolutional neural network of Twitter. Future Gener. Comput. Syst. 2019, 95, 292–308

[10]. Aggarwal, C.C. Neural Networks and Deep Learning;"Clustering — scikit-learn 0.24.2 documentation." [Online]. Available: https: //www.scikit-yb.org/en/latest/api/cluster/elbow.html

[11]. Ajay Shrestha and Ausif Mahmood. "Review of deep learning algorithms and architectures". In: IEEE Access 7 (2019), pp. 53040–53065. 45

[12]. Shanshan Yu, Jindian Su, and Da Luo. "Improving bert-based text classification with auxiliary sentence and domain knowledge". In: IEEE Access 7 (2019), pp. 176600– 176612.

[13]. Mickel Hoang, Oskar Alija Bihorac, and Jacobo Rouces. "Aspect-based sentiment analysis using bert". In: Proceedings of the 22nd Nordic Conference on Computational Linguistics (2019), pp. 187–196.

[14]. Lei Zhang, Shuai Wang, and Bing Liu. "Deep learning for sentiment analysis: A survey". In: Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 8.4 (2018), e1253.

[15]. Vineeth G Nair. Getting started with beautiful soup. Packt Publishing Ltd, 2014

[16]. S Chris Colbert et al. "The NumPy array: a structure for efficient numerical computation". In: Computing in Science Engineering. Citeseer. 2011.

[17]. Aurélien Géron. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. O'Reilly Media, 2019

[18]. Zhang, L.; Wang, S.; Liu, B. Deep learning for sentiment analysis: A survey. WIREs Data Min. Knowl. Discov. 2018, 8, e1253.