SN Computer Science (2020) 1:269
https://doi.org/10.1007/s42979-020-00279-9

ORIGINAL RESEARCH

Sentiment Analysis on Urdu Tweets Using Markov Chains

Zarmeen Nasim[1] · **Sayeed Ghani**[1]

Received: 27 January 2020 / Accepted: 30 July 2020 / Published online: 14 August 2020

Abstract
This paper presents a sentiment analysis approach based on Markov chains for predicting the sentiment of Urdu tweets. Sentiment analysis has been a focus of natural language processing (NLP) research community from the past few decades. The reason for this growing interest is twofold. First, the complexity involved in identifying sentiment from the unstructured text makes it a challenging problem for the research community. Second, sentiment analysis has a wide variety of applications ranging from industry to academia has made it a popular area in the research field of NLP. However, very little work has been done on sentiment analysis for the low resource languages which include Urdu, Bengali, Hindi, and other Asian languages. This work focuses on developing a 3-class (positive, negative, and neutral) sentiment classification model for the Urdu language. The experiments were conducted on the labeled corpus of Urdu tweets extracted from the Twitter network. One of the main contributions of this research includes the development of a large labeled corpus of Urdu Tweets for sentiment analysis. To the best of our knowledge, there is no such corpus available publicly in the Urdu Language. The labeled dataset is available on GitHub (https://github.com/zarmeen92/urdutweets). Furthermore, the results showed that the proposed approach outperforms the lexicon-based and traditional machine learning-based approaches of sentiment analysis.

Keywords Sentiment analysis · Markov chains · Urdu language · Opinion mining

Introduction

With the recent advances in social media, sentiment analysis has become an active area of research in natural language processing domain. Social media sites which include Twitter, Facebook, and Instagram enable people to share their opinion on a variety of topics with a large network of people. One of the most popular websites which we considered for this research is Twitter. Twitter has 330 million active users. More than 500 million tweets are posted every day on Twitter. Moreover, Twitter supports 40 languages including the Urdu language (https://blog.hootsuite.com/twitter-statistics/). With these interesting statistics, we can consider Twitter as one of the largest social networks.

Sentiment analysis, also known as opinion mining, is the technique of determining the semantic orientation or the polarity of the text. It comes under the umbrella of tasks performed in natural language processing. Over the past few decades, sentiment analysis has gained overwhelming popularity in the research community due to the rapid growth of social media websites. Sentiment analysis provides interesting insights into understanding the emotions of people at large. People tend to post on social events, political developments, religion, social life, and various other topics. Sentiment analysis is used to answer whether people are feeling positively or negatively about a certain topic of interest. We can understand the overall mood of society through the sentiment analysis of postings of people on social media sites. This information is helpful for market strategists to understand what the people are feeling about the recent advertisement of their product. Politicians can get insights into the sentiment of the public after any recent political action.

Sentiment analysis can be done at three different levels of granularities which include document-level sentiment analysis, sentence-level sentiment analysis, and aspect-level sentiment analysis. Document-level sentiment analysis deals with the overall opinion expressed in a whole document. A document may be considered as a collection of sentences. In sentence-level sentiment analysis, we deal with the sentiment expressed in a sentence. It is often observed that the

✉ Zarmeen Nasim
znasim@iba.edu.pk

Sayeed Ghani
sghani@iba.edu.pk

[1] Faculty of Computer Science, Institute of Business Administration (IBA), Karachi, Pakistan

document may not have a completely positive or negative sentiment; rather, it may comprise some positive and some negative sentiment sentences. Thus, sentence-level sentiment analysis reveals which part of the document has positive and which part has negative sentiments in a document. Aspect-level sentiment analysis is a more detailed analysis in comparison to document-level and sentence-level sentiment analysis. In aspect-level sentiment analysis, the opinion expressed for a certain attribute of a product or a service is captured. For instance, consider a review statement, "The food served here was delicious, however, the service provided by the staff was not impressive." In the given statement, the customer has expressed positive sentiment about the attribute *'food',* whereas negative sentiment was expressed for the *'service'* attribute.

Being a popular and active area of research, a variety of techniques have been developed for sentiment analysis. However, it has been observed that these techniques and approaches experiment on high resource languages such as English and Chinese. In the natural language processing domain, the language is termed as a high resource language, if there exist enough resources such as lexicons and corpus are available in a digital format for the NLP research community. The languages for which a limited or no resource is available in a digital format are termed as low resource languages. Very little research has been done on identifying sentiments in low resource languages such as Urdu, Bengali, Hindi, and other Asian languages.

Urdu is spoken by around 100 million people across the globe [1]. Due to the support of the Urdu language on the web, there is a huge corpus available on the web in the Urdu language. This opens the research opportunity in Urdu language processing. The existing techniques developed for sentiment analysis in English and other languages cannot be directly applied to the Urdu language due to the structural differences between these languages.

This research focuses on document-level sentiment analysis of tweets written in the Urdu language. The proposed methodology of sentiment analysis is based on Markov chains. Later on, the proposed approach is compared with machine learning and lexicon-based methods of sentiment analysis. Main contributions of this research are as follows:

(a) The development of a labeled corpus containing around 3000 Urdu Tweets. To the best of our knowledge, there is no such dataset publicly yet.
(b) A novel approach of sentiment analysis based on Markov Chains. Though the experiments were performed on Urdu tweets, the approach can be applied to any other language, as well.
(c) Comparison of the proposed approach against the existing approaches of sentiment analysis.

The rest of the paper is organized as follows. "Literature review" provides a brief literature review. "Research objective" presents the research objective. "Methodology" describes the proposed methodology. The results are discussed in section "Results". A comparative analysis between the proposed approach and other approaches of sentiment analysis is discussed in "Comparative analysis". Finally, "Conclusion" concludes the paper.

Literature Review

This section gives a brief literature review. The study conducted on the existing literature can be divided into further two sub-sections. The first part discusses the summary of approaches proposed for sentiment analysis. The second part presents the application of Markov models in natural language processing.

Sentiment Analysis

The different methodologies proposed in the literature for sentiment can be divided into two categories which include Lexicon-based methods and machine learning-based methods.

Lexicon-Based Methods

In lexicon-based methods, a list of words associated with sentiment polarities (positive/negative) is used to determine the polarity of a document. Each word in the document is assigned a polarity score using the sentiment lexicon. For instance, words such as good, excellent, fantastic, and fabulous can be considered as positive words and can be assigned a polarity score of + 1. In contrast to it, words such as bad, poor, pathetic, and sad can be listed as negative words with a polarity score of − 1. The overall polarity of the document is computed by aggregating the sentiment polarity score of words in a document.

Rehman and Bajwa [2] proposed a lexicon-based approach for analyzing sentiments expressed in Urdu corpus constructed using Urdu websites. The proposed approach utilized a publicly available sentiment lexicon of Urdu to assign positive weights to the words with positive semantic orientation and negative weight to the words with negative sentiment polarity. The overall polarity score of the document was determined by aggregating the polarity scores of words in a document.

Khan et al. [3] proposed another lexicon-based approach for analyzing the sentiment of Urdu tweets. The lexicon was constructed by translating various English lexicons into Urdu language using Google translator. The approach was then compared with different machine learning-based approaches. The evaluation results showed the effectiveness

of the proposed approach in comparison to machine learning-based approaches.

Abdulla et al. [4] discussed a lexicon-based approach for determining sentiment expressed in Arabic text. Al-Ayyoub et al. [5] also presented a sentiment analysis methodology based on the automatic construction of the lexicon in the Arabic language. The approach was evaluated using a dataset comprised of Arabic tweets. The construction of the Arabic lexicon was based on three steps. Initially Arabic stems were collected and processed. After processing, the translation was performed from Arabic to English. After translation, sentiment polarity of the Arabic stems was determined using English lexicons.

Trinh et al. [6] constructed an emotions dictionary to classify sentiments expressed in Facebook comments written in the Vietnamese language. Mukhtar et al. [7] presented a comparison of the lexicon-based approach of sentiment analysis against the supervised machine learning method. The experiments were conducted on Urdu blogs dataset. The results showed that the lexicon-based approach outperformed supervised machine learning methods.

Machine Learning-Based Methods

Machine learning-based methods can be further divided into two categories including supervised learning methods and unsupervised learning methods.

Supervised Learning Methods Supervised learning methods of machine learning require a labeled dataset for training the model. The training dataset comprises of text documents labeled with sentiment polarities.

Bilal et al. [8] performed sentiment analysis on opinions written in Roman–Urdu. They used Naive Bayes, decision trees, and KNN classification techniques to perform sentiment analysis. The experiment showed that Naive Bayes outperformed the other two algorithms.

Mukhtar and Khan [1] proposed a supervised learning method to identify sentiment polarity in Urdu blogs dataset. K-Nearest Neighbor, naïve Bayes, and support vector machines algorithm were employed and compared against each other. Kang et al. [9] discussed the sentiment analysis approach based on text-based hidden Markov models. An ensemble of text-based hidden Markov models using boosting and word clusters was used to compute the semantic polarity of the document. Soni and Sharaff [10] also proposed hidden Markov models for sentiment analysis of customer reviews. However, they have used part of speech tags for training their HMM model unlike [9].

Unsupervised Learning Methods Unsupervised learning methods do not require labeled training datasets which makes them more useful when the training corpus size is too big for manual labeling.

Li and Liu [11] proposed clustering-based approach for analyzing sentiments expressed in customer reviews. K-means clustering algorithm was employed along with some modifications to group the expressions containing similar sentiments in the same cluster. Zimmermann et al. [12] discussed a clustering-based approach to clustering similar product features before predicting sentiment polarity. Fuzzy c-means was employed for clustering product features.

Table 1 presents a summary of various approaches used for sentiment analysis. For the Urdu language, it was identified that the existing work of sentiment analysis is based on the lexicon-based approach and machine learning approaches based on SVM, Naïve Bayes, and other classification algorithms.

Applications of Markov Chain in NLP

Markov chains have been widely studied in the literature for various natural language processing tasks. Al-Anzi and AbuZeina [13] proposed a Markov chain-based approach for hierarchical classification of Arabic text documents. For each category of Arabic document, a probability transition matrix was computed, and then, the test data are passed to each of these matrices. The category label for which the score produced by the probability transition matrix was maximum was assigned to the given document.

In natural language processing, morphemes are considered as the basic semantic units. Morphological segmentation is the process of splitting text into morphemes. He et al. [14] applied hidden Markov chains for morphological segmentation in the Mongolian language. Mongolian affixes were identified using hidden Markov models to perform segmentation.

Table 1 Summary of sentiment analysis approaches

Paper	Approach	Language
[2]	Lexicon-based	Urdu
[3]	Lexicon-based	Urdu
[4]	Lexicon-based	Arabic
[5]	Lexicon-based	Arabic
[6]	Lexicon-based	Vietnamese
[7]	Lexicon-based	Urdu
[8]	Supervised learning-based	Roman-Urdu
[1]	Supervised learning-based	Urdu
[9]	A supervised approach using hidden Markov chains	English
[10]	A supervised approach using hidden Markov chains	English
[11]	An unsupervised approach using clustering	English
[12]	An unsupervised approach using clustering	English

Table 2 Application of Markov chains in NLP tasks

NLP task	Reference
Document classification	[13]
Morphological segmentation	[14]
Document clustering	[15, 16]
Twitter user location identification	[17]

Document clustering is the task of clustering similar documents together. Goyal et al. [15] used a Markov based model to cluster documents containing short-length text. Seara Vieira et al. [16] proposed hidden Markov models to represent the cluster of text documents.

Rodrigues et al. [17] addressed the problem of inferring user location on Twitter network from the text of the tweet and the friend network of a user. Markov chain Monte Carlo simulation was employed to learn the geographical labels from the friendship network.

To the best of our knowledge, the application of Markov chains for predicting the semantic orientation of text documents written in the Urdu language has not experimented before. Table 2 presents the application of Markov chains and its variants in various tasks related to natural language processing.

Research Objective

This research aims at identifying the semantic orientation of the Urdu tweets. The proposed methodology is inspired by the work of Al-Anzi and AbuZeina [13]. In the referenced work, the Markov chain approach was employed to perform hierarchical text classification of Arabic documents. The Markov chain approach based on the transition probability matrix of adjacent characters helped in reducing the dimensionality of feature space. In contrast to vector space models which use a large vocabulary of words to classify a document into predefined classes:

The objective of this research is to answer the following research questions.

(a) Can a Markov chain-based approach be used to identify the sentiment expressed by the twitter user in the tweet written in the Urdu language?
(b) Does the Markov chain approach perform better than the lexicon-based approach and other machine learning-based approaches of sentiment analysis experimented so far for the Urdu language?

To answer the first research question, the use of a Markov chain is evaluated for sentiment analysis of tweets posted in the Urdu language. In contrast to lexicon-based approaches and machine learning-based approaches that use a large vocabulary of words, a Markov chain-based approach captures the transition probability between consecutive characters present in a word. For each sentiment polarity class positive, negative, and neutral, a transition probability matrix is learned from the labeled training dataset. This produced three transition probability matrices. The test instance is passed to each of these transition probability matrices. The sentiment label for which the transition probability score is maximum is assigned to the test instance. The process of computing the transition probability score is described in detail in section "Methodology".

To answer the second question, a comparison of the Markov chain method was made against the lexicon-based method and the machine learning-based method of sentiment analysis.

Methodology

This section describes the proposed methodology. Figure 1 described the workflow developed for the execution of the proposed approach.

The following steps were involved in the process.

Collection of Tweets

A script written in python language was used to extract Urdu tweets from the Twitter Network. Twitter provides an API to

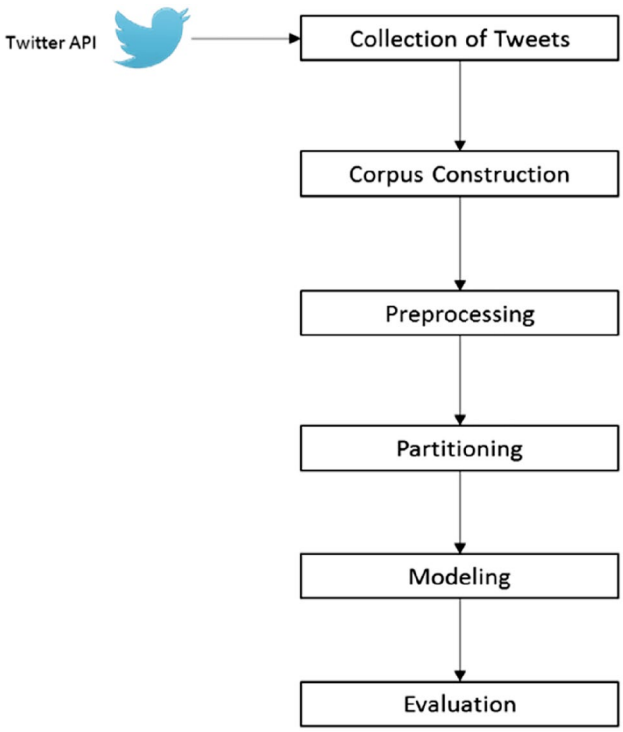

Fig. 1 Methodology of the proposed approach

download tweets. 3103 tweets that were posted for 1 week starting from 21st December 2017 to 27th December 2017 were downloaded.

Corpus Construction

The proposed approach was based on a supervised machine learning technique; therefore, a labeled dataset was required for training the proposed model. A corpus was constructed using an Active Learning approach [18].

In machine learning, obtaining a sufficiently large set of labeled instances is a challenging task. Active learning is the process of building a labeled corpus from a seed of high-quality labeled instances. The algorithm written for building a sufficient sized labeled corpus from a seed set of manually labeled samples worked as follows:

(a) Initially, a seed set consisting of 1400 tweets was labeled by three human annotators. The final label of the tweets in the seed set was decided using a majority voting scheme. Table 3 presents the distribution of tweets in the seed set.
(b) The seed set constructed in the previous step was used to train a classifier. A set of 980 tweets were used as the training set and the remaining tweets were used to evaluate the classifier. The classifier was trained using XGBoost classification algorithm [19] which is based on gradient boosted decision trees. Word vectors were used as the input feature of this classifier.
(c) The classifier trained in Step (b) was found to have an accuracy of 54% on the remaining 420 tweets. The classifier was used to predict the sentiment polarity of 1703 unlabeled tweets iteratively. During each iteration, a set of 100 unlabeled tweets were passed to the classifier to obtain polarity labels. The classifier returned the predicted sentiment label along with the probability of tweet belonging to the predicted sentiment label. The tweets for which the classifier predicts the label with 80% probability were added to the initial seed set generated in Step (a). In each iteration, tweets for which the classifier predicts the label with the probability below 80% were manually inspected by the human expert. After correction, these tweets were added to the initial seed set generated in Step (a).
(d) After increasing the size of the seed set in Step(c), the classifier is retrained using the same classification algorithm as described in Step (b).
(e) Steps (c) and (d) were repeated until all the unlabeled tweets were assigned sentiment polarity labels.

Table 3 Distribution of sentiment labels in seed set

Sentiment label	Count
Negative	618
Neutral	569
Positive	213
Total	1400

Table 4 Distribution of sentiment labels in the corpus

Sentiment label	Count
Negative	1604
Neutral	1171
Positive	328
Total	3103

Table 4 presents the distribution of sentiment polarity labels in the final dataset. The table shows that there is an imbalance distribution of sentiment labels in the dataset. In this way, a corpus comprised of 3103 labeled Urdu tweets was constructed.

Preprocessing

Preprocessing is an essential step in any natural language processing tasks. It is required to clean the text before using it in the modeling phase. The following are the preprocessing steps that were performed on the corpus of raw tweets.

(a) *Removal of stop words* Stop words are the common words frequently used in a language to support grammar and syntax but have no essential role in extracting semantics of the sentence. These include helping verbs, articles, and prepositions. A list of stop words in the Urdu language was obtained from https://githu

Table 5 Few examples of stop words

ہوگیے	ہے	ہوچکی	ہوٗے
یہ	کھولا	کیطرف	کہے
لیے	کرو	کررہے	کل
لگتی	والے	چلو	پھر

Table 6 Sample Tweets after preprocessing

S. No	Tweet	Tweet after preprocessing
1.	پيپلزپارٽي کا قافلہ کاميابي سے منزل کي جانب رواں دواں رہا، بلاول بھٹو	پیپلزپارٹی قافلہ کامیابی منزل جانب روان دوان بلاول بھٹو
2.	اداروں کي کارکردگي باعث شرم، انگلياں ہم پر اٹھتي ہيں، ارکان اسمبلي	اداروں کارکردگی باعث شرم انگلیاں اٹھتی ارکان اسمبلی
3.	سانحہ ماڈل ٹاؤن باقر نجفي رپورٹ کے خاص صفحے منظرعام پر آگئے	سانحہ ماڈل ٹاون باقر نجفی رپورٹ خاص صفحے منظرعام آگئے

Table 7 Distribution of dataset

Dataset	Size	Percentage
Training set	2172 tweets	70
Test set	931 tweets	30
Total	3103	100

b.com/stopwords-iso/stopwords-ur. The list contained 518 words. Table 5 shows a few examples of Urdu stop words.

(b) *Removal of Non-Urdu Characters* Non-Urdu characters such as English alphabets or numbers were removed from the given set of tweets.

(c) *Removal of URLs, Usernames, Special Characters,* etc. Tweets often contain URLs, hashtags, mentions, etc. These were removed during the preprocessing stage.

(d) *Removal of Emoticons* Emoticons were also removed from the tweets.

(e) *Removal of Diacritics* Diacritics in a language are used to support the pronunciation of the word. In Urdu language *Zer, Zabar, and Pesh* are diacritics. These diacritics were also removed during preprocessing.

(f) *Replacement of Selected Characters* In tweets, it was observed that the character ي was used in place of ى. Since ي belongs to the Sindhi language alphabet, therefore, we replaced ي with ى. Similarly, ں (*noon-ghuna)* was replaced by ن (*noon).* Table 6 shows a few examples of tweets before and after preprocessing.

Partitioning

After preprocessing, the dataset was partitioned into training and test dataset. The training dataset was used for learning the Markov chain model. The test dataset was used to evaluate the proposed model.

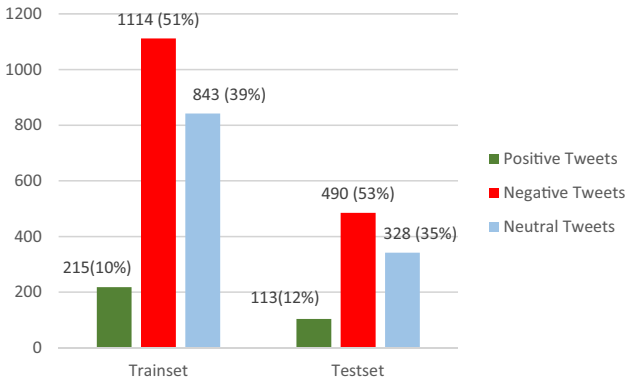

Fig. 2 Distribution of sentiment labels

Table 8 Statistics of word lists

	Word list	Tweets count	Size of word list
1.	Positive words list	215	1297
2.	Negative words list	1114	4766
3.	Neutral words list	843	3754
Total Tweets: 2172			

Table 9 List of frequent words

Positive words	Negative words	Neutral words
قبول	جرم	اجلاس
طاقت	احتجاج	صبح
کامیابی	طالبان	پروگرام
اعتماد	بحران	سماعت

Table 10 List of Urdu alphabets

S. no	Alphabet	S. no	Alphabet	S. no	Alphabet
1	آ	14	ذ	27	ف
2	ا	15	ر	28	ک
3	ب	16	ز	29	گ
4	پ	17	ڑ	30	ل
5	ت	18	ژ	31	م
6	ٹ	19	س	32	ن
7	ث	20	ش	33	ق
8	ج	21	ص	34	و
9	چ	22	ض	35	ہ
10	ح	23	ط	36	ی
11	خ	24	ظ	37	ے
12	د	25	ع	38	ھ
13	ڈ	26	غ	39	ء

The most commonly used split ratio in the machine learning community is 70:30, i.e., 70% of the dataset is used to train the model, whereas 30% of the dataset is used for evaluation purposes. Table 7 presents the distribution of the dataset into training and test set. 2,172 tweets were used to train the proposed model and 931 tweets were used to evaluate the proposed model.

Figure 2 presents the distribution of sentiment labels in the training and test sets.

Modeling

During the modeling phase, we follow the steps proposed by Al-Anzi and AbuZeina [13] for text classification using Markov Chains. The steps were as follows.

(a) *Creation of Word Lists* For each sentiment polarity label, a wordlist containing distinct words present in a set of tweets belonging to the same sentiment polarity label was created from the preprocessed training dataset. Table 8 presents the statistics of three wordlists created from the training dataset.

Table 9 presents examples of frequent words in each of the positive, negative, and neutral word lists.

Table 12 Transition of characters in sample Tweet

Words	Transitions
مصر	(م،ص)،(ص،ر)
مسجد	(م،س)،(س،ج)،(ج،د)
حملہ	(ح،م)،(م،ل)،(ل،ہ)
بحق	(ب،ح)،(ح،ق)
تعداد	(ت،ع)،(ع،د)،(د،ا)،(ا،د)

(b) *Training* After creating positive, negative, and neutral word lists, a character transition probability matrix was built for each sentiment class label. In a character transition probability matrix, each entry P_{ij} represents the probability of character i followed by character j. The characters were used to represent the states of the Markov Chain. The total number of characters in the Urdu language is 39, as shown in Table 10. Therefore, the dimension of the transition probability matrix was 39×39.

For learning a transition probability matrix of positive sentiment class, a set of positive preprocessed tweets filtered from the training set were used. For each of these tweets, words that were not present in the positive word list were removed. After removing these extra words, the transition probability between the successive characters of each word in the tweet was computed. The transition probability P_{ij} represents the number of times that the character i was followed by the character j in the set of tweets labeled with positive polarity in the training dataset. Table 11 presents the transition probability matrix learned for positive sentiment class.

A similar process was applied to learn the transition probability matrix for the negative sentiment class and neutral sentiment class. After the transition probability matrix was learned for each class, the matrix was normalized, such that the row-wise sum of the matrix is equal to 1. Consider the following sentence:

Tweet	مصر میں مسجد پر حملہ، جاں بحق افراد کي تعداد 80 ہوگئ
After preprocessing	مصر مسجد حملہ جان بحق تعداد ہوگئ
Sentiment Polarity	Negative
After filtering words from the Negative word list	مصر مسجد حملہ بحق تعداد

Table 11 Transition probability matrix for positive sentiment class

	ء	آ	ا	ب	ت	…	گ	ھ	ہ	ی	ے
ء	0	0	1	0	0	…	0	0	0	0	0
آ	0	0	0	0.16	0.04	…	0.05	0	0	0	0
ا	0.01	0	0	0.03	0.04	…	0.01	0	0.04	0.04	0.01
ب	0	0	0.16	0	0.03	…	0	0.17	0.05	0.1	0.01
ت	0	0	0.3	0.02	0	…	0.01	0.05	0.05	0.11	0.05
…	…	…	…	…	…	…	…	…	…	…	…
گ	0	0	0.23	0	0.01	…	0	0.01	0.04	0.12	0.09
ھ	0	0	0.17	0	0	…	0	0	0	0.16	0.05
ہ	0	0	0.14	0.03	0.08	…	0	0	0	0.25	0.06
ی	0	0	0.11	0.01	0.03	…	0.01	0	0.01	0	0.02
ے	0	0.01	0.05	0.03	0.03	…	0.06	0	0.06	0	0

The above sentence includes a transition between characters for each word, as shown in Table 12.

The transition probability matrix for the negative sentiment polarity computed for the above pair of transitions (i, j) where character i is followed by character j will be as shown below in Fig. 3.

Evaluation

During the evaluation phase, each instance in the test set was passed to the probability transition matrices learned during the modeling phase for each sentiment label. The sentiment label for which the transition probability score is maximum is assigned to the test document. The transition probability score is the summation of transition probabilities between two successive characters of each word present in a sentence.

The transition probability score is computed as follows:

$$\text{Transition Probability Score (TPS)} = \sum_{i=1}^{N} \sum_{j=1,k=j+1}^{C} p_{jk}^{(w_i)}, \quad (1)$$

where N is the total number of words in a sentence, C is the total number of characters in a word w_i, and $p_{jk}^{(w_i)}$ is the transition probability between characters' j and k of word w_i.

Consider a test tweet given below:

پاکستان خوشحالی کی طرف گامزن ہے

During preprocessing, the stop words (ہے، ہ، طرف، کی) will be removed and the tweet will be transformed as:

پاکستان خوشحالی گامزن

On the preprocessed tweet, the following operations will be performed to predict the sentiment polarity.

First, the tweet will be transformed using positive, negative, and neutral word lists. The wordlists were constructed during the modeling phase. During transformation, the words which do not exist in the respective wordlist will be removed from the tweet. This will result in three versions of the preprocessed tweet. The word گامزن does not exist in any of the word lists; therefore, it will be removed from each of the transformed version of the tweet. Similarly, the word خوشحالی was present in positive word list only; therefore, the tweet transformed using positive wordlist contains خوشحالی, as shown in Table 13.

Each transition probability matrix will generate a transition probability score. Table 14 presents the computation of a score from the transition probability matrix of positive sentiment class.

From the transition probability matrix learned for negative sentiment class, the score will be computed, as shown in Table 15.

Similarly, from the transition probability matrix learned for neutral sentiment class, the score will be computed, as shown in Table 16.

Table 17 presents the transition probability score (TPS) computed for each sentiment class on the test tweet.

Since the maximum score was generated from the transition probability matrix of the positive sentiment class, therefore, the proposed model will predict the sentiment polarity of the test tweet as **positive**.

Results

Following evaluation metrics were used for validating the predictions made by the proposed model:

	ا	ت	ج	ح	د	ر	س	ص	ع	ق	ل	م	ہ
ا	0	0	0	0	1	0	0	0	0	0	0	0	0
ت	0	0	0	0	0	0	0	0	1	0	0	0	0
ج	0	0	0	0	1	0	0	0	0	0	0	0	0
ح	0	0	0	0	0	0	0	0	0	0.5	0	0.5	0
د	1	0	0	0	0	0	0	0	0	0	0	0	0
ر	1	0	0	0	0	0	0	0	0	0	0	0	0
س	0	0	1	0	0	0	0	0	0	0	0	0	0
ص	0	0	0	0	0	1	0	0	0	0	0	0	0
ع	0	0	0	0	1	0	0	0	0	0	0	0	0
ق	0	0	0	0	0	0	0	0	0	0	0	0	0
ل	0	0	0	0	0	0	0	0	0	0	0	0	1
م	0	0	0	0	0	0	0.3	0.3	0	0	0.3	0	0
ہ	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 3 Probability transition matrix

Accuracy

Accuracy is defined as the ratio of correct predictions made by the model to the total number of predictions made by the model. The higher the accuracy, the better the performance of the model:

$$\text{Accuracy} = \frac{\text{Number of true predictions}}{\text{Size of the test dataset}}. \quad (2)$$

F-Score

F-Score is the geometric mean of the precision and recall of the model. **Precision** is defined as the ratio of correctly classified positive instances out of the total number of instances classified as positive by the model. A **recall** is the ratio of correctly classified positive instances to the total number of positive instances in the test dataset.:

$$F\text{ - Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (3)$$

The higher the value of *F*-Score, the better the classification results. Since the dataset has an imbalanced distribution of sentiment polarity labels, therefore, a weighted *F*-Score was computed. The weighted F-Score weighs the F-Score of the class by the number of true instances present for that class label in the dataset. Table 18 presents the evaluation results of the proposed Markov chain model on three-class sentiment classification.

The confusion matrix shown below in Table 19 presents the count of tweets classified correctly or incorrectly.

It was identified that the proposed approach was good at classifying negative tweets, whereas a high ratio of neutral and positive tweets was misclassified. This was due to the imbalanced distribution of class labels in the training dataset.

Furthermore, the proposed model was evaluated on the two-class classification of sentiments. The two classes considered were *positive* and *negative*. It was observed that the performance was greatly improved when the model was evaluated for 'positive' and 'negative' sentiment classes only. This was due to the reason that neutral tweets are often hard to label even for humans, whereas tweets containing positive sentiments can be easily distinguished from the tweets containing negative sentiments. Table 20 presents the evaluation results of the proposed Markov chain model on two-class sentiment classification.

Table 21 shows the confusion matrix of the two-class sentiment classification using the proposed approach. Due to the imbalanced distribution of positive tweets in the dataset, the approach misclassified positive tweets as negative tweets. The results can be further improved by increasing the ratio of positive tweets in the training dataset.

Comparative Analysis

To evaluate the effectiveness of the proposed approach, the comparative analysis was performed among the machine learning-based approach, lexicon-based approach, and the proposed approach.

Lexicon-Based Approach

The open-source Urdu lexicon (https://chaoticity.com/urdus entimentlexicon/) was used in the lexicon-based approach.

Table 13 Word list

Word list	Transformed Tweet using wordlist
Positive Word List	پاکستان خوشحالی
Negative Word List	پاکستان
Neutral Word List	پاکستان

It contained 2607 positive words and 4732 negative words. For each tweet, the sentiment score was computed using the lexicon:

$$\text{Sentiment Score} = \text{Count(Positive words)} - \text{Count(Negative words)}. \quad (4)$$

The tweet is assigned a 'positive' sentiment label if the sentiment score was positive. A 'negative' sentiment label is assigned to the tweet if the sentiment score was negative. The tweet was considered as neutral if the sentiment score was zero. Table 22 presents the results of the evaluation on the test dataset using a lexicon-based approach.

Table 14 Transition probability score computation for positive sentiment class on test case

Words	Transition pairs	Probability from transition probability matrix
پاکستان	(پ،ا)	0.26
	(ک،ا)	0.04
	(س،ک)	0.10
	(ت،س)	0.19
	(ا،ت)	0.29
	(ن،ا)	0.17
Total score of word پاکستان: 1.082		
خوشحالی	(خ،و)	0.28
	(و،ش)	0.04
	(ش،ح)	0.006
	(ح،ا)	0.14
	(ا،ل)	0.070
	(ل،ی)	0.196
Total score of word خوشحالی: 0.732		
Total score of پاکستان خوشحالی: 1.814		

Table 15 Transition probability score computation for negative sentiment class on test case

Words	Transition pairs	Probability from transition probability matrix
پاکستان	(پ،ا)	0.174
	(ا،ک)	0.036
	(ک،س)	0.076
	(س،ت)	0.134
	(ا،ت)	0.226
	(ن،ا)	0.172
Total score of word پاکستان: 0.820		
Total score of پاکستان: 0.820		

Table 16 Transition probability score computation for neutral sentiment class on test case

Words	Transition pairs	Probability from transition probability matrix
پاکستان	(پ،ا)	0.201
	(ا،ک)	0.039
	(ک،س)	0.085
	(س،ت)	0.178
	(ا،ت)	0.264
	(ن،ا)	0.167
Total score of word پاکستان:0.937		
Total score of پاکستان: 0. 937		

Machine Learning-Based Approach

In machine learning-based approach, a classifier was trained using the Naïve Bayes algorithm. A feature matrix was computed using the TF-IDF metric. The TF-IDF stands for Term Frequency-Inverse Document Frequency. TF-IDF is used to capture the importance of the word in a corpus. Table 23 shows the evaluation results of the machine learning-based approach for three-class and two-class sentiment classification.

Table 17 TPS computed using three transition probability matrices for the test case

Sentiment class	Transition probability score
Positive	1.814
Negative	0.820
Neutral	0.937

Table 18 Evaluation results on three-class sentiment classification

Evaluation metrics	Results
Accuracy	0.69 (69%)
F-Score	0.688

Table 19 Confusion matrix

True labels	Predicted labels		
	Negative (%)	Neutral (%)	Positive (%)
Negative	407 (83)	53 (11)	30 (6)
Neutral	100 (30)	180 (55)	48 (15)
Positive	35 (31)	21 (19)	57 (50)

Table 20 Evaluation results on two-class sentiment classification

Evaluation metrics	Results
Accuracy	0.865 (86.5%)
F-Score	0.857

Table 21 Confusion matrix of two-class sentiment classification

True labels	Predicted labels	
	Negative (%)	Positive (%)
Negative	461 (94)	27 (6)
Positive	51 (55)	41 (45)

Table 22 Evaluation results of lexicon-based approach

Evaluation metrics	Results
Three-class classification	
Accuracy	0.42 (42%)
F-Score	0.453
Two-class classification	
Accuracy	0.65 (65%)
F-Score	0.69

Table 23 Evaluation results of machine learning-based approach

Evaluation metrics	Results
Three-class classification	
Accuracy	0.65 (65%)
F-Score	0.59
Two-class classification	
Accuracy	0.83 (83%)
F-Score	0.76

Table 24 Comparative analysis

Evaluation metrics	Proposed approach	Lexicon-based approach	Machine learning-based approach
Three-class Classification			
Accuracy	**0.69 (69%)**	0.42 (42%)	0.66 (66%)
F-Score	**0.688**	0.453	0.596
Two-class classification			
Accuracy	**0.865 (86.5%)**	0.65 (65%)	0.84 (84%)
F-Score	**0.857**	0.69	0.77

Bold value indicate the highest values of accuracy and f-scores

Table 24 presents the result of comparison among the proposed approach, lexicon-based approach, and the machine learning-based approach. It was identified that the F-Score and accuracy obtained using the proposed approach were significantly greater than the F-Score and accuracy of the machine learning-based approach and lexicon-based approach on 3-class and 2-class sentiment classification. The lexicon-based approach performed the worst among the three approaches.

Conclusion

The proposed research aimed at building a sentiment analysis model for Urdu tweets. The methodology proposed in this paper was based on a Markov chain model. The experiments were conducted on the Urdu tweet corpus collected using Twitter API.

The sentiment analysis model developed using the proposed approach can have various applications. The model can be used to predict the sentiment of people from their tweets. This can help us to understand the mood of people on different social and political issues being discussed on the Twitter network. Various marketing strategists can use the proposed model to analyze the sentiment of people regarding their campaigns.

Furthermore, the evaluation performed on the test dataset showed that the proposed approach worked better than

the lexicon-based approach and machine learning-based approach of sentiment classification. However, we see that due to the presence of a very low ratio of positive tweets in the dataset, the model was not very good at predicting positive sentiment. The results can be further improved by increasing the training dataset for positive tweets.

Compliance with Ethical Standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Mukhtar N, Khan MA. Urdu sentiment analysis using supervised machine learning approach. Int J Pattern Recognit Artif Intell. 2018. https://doi.org/10.1142/S0218001418510011.
2. Rehman ZU, Bajwa IS. Lexicon-based sentiment analysis for Urdu language. In: 2016 6th international conference on innovative computing technology, INTECH 2016. 2017.
3. Khan MY, Emaduddin SM, Junejo KN. Harnessing English sentiment lexicons for polarity detection in Urdu tweets: a baseline approach. In: Proceedings—IEEE 11th international conference on semantic computing, ICSC 2017. 2017.
4. Abdulla NA, Ahmed NA, Shehab MA, et al. Towards improving the lexicon-based approach for arabic sentiment analysis. Int J Inf Technol Web Eng. 2014. https://doi.org/10.4018/ijitwe.2014070104.
5. Al-Ayyoub M, Bani Essa S, Alsmadi I (2015) Lexicon-based sentiment analysis of Arabic tweets. Int J Soc Netw Min.
6. Trinh S, Nguyen L, Vo M, Do P (2016) Lexicon-based sentiment analysis of Facebook comments in Vietnamese language. In: Studies in computational intelligence.
7. Mukhtar N, Khan MA, Chiragh N. Lexicon-based approach outperforms Supervised Machine Learning approach for Urdu Sentiment Analysis in multiple domains. Telemat Inform. 2018. https://doi.org/10.1016/j.tele.2018.08.003.
8. Bilal M, Israr H, Shahid M, Khan A. Sentiment classification of Roman-Urdu opinions using Naïve Bayesian, Decision Tree and KNN classification techniques. J King Saud Univ—Comput Inf Sci. 2016. https://doi.org/10.1016/j.jksuci.2015.11.003.
9. Kang M, Ahn J, Lee K. Opinion mining using ensemble text hidden Markov models for text classification. Expert Syst Appl. 2018. https://doi.org/10.1016/j.eswa.2017.07.019.
10. Soni S, Sharaff A. Sentiment analysis of customer reviews based on Hidden Markov Model. In: ACM international conference proceeding series. 2015.
11. Li G, Liu F. Sentiment analysis based on clustering: a framework in improving accuracy and recognizing neutral opinions. Appl Intell. 2014. https://doi.org/10.1007/s10489-013-0463-3.
12. Zimmermann M, Ntoutsi E, Spiliopoulou M. Extracting opinionated (sub)features from a stream of product reviews using accumulated novelty and internal re-organization. Inf Sci (Ny). 2016. https://doi.org/10.1016/j.ins.2015.06.050.
13. Al-Anzi FS, AbuZeina D. Beyond vector space model for hierarchical Arabic text classification: a Markov chain approach. Inf Process Manag. 2018. https://doi.org/10.1016/j.ipm.2017.10.003.
14. He M, Li M, Chen L. Mongolian morphological segmentation with hidden Markov model. In: Proceedings—2012 international conference on asian language processing, IALP 2012. 2012.
15. Goyal A, Jadon MK, Pujari AK. Spectral approach to find number of clusters of short-text documents. In: 2013 4th National conference on computer vision, pattern recognition, image processing and graphics, NCVPRIPG 2013. 2013.
16. Seara Vieira A, Borrajo L, Iglesias EL. Improving the text classification using clustering and a novel HMM to reduce the dimensionality. Comput Methods Programs Biomed. 2016. https://doi.org/10.1016/j.cmpb.2016.08.018.
17. Rodrigues E, Assunção R, Pappa GL, et al. Uncovering the location of Twitter users. In: Proceedings—2013 Brazilian conference on intelligent systems, BRACIS 2013. 2013.
18. Schohn G, Cohn D. Less is more: Active learning with support vector machines. Mach Learn Work Then Conf. 2000.
19. Chen T, Guestrin C. XGBoost: a scalable tree boosting system. In: Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining. 2016.