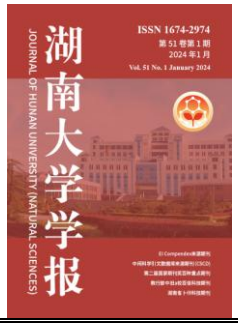

湖南大学学报（自然科学版）
Journal of Hunan University (Natural Sciences)

第 51 卷 第 1 期
2024 年 1 月

Available online at http://jonuns.com/index.php/journal/index

Vol. 51 No. 1
January 2024

Open Access Article https://doi.org/10.55463/issn.1674-2974.51.1.13

Event Extraction Using Word Clustering and Word Embedding for Roman Urdu

Asia Samreen[1,2]*, Hina Shakir[3], Muhammad Hussain[3], Muhammad Zuhair Arfeen[4], Syed Asif Ali[2]

[1] Department of Computer Science, Bahria University, Karachi, Pakistan

[2] Department of Computer Science, Sindh Madressatul Islam University, Karachi, Pakistan

[3] Department of Software Engineering, Bahria University, Karachi, Pakistan

[4] Department of Electrical Engineering, Bahria University, Karachi, Pakistan

* Corresponding author: asiasamreen.bukc@bahria.edu.pk

Received: October 19, 2023 / Revised: November 8, 2023 / Accepted: December 4, 2023 / Published: January 31, 2024

Abstract: Urdu is the national language of Pakistan, and Roman Urdu is a writing system representing the Urdu language using the Latin alphabet. The extraction of events from conversations conducted in Roman Urdu is a challenging task. This study aims to recognize multi-word events from text, specifically conversations among users of social networks. The expressed work consists of two phases: data preparation and event detection. In the first phase, four different approaches were investigated to remove useless and stop words in Roman Urdu, providing a comparison of the techniques. In the second phase, two different approaches were used to detect multi-word events in the cleaned text. Among the two, one is based on word clustering and the other is based on word embedding with the help of BiLM (a bidirectional language model). The clustering model efficiently finds events from conversations without complex feature engineering or data training. The second approach uses BiLSTM to obtain word vectors combined with NER (BIO) tagging to sequence the tagged dataset. Datasets for both approaches were prepared separately from the tweets. A comparison shows that word clustering can accommodate new data more efficiently, whereas a deep learning-based approach requires a huge amount of training data. The proposed approach detects the best sequence of two- or three-word events and single-word events in Roman Urdu. Since text-based events can play a significant role in finding the polarity or intensity of a conversation, two well-known techniques have been tested and evaluated.

Keywords: natural language processing, Roman Urdu, useless words, multi-word event extraction, bidirectional language models, event cluster.

使用羅馬烏爾都語的詞聚類和詞嵌入進行事件提取

摘要：乌尔都语是巴基斯坦的国语，罗马乌尔都语是使用拉丁字母代表乌尔都语的书写系统。从罗马乌尔都语对话中提取事件是一项具有挑战性的任务。这项研究旨在识别文本中的多词事件，特别是社交网络用户之间的对话。所表达的工作包括两个阶段：数据准备和事件检测。在第一阶段，研究了四种不同的方法来删除罗马乌尔都语中的无用词和停用词，并对这些技术进行了比较。在第二阶段，使用两种不同的方法来检测已清理文本中的多词事件。其中，一种是基于词聚类，另一种是借助 BiLM（双向语言模型）基于词嵌入。聚类模型可以有效地从对话中查找事件，无需复杂的特征工程或数据训练。第二种方法使用双向长短

期记忆网络 获取词向量并结合内尔（生物）标记对标记数据集进行排序。这两种方法的数据集都是与推文分开准备的。比较表明，词聚类可以更有效地容纳新数据，而基于深度学习的方法需要大量的训练数据。所提出的方法检测罗马乌尔都语中两个或三个单词事件和单个单词事件的最佳序列。由于基于文本的事件在发现对话的极性或强度方面可以发挥重要作用，因此已经测试和评估了两种众所周知的技术。

关键词：自然语言处理、罗马乌尔都语、无用词、多词事件提取、双向语言模型、事件集群。

1. Introduction

Urdu is one of the most widely used languages on social networks. For ease, most people used to type Urdu words in English alphabets, which is known as “Roman Urdu.” Roman Urdu lacks structure and grammatical rules, making it difficult to find stop words in the text [1].

We often have conversations in our native languages on social networks where most people express their opinions about some topic or event that has occurred recently. As Urdu typing is a tedious process, people use Roman Urdu to make chatting easier during conversations. It is a difficult task to process such text that includes not only special characters but also useless words, so it is better to extract keywords to extract some information regarding the topic from such texts.

1.1. Natural Language Processing Techniques

Natural language processing (NLP) techniques are methods used to process and analyze natural language data, such as text and speech, using computational algorithms and statistical models.

Commonly adopted NLP-based techniques include tokenization, part-of-speech (POS) tagging, named entity recognition (NER), and topic modeling. Word embedding is used to create a vector from words obtained from the tokenization process. For the under-consideration problem, we have specifically worked on NER, n-gram, and TF-IDF (Term Frequency-Inverse Document Frequency) along with machine learning algorithms. NER has been used to tag the event data, while TFIDF has been employed to obtain the word vector for ranked tokens.

1.2. Issues with Text Processing for Roman Urdu

Urdu text, or especially text written in Roman Urdu, requires a few more steps to obtain the word vector. The extra step in language preprocessing is the removal of useless words and their short forms, such as “hai” (is) being written as “h” or “ha”. Another issue is multi-spelled words like “bimaar” (sick) or “beemar” or “bemaar”. People also write incomplete words like “gmsm” (lost), which should be written as “gumsum”. These issues need to be tackled with special attention. Writing effective code to process the data according to an algorithm is necessary to address such problems.

1.3. Event Extraction

Event extraction is an NLP technique that involves automatically identifying and extracting information about events or key terms from text [2]. A useful tool for extracting information from textual data is NER, which can be effectively used for the automatic recognition of specific names from various fields such as weaponry names, clinical names, and archeological names [3]-[5]. To mitigate the drawbacks of individual approaches, hybrid solutions combine the advantages of several approaches [6]. Word cluster is the collection of words that may have a common global notion. For example, “Khushi” (happiness), “rahat” (euphoria), and “mazey” (fun), are all expressions of delight, which can be used while talking about an event such as “shadi” (marriage). Keyword extraction can also be referred to as keyword detection or keyword analysis. When event extraction techniques are applied, this text analysis approach facilitates a faster process of information extraction from text.

The presented research contributes to a novel approach for text-based event detection. Finding a list of Roman Urdu stop words is one of the contributions, as it presents a challenge while cleaning the text because there is no library to eliminate stop words for Roman Urdu. Another contribution is to apply two different approaches to detect the event from the conversation initiated on a social network.

2. Research Background

Textual events can be extracted in terms of words that may have the same or different meanings. However, phrases used in a close group discussion may have the same meaning [7]. Textual content can help identify offensive words from written text by using the n-gram model [1], but the hindrance is the mixed codes that are usually used on social networks, as these codes make it easy to chat or exchange opinions via text. One of the primary objectives of event extraction is to determine whether real-world events have been

recorded in articles and blogs [8]. Usually, researchers interested in issues relevant to linguistic features or text-based data consider English as the source. People want to talk or chat in their native language for ease, or they might think that opinions can be accurately conveyed in their native language. In the disciplines of preprocessing, feature extraction, lexicon, parsing, and classification, [9] presented studies done in Roman and Urdu. In [10], a framework for character-level event extraction via a graph attention network (CAEE) has been introduced. A comprehensive survey on the extraction of events from text-based data has been presented in [8].

Named entity recognition is employed for various purposes because it makes it easy to extract knowledge from words after tagging all the entities in the system. A corpus in Dutch has been formed, tagging all six entities mostly relevant to the system in the domain of archeology [4]. Morpheme level NE tagging and context-based NE tagging are provided for the Korean language in [5]. To find medical records electronically, a Chinese language-based clinical entity recognition model has been proposed, which is a pre-trained method [11]. It has been examined in [12] how long twins can be observed in a random word with a k-letter alphabet. ELMo (Embedding from Language Models) is a type of deep contextualized word representation that models both sophisticated aspects of word use and the variety of word use across linguistic contexts [13]. ELMo phrase vectors for any language collect both semantic and syntactic knowledge of the same type [14]-[15]. Deep learning has provided promising results for several types of natural language-based tasks [16]. The ANN model, with adequate fine tuning, could perform well in terms of accuracy in complicated multi-label tasks, including personality classification [17]. The studies reported in [18] compare classical machine learning algorithms with deep learning approaches for NLP challenges. Clustering of words might play a significant role in the extraction of valuable knowledge from text [19]-[21].

To obtain topic-based sentiment analysis, "Sentiment Word Co-occurrence and Knowledge Pair Feature Extraction with LDA" has been proposed, and experimental findings show that SKP-LDA performs better in terms of semantics and emotional content clustering [22].

3. Materials and Methods

The very first task was to normalize the collected text, which was in the form of conversations executed by certain group with IDs. Subsequently, two methods were employed for event identification or detection: K-means clustering and neural networks integrated with natural language processing techniques. Figure 1 illustrates the scenario in its entirety.

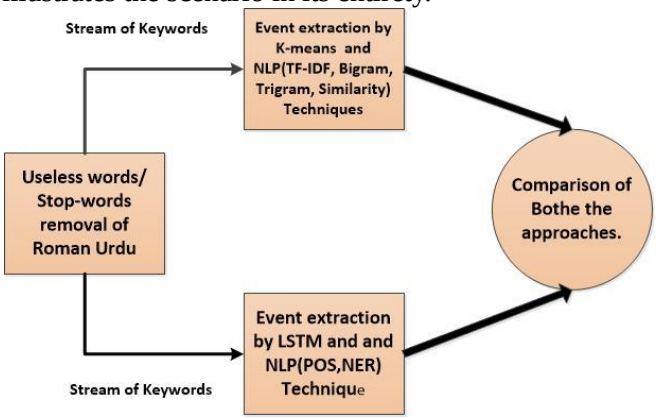

Fig. 1 Proposed technique for event extraction (Developed by the authors)

3.1. Initial Text Processing of the Dataset

We chose a dataset of Roman Urdu comments to test our strategy for extracting stop words from text. The first task to normalize the text is to remove emojis, hash tags, web addresses, numbers, and special characters or alphabets from other languages such as Urdu and Hindi.

All documents or texts are tokenized as $T=\sum W_i$, where T is the set of all words in the combined document D and W_i is the i^{th} token. It has been observed that single- and double-character words are as meaningful as desired; therefore, by applying the length count method, all such words are extracted. Table 1 shows the technique of the length count method, which pulls out useless words.

Table 1 Single and double letter removal (Developed by the authors)

Token Set T	Length Count	Removed Words
{'sahi','bt', 'Jin', 'ki', 'ibtedai', 'zindagi', 'aur', 'urooj', 'ki', 'daastan', 'qabil', 'e',' rashk', 'bhi, 'ha', 'dilchasp', 'b', 'aur', 'kai', 'hawalon', 'se' 'qabil', 'e', 'taqleed', 'bi'} (It is true that the story of his early life and rise is also enviable and interesting, and the references to him are also exemplary.)	Length Count (1)	'e'
{'sahi', 'bt', 'Jin', 'ki', 'ibtedai', 'zindagi', 'aur', 'urooj', 'ki', 'daastan', 'qabil','rashk', 'bhi, 'ha', 'dilchasp', 'b',' aur', 'kai', 'hawalon', 'se' 'qabil', taqleed', 'bi'} Remaining words: {'sahi', 'Jin', 'ibtedai', 'zindagi', 'aur', 'urooj', 'daastan', 'qabil', 'rashk', 'bhi', 'dilchasp','aur', 'kai', 'hawalon', 'qabil', 'taqleed'}	Length Count (2)	'bt', 'ki', 'se', 'bi'

3.2. Stop-Words Extraction

Method 1: TF-IDF Calculation

A corpus *D* can be defined as the collection of several documents d_1, d_2, … d_n, hence $D = \sum_{i=0}^{n} di$; where d_i is the i_{th} document. The number of

occurrences of a word in some document di can be calculated as:

$$tf_{(t,di)} = count(t, d_i) \quad (1)$$

where count is the function that computes the number of occurrences of t in di

Inverse document frequency reveals rare terms in the documents. For example, most tweets are meaningless or consist of repeated words; therefore, this measure can be used to find meaningful content. It can be calculated as

$$idf_t = \log_{10} \frac{N}{df_t} \quad (2)$$

where N represents the total number of documents and dft is the number of documents containing the word t. Equation 3 can simply be obtained by multiplying equations 1 and 2 to obtain the rare terms.

$$W_{td} = tf_{(t,di)} X\ idf_t = count(t, d_i) X\ \log_{10} \frac{N}{df_t} \quad (3)$$

Method 2: Set Union and Intersection

In this method [12], the pairwise intersections of the intersections of all documents are computed in two steps as $I_1 = D_1 \cap D_2$, $I_2 = D_2 \cap D_3 \ldots\ldots I_i = D_i \cap D_{i+1}$; then again, the intersections of all obtained sets are computed as $I'_1 = I_1 \cap I_2$, $I'_2 = i_2 \cap I_3 \ldots\ldots I'_k = I_k \cap I_{k+1}$; where D is any document, I is the pairwise intersection, and I' is the intersection of the pairwise intersection. Finally, the union of all intersections is computed as $S = \bigcup_{i=1}^{k} I'$ where k is the number of documents.

Method 3: Frequency Count

In this method, all words are tokenized first, and then repeated occurrences of each token are counted. A threshold number is used to select the top N words from the entire list.

Method 4: Zipf's Law

As asserted in [14], Zipf's law can be used to compute stop words for the Urdu language using frequencies and rank relationships. It states that the frequency of words defines their rank position. The most frequently repeated word will get the first rank, and rank 2 will be assigned to the second most frequently repeated word.

3.3. Event Extraction Using Clustering

K-means is a popular unsupervised technique in which an initial value of *k* is supplied and clusters are constructed progressively around that value. Various NLP tasks, such as putting comparable data elements together to identify underlying trends or finding similar terms, can be performed using clustering [20]. Before clustering, we obtain the keywords, as mentioned in Figure 2.

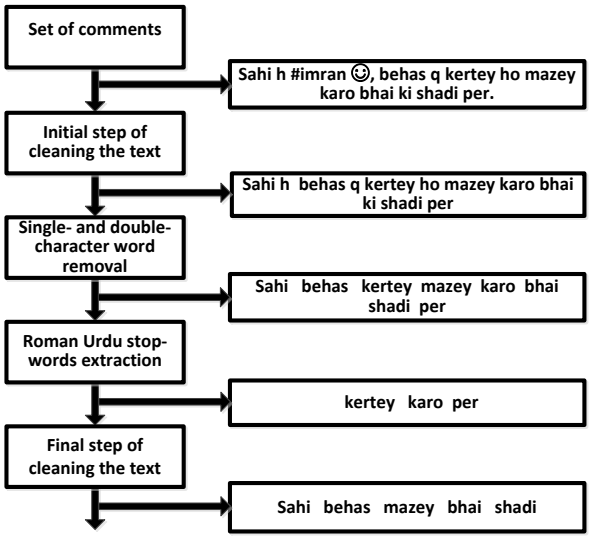

Fig. 2 Extraction of keywords (Developed by the authors)

To employ K-means to recognize some words as events, we have a dataset C = (C_1, C_2, C_3, ...) that consists of N conversations, and our goal is to extract words as events for each conversation C_i.

For this purpose, an intersection of top-ranked keywords and obtained word clusters was computed, which can be called an event cluster (EC). The whole process has been mentioned in the algorithm given below.

Algorithm: EC formation

1. Data cleaning
2. Data vectorization
3. Best bigram/trigram extraction
4. Applying the K-means model to data to form N clusters
5. Finding top-ranked words by applying TF-IDF
6. Event Term extraction

   If the top-ranked term is in cluster Ei

   If top-ranked terms occur in bigrams or trigrams

   then the event would be of two-word or three-word

   term and added to the EC

   else

   single word event term will be added to the EC

In Table 2, one-word, bigram, or trigram events are exemplified to understand the proposed technique for ordering two-word or three-word terms as events.

Table 2 Event type (Developed by the authors)

Sentence	Examples of an event	Event type
Apki shadi kab hogi? (When will you get married?)	"Shadi" (unigram)	One-word
Yes! Dekha tha wo cricket match bhi. (Yes, I watched the cricket match, too)	"cricket", "match" (bi-gram)	Two-word

Continuation of Table 2		
Eid miladun nabi ka intezaar hai bus. (Just waiting for Eid Miladunn Nabi)	“eid”, “miladun” , ”nabi” (tri-gram)	Three-word

3.4. Event Recognition Using NER and Neural Networks

Named entity recognition (NER) helps identify key terms in a text, such as the names of people, locations, food, medical equipment, and more. Tagging the major entities in a text aids in the sorting of unstructured data and the detection of notable information, which is critical when dealing with large datasets [18].

3.5. Preparing the Dataset

We considered raw data in the form of mixed wordings of Roman Urdu and English, but as our primary task was to work with Roman Urdu, the data were tagged only for words written in Roman Urdu. An event, as described before, could be of one word, consist of two words, or be a trigram. Analyzing the sets of conversations, it has been observed that active users use the word or words repeatedly for which they are talking about, like “shadi” (marriage) or “cricket match”. Table 3 shows some words that are tagged as events using the BIO tagging scheme. Each word has been tagged according to the same scheme.

Table 3 BIO tagging of events (Developed by the authors)

Sentence	**Tagged word as event**	**Tags**
Ap ki shadi kab ho gi? (When will you get married?)	“Shadi”	B-eve
Yes! Dekha tha wo cricket match bhi. (Yes, I watched the cricket match too)	“cricket”, “match”	B-eve, I-eve
Eid milad un nabi ka intezaar hai bus. (Just waiting for Eid Miladunn Nabi)	“eid”, “miladun”, “nabi”	B-eve, I-eve, I-eve

All other words are tagged as “O” for instance, in the first sentence “Ap”, “ki”, “kab”, “ho”, “gi”, and are tagged with “O” which means outsi.

3.6. Bidirectional Language Models

Bidirectional long-short term memory (biLSTM) is a technique that allows any neural network to store sequence information in both directions, backward and forward, as described in Figure 3.

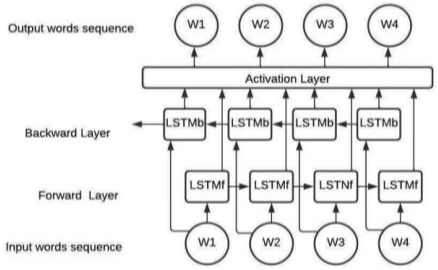

Fig. 3 Bidirectional LSTM operation (Developed by the authors)

A systemic approach has been presented in [15] for learning high-quality deep context-dependent interpretations from bi-directional language models (biLMs) and demonstrating significant improvements when using ELMo for various NLP tasks such as textual entailment, semantic role labeling, and named entity extraction. We applied ELMo embedding along with bidirectional LSTM to our dataset.

The standard method for developing a language model is to infer a word based on prior words. An n-gram model was used to find significant word sequences. The same task could be performed employing recurrent neural networks, but an RNN provides a limitless left context only of some target word.

To design a deep learning-based framework, two biLSTM layers are required: the first for ELMo word embedding and the second for processing the output from the embedding layer. The concatenated output from both forward and backward LSTM of the second layer is then sent to an additional layer to find the correct tags. Figure 4 presents the complete picture of the event recognition using a neural network.

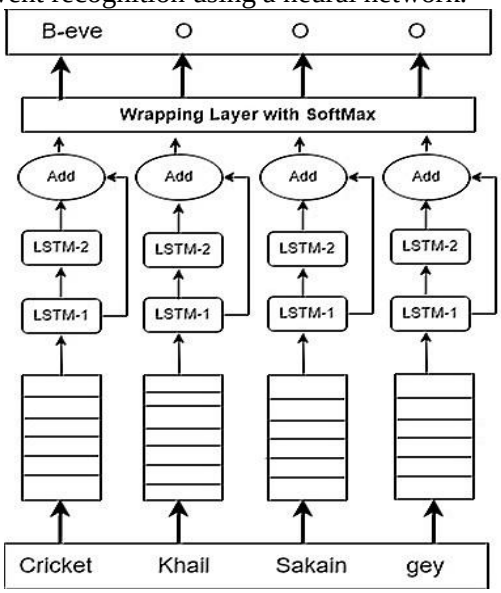

Fig. 4 Prediction of the event (Developed by the authors)

4. Experimental Results and Discussion

4.1. Roman Urdu Stop-Word /Useless-Word Removal

We used a sequential strategy to normalize the content. Data were first split down into individual words, and, after non-alphabetical terms were removed, words were rendered in lower case. Four approaches were explored to generate a list of useless and stop words for Roman Urdu, as shown in Tables 4-7.

In Table 4, the TF-IDF method results have been presented for Roman Urdu stop-word extraction. TF-IDF is supposed to work well while finding stop-words for any natural language.

Table 4 TF-IDF method (Developed by the authors)

Experiment #	No. of Docs	Total No. of words	Stop-words extracted by the TF-IDF method	
			Correct	Incorrect
1	392	3387	29	6
2	393	4760	24	2
3	325	6252	30	4
4	816	19768	60	13

Table 5 Union of intersection method (Developed by the authors)

Experiment #	No. of Docs	Total No. of words	Stop-words extracted by the Set IU method	
			Correct	Incorrect
1	392	3387	25	10
2	393	4760	21	4
3	325	6252	55	26
4	816	19768	161	80

It can be observed that the Set UI method does not work accurately, while in Table 6, the frequency method seems good, but its accuracy decreases when the count of top-extracted terms is increased.

Table 6 Frequency count method (Developed by the authors)

Experiment #	No. of Docs	Total No. of words	Stop-words extracted by the frequency count method	
			Correct	Incorrect
1	392	3387	19	2
2	393	4760	19	1
3	325	6252	20	0
4	816	19768	26	4

Zipf's law provides almost the same results as TF-IDF.

Table 7 Zipf's method (Developed by the authors)

Experiment #	No. of Docs	Total No. of words	Stop-words extracted by Zipf's method	
			Correct	Incorrect
1	392	3387	29	6
2	393	4760	24	2
3	325	6252	30	4
4	816	19768	60	13

All four of these strategies are compared in Table 8.

Table 8 Comparison of the four approaches (Developed by the authors)

No. of Documents	Accuracy scores			
	TF-IDF	Zipf's law	Count Method	Set UI Method
3387	0.79	0.88	0.80	.60
4760	0.91	0.96	0.95	0.80
6252	0.87	0.93	0.99	0.53
19768	0.78	0.84	0.83	0.50

We prepared a list of such useless words or stop-words using TF-IDF and Zipf's law for the proposed model. This list was then used to eliminate all unnecessary words from Roman scripts to obtain valuable terms. After cleaning the text, we use both English stop-word removal and Roman Urdu useless word removal. At the time of cleaning text, English stop-word removal is required as usually people write text as a mixture of English and their native language for instance "aap aur main only at khana, okay?" (translation: You and I will have dinner, okay?) which includes "only", "at" and "okay". The list of useless Roman Urdu words is given in Table 9.

Table 9 Useless words (Developed by the authors)

S. No.	Useless words	Meaning in English	S. No.	Useless words	Meaning in English
1	aaj	Today	24	Karte	Would do
2	aap	You	25	kay	Of the
3	acha	Good	26	kisi	Someone
4	aik	One	27	kiya	What
5	apney	Own	28	koi	Someone
6	aur	And	29	Lekin	But
7	bad	After	30	kon	Who
8	bhee	Too	317-85692+*-	liye	For
9	bohat	Much	32	Main	I
10	bus	Enough	33	Mein	In
11	diya	Gave	34	maira	Mine
12	hai	Is	35	Nahi	No
13	hain	Are	36	Rahe	Stay
14	hogaey	Will be	37	Sab	All
15	honge	Will be	38	Sath	With
16	hum	We	39	Sey	From
17	inho	They	40	Tau	So
18	jab	When	41	tha	Was
19	jis	Which	42	tum	You
20	kal	Yesterday	43	unhon	He
21	kabhi	Ever	44	waley	The ones
22	karey	Do it	45	woh	They
23	karo	Do it	46	yeh	This

4.2. Event Finding Using K-Means Clustering

One of the objectives of this research is the formation of an EC or event cluster that can also be used to perform NER tagging of a huge amount of data. About 10 different small datasets that contain texts ranging from 650 to 3000 sentences have been used to form EC.

N clusters are formed by the K-means algorithm, and the top three terms for each cluster are found. These terms are then matched with top-ranked tokens, and the cluster that has the maximum common terms with the top-ranked token is selected. To find the best order of these terms, we used bigrams and trigrams. The event is then added to the EC as a single, double, or triple term. The simple rule for constructing event terms is to take the intersection of top-ranked words and order the common terms according to bigrams and trigrams.

For example, if two top-ranked phrases occur in just one bigram pattern, the two-word event is added to EC; otherwise, a single-word event is added to EC. In table 10, for dataset 1, cluster 2 has all three terms, which are also top- ranked tokens. Therefore, the event will be a three-letter word, and with the help of a trigram, the event should be "plane crash Karachi". In dataset 3, two events "eidul fitar" and "plane crash" occur during the same time, but the text for event 1 is comparatively larger in size so significantly it appears as the event.

Table 10 Event extraction (Developed by the authors)

S. No.	Formation of an EC (Event Cluster)	
Dataset	Total No. of	1301

S. No.	Formation of an EC (Event Cluster)	
1	Sentences in Ci	
	No. of Words in Ci	21440
	Best Bigrams	[('plane', 'crash'), ('crash', 'karachi'), ('loved', 'one')]
	Best Trigrams	[('plane', 'crash', 'karachi'), ('pia', 'plane', 'crash'), ('plane', 'crash', 'crash'),]
	Top-ranked Words	‘crash’ ,’karachi’ ,’plane’
	Cluster	2
	Event added to the EC	[‘plane crash karachi’]
Dataset 2	Total No. of Sentences in Ci	1395
	No. of Words in Ci	13340
	Best Bigrams	[('please', 'play'), ('star', 'hina'), ('plz', 'play')]
	Best Trigrams	[('please', 'play', 'energetic'), ('please', 'play', 'right'), ('please', 'play', 'spring')]
	Top-ranked Words	‘khushi’,’play’,’please’
	Cluster	3
	Event added to the EC	[‘khushi’]
Dataset 3	Total No. of Sentences in Ci	695
	No. of Words in Ci	7836
	Best Bigrams	[('eidul', 'fitar'), ('fitar', 'mubarak'), ('plane', 'crash')]
	Best Trigrams	[('eidul', 'fitar', 'mubarak'), ('apko', 'eid', 'mubarik'), ('eidul', 'fitar', 'barkatain'),
	Top-ranked Words	‘fitar’,’eidul’,’mubarak’
	Cluster	2
	Event added to the EC	[‘eidul’,’fitar’,’mubarak’]
Dataset 4	Total No. of Sentences in Ci	3366
	No. of Words in Ci	27125
	Best Bigrams	[(‘imran’, ‘khan’), (‘prime’, ‘minister’), (‘gilgit’, ‘baltistan’)]
	Best Trigrams	[(‘minister’, ‘imran’, ‘khan’), (‘imran’, ‘khan’, ‘buzdar’), ('pakistan', 'imran', 'khan']
	Top-ranked Words	‘khan’, ‘imran’, ‘shadi’
	Cluster	2
	Event added to the EC	[‘imran khan’]

4.3. Event Recognition Using the NER Approach

The event recognition targets were met by combining NER tagging and ELMo vectorization with a neural network technique. The dataset for events was created by applying NER tagging. The network was trained using three epochs, and a report of assessment metrics was created, revealing an accuracy score of nearly 96 %, a recall score of approximately 86%, and an f1 score of more than 90 %, as shown in Table 11.

Table 11 Event recognition by NER and ANN (Developed by the authors)

Analysis Report	Precision	recall	f1-score	Support
Event Tag (eve)	0.96	0.86	0.91	222
Micro Avrg	0.96	0.86	0.91	255
Macro Avrg	0.98	0.85	0.91	255
Weighted Avrg	0.97	0.86	0.91	255

A comparison can be taken as given below for the two techniques. Both techniques can recognize multi-words event efficiently but clustering based technique is more flexible, as shown in Table 12.

Table 12 Comparison of the two techniques (Developed by the authors)

Event extraction technique	Significance of useless word removal	Multi-word event extraction	Accuracy	Flexibility of the technique
Clustering Based	Fasten the process of extraction	Automatically finds events consisting of multiple words	Depends on the frequency of event terms	New terms can be added to the event cluster.
LSTM based	No prominent effect on working	Multi-word events can be detected	More accurate if trained	Might generate incorrect results for new terms

5. Conclusion

Roman Urdu is widely used for text-based communication. Our work involves first detecting and removing unwanted words, followed by event extraction. In this paper, event extraction for Roman Urdu, which is a low-resource language, has been presented. This required additional processing such as Urdu stop-word detection and removal. For this purpose, we examined four approaches for removing unnecessary words. The results reveal that the Set UI approach is not applicable to all Roman Urdu datasets. All three additional approaches that employ the count of individual words or the frequency with which they appear in a document produce efficient results. We have constructed a list of useless words/stop-words for Roman Urdu by combining the results of all three approaches. The event was then extracted from the text using two distinct methods: applying word clustering and tagged sequence. In the clustering approach, all keywords are divided among N clusters, and then the cluster that has the top three words common with the top-ranked words is chosen to detect the event. Then, that event is kept in an event cluster, which can be used to detect event order, event trigger period, before-

events and after-events for some specific event, etc. The proposed model can efficiently find events from conversations without requiring complex feature engineering or training of the data. The second approach used ELMo word embedding to sequence the tagged dataset using NER (BIO) tagging and neural networks to recognize text-based events, and the system was given an event-tagged dataset for training. For both techniques, sets of conversations were used and datasets were created.

Our contribution to this research was to prepare a list of Urdu stop words that can be used to preprocess any text containing Roman Urdu. Another contribution is to maintain an event cluster along with identifying multi-word sequences as events, which can be used to analyze how texts function within social and cultural contexts.

Both of these strategies will be useful when dealing with textual events because the cluster of events will give out information not only about often occurring events but also about certain new textual occurrences, such as "Plane crash at Karachi." While NER techniques function quickly in conjunction with neural networks, once trained, they recognize events in a very fast and accurate manner. The main advantage of clustering is that it can identify some textual term(s) as an event without requiring any specific data. NER tagging and ANN can recognize an event quickly if trained, but finding events that are not included in the dataset system might produce incorrect results.

The proposed work can be used for various purposes, such as multi-word topic finding, hot trends in conversations, and polarity of opinions toward the topic, etc. Low-resource languages such as Roman Urdu are of keen interest to researchers, but the unavailability of digital processing facilities is a hindrance. Furthermore, our research might help them to preprocess the texts easily.

References

[1] AKHTER M P, JIANGBIN Z, NAQVI I R, ABDELMAJEED M, and SADIQ M T. Automatic detection of offensive language for urdu and roman urdu. *IEEE Access*, 2020, 8: 91213-91226. https://doi.org/10.1109/ACCESS.2020.2994950.

[2] ZHANG D, XU J, ZADOROZHNY V, and GRANT J. Fake news detection based on statement conflict. *Journal of Intelligent Information Systems*, 2022, 59(1): 1-20. https://doi.org/10.1007/s10844-021-00678-1

[3] LIU C, YU Y, LI X, and WANG P. Named entity recognition in equipment support field using tri-training algorithm and text information extraction technology. *IEEE Access*, 2021, 9: 126728-126734. https://doi.org/10.1109/ACCESS.2021.3109911

[4] BRANDSEN A, VERBERNE S, LAMBERS K, et al. Creating a dataset for named entity recognition in the archaeology domain. *Conference Proceedings LREC,* 2020, 4573-4577. https://aclanthology.org/2020.lrec-1.562

[5] JIN G, and YU Z. A Korean named entity recognition method using bi-LSTM-CRF and masked self-attention. *Computer Speech and Language*, 2021, 65: 101134. https://doi.org/10.1016/j.csl.2020.101134

[6] CHANDRASEKARAN D, and MAGO V. Evolution of semantic similarity—A survey. *ACM Computing Surveys*, 2021, 54(2): 1-37. https://doi.org/10.1145/3440755

[7] ANSAH J, LIU L, KANG W. et al. Leveraging burst in twitter network communities for event detection. *World Wide Web*, 2020, 23: 2851–2876. https://doi.org/10.1007/s11280-020-00786-y

[8] XIANG W, and WANG B. A survey of event extraction from text. *IEEE Access*, 2019, 7: 173111-173137. https://doi.org/10.1109/ACCESS.2019.2956831

[9] KHAN I U, KHAN A, KHAN W, et al. A Review of Urdu Sentiment Analysis with Multilingual Perspective: A Case of Urdu and Roman Urdu Language. *Computers*, 2022, 11(1): 3. https://doi.org/10.3390/computers11010003

[10] WU X, WANG T, FAN Y, and YU F. Chinese Event Extraction via Graph Attention Network. *Transactions on Asian and Low-Resource Language Information Processing*, 2022, 21(4): 1-12. https://doi.org/10.1145/3494533

[11] GONG L, ZHANG Z, and CHEN S. Clinical named entity recognition from Chinese electronic medical records based on deep learning pertaining. *Journal of Healthcare Engineering*, 2020: 8829219, https://doi.org/10.1155/2020/8829219

[12] DUDEK A, GRYTCZUK J, and RUCIŃSKI, A. Long twins in random words. *arXiv preprint*, 2021, ARXIV:2112.14197. https://doi.org/10.48550/arXiv.2112.14197

[13] LAGUTINA K, and LAGUTINA N. A Survey of Models for Constructing Text Features to Classify Texts in Natural Language. *Proceedings of the 29th Conference of Open Innovations Association*, 2021: 222-233, IEEE. https://doi.org/10.23919/FRUCT52173.2021.9435512

[14] SARZYNSKA-WAWER, J, WAWER, A, PAWLAK, A, et al. Detecting formal thought disorder by deep contextualized word representations. *Psychiatry Research*, 2021, 304: 114135. https://doi.org/10.1016/j.psychres.2021.114135

[15] PETERS M E, NEUMANN M, IYYERM, et al. Deep contextualized word representations. *arXiv preprint*, 2018, https://doi.org/10.48550/arXiv.1802.05365

[16] TORFI A, SHIRVANI R A, KENESHLOO Y, et al. Natural language processing advancements by deep learning: A survey. *arXiv preprint,2020*, https://doi.org/10.48550/arXiv.2003.01200

[17] BAŞARAN S, and EJIMOGU O. H. A neural network approach for predicting personality from Facebook data. *Sage Open*, 2021, 11(3): 21582440211032156. https://doi.org/10.1177/21582440211032156

[18] MAGNINI B, LAVELLI A, and MAGNOLINI S. Comparing machine learning and deep learning approaches on NLP tasks for the Italian language. *Proceedings of the 12th Conference on Language Resources and Evaluation*, 2020: 2110-2119. https://aclanthology.org/2020.lrec-1.259

[19] KHAN A R, KARIM A, SAJJAD H, et al. A clustering framework for lexical normalization of Roman Urdu. *Natural Language Engineering*, 2022, 28(1): 93-123. https://doi.org/10.1017/S1351324920000285

[20] AL-AZZAWY D S, and AL-RUFAYE F. M L. Arabic words clustering by using K-means algorithm. *Proceedings of the 2017 Annual Conference on New Trends*

in Information & Communications Technology Applications, 2017: 263-267. IEEE. https://doi.org/10.1109/NTICT.2017.7976098
[21] BALAJI B S, BALAKRISHNAN S, VENKATACHALAM K, and JEYAKRISHNAN V. Automated query classification based web service similarity technique using machine learning. *Journal of Ambient Intelligence and Humanized Computing*, 2021, 12(6): 6169-6180. https://doi.org/10.1007/s12652-020-02186-6
[22] WU D, YANG R, and SHEN C. Sentiment word co-occurrence and knowledge pair feature extraction based LDA short text clustering algorithm. *Journal of Intelligent Information Systems*, 2021, 56(1): 1-23. https://doi.org/10.1007/s10844-020-00597-7

参考文:

[1] AKHTER M P、JIANGBIN Z、NAQVI I R、ABDELMAJEED M 和 SADIQ M T。乌尔都语和罗马乌尔都语攻击性语言的自动检测。IEEE 访问，2020 年，8：91213-91226。https://doi.org/10.1109/ACCESS.2020.2994950。
[2] ZHANG D, XU J, ZADOROZHNY V, 和 GRANT J. 基于陈述冲突的假新闻检测. 智能信息系统学报, 2022, 59(1): 1-20. https://doi.org/10.1007/s10844-021-00678-1
[3] 刘超,余勇,李新,王鹏.利用三重训练算法和文本信息提取技术的装备保障领域命名实体识别. IEEE 访问，2021 年，9：126728-126734。https://doi.org/10.1109/ACCESS.2021.3109911
[4] BRANDSEN A、VERBERNE S、LAMBERS K 等人。创建用于考古领域命名实体识别的数据集。会议论文集 LREC，2020 年，4573-4577。https://aclanthology.org/2020.lrec-1.562
[5] JING G, 和 YU Z. 一种使用双长短期记忆网络-病例报告表和掩蔽自注意力的韩国命名实体识别方法。计算机语音和语言，2021，65：101134。https://doi.org/10.1016/j.csl.2020.101134
[6] CHANDRASEKARAN D，和 MAGO V。语义相似性的演变——一项调查。ACM 计算调查，2021 年，54(2)：1-37。https://doi.org/10.1145/3440755
[7] ANSAH J，LIU L，KANG W.等。利用推特网络社区的爆发进行事件检测。万维网，2020 年，23：2851–2876。https://doi.org/10.1007/s11280-020-00786-y
[8] 项文，王波。文本事件抽取综述。IEEE 访问，2019，7：173111-173137。https://doi.org/10.1109/ACCESS.2019.2956831
[9] KHAN I U，KHAN A，KHAN W，等。多语言视角下的乌尔都语情感分析综述：以乌尔都语和罗马乌尔都语为例。计算机，2022 年，11(1)：3。https://doi.org/10.3390/computers11010003
[10] WU X, WANG T, FAN Y, 和 YU F. 基于图注意力网络的中文事件提取。亚洲和低资源语言信息处理学报，2022，21(4)：1-12。https://doi.org/10.1145/3494533
[11] 宫丽，张志，陈思. 基于深度学习的中文电子病历临床命名实体识别. 医疗保健工程杂志，2020 年：8829219, https://doi.org/10.1155/2020/8829219
[12] DUDEK A、GRYTCZUK J 和 RUCIŃSKI, A. 随机词中的长双胞胎。arXiv 预印本，2021 年，ARXIV：2112.14197。https://doi.org/10.48550/arXiv.2112.14197
[13] LAGUTINA K 和 LAGUTINA N。构建文本特征以对自然语言文本进行分类的模型调查。开放创新协会第 29 届会议论文集，2021 年：222-233，IEEE。https://doi.org/10.23919/FRUCT52173.2021.9435512
[14] SARZYNSKA-WAWER, J、WAWER, A、PAWLAK, A 等人。通过深层语境化的单词表征来检测形式思维障碍。精神病学研究，2021，304：114135。https://doi.org/10.1016/j.psychres.2021.114135
[15] PETERS M E、NEUMANN M、IYYERM 等人。深度语境化的单词表示。arXiv 预印本，2018 年，https://doi.org/10.48550/arXiv.1802.05365
[16] TORFI A、SHIRVANI R A、KENESHLOO Y 等。深度学习带来的自然语言处理进步：一项调查。arXiv 预印本，2020 年，https://doi.org/10.48550/arXiv.2003.01200
[17] BAŞARAN S 和 EJIMOGU O. H. 一种从 Facebook 数据预测个性的神经网络方法。圣人公开赛，2021 年，11(3)：21582440211032156。https://doi.org/10.1177/21582440211032156
[18] MAGNINI B、LAVELLI A 和 MAGNOLINI S。比较意大利语自然语言处理任务上的机器学习和深度学习方法。2020 年第十二届语言资源与评估会议论文集：2110-2119。https://aclanthology.org/2020.lrec-1.259
[19] KHAN AR、KARIM A、SAJJAD H 等人。用于罗马乌尔都语词汇规范化的聚类框架。自然语言工程，2022，28(1): 93-123. https://doi.org/10.1017/S1351324920000285
[20] AL-AZZAWY D S, 和 AL-RUFAYE F. M L. 使用 K-均值算法进行阿拉伯语单词聚类。2017 年信息通信技术应用新趋势年会论文集，2017：263-267。IEEE。https://doi.org/10.1109/NTICT.2017.7976098
[21] BALAJI B S、BALAKRISHNAN S、VENKATACHALAM K 和 JEYAKRISHNAN V。使用机器学习的基于网络服务相似性技术的自动查询分类。环境智能与人性化计算学报，2021，12(6): 6169-6180. https://doi.org/10.1007/s12652-020-02186-6
[22] 吴德，杨瑞，沉春. 基于 LDA 短文本聚类算法的情感词共现和知识对特征提取。智能信息系统学报，2021，56(1): 1-23。https://doi.org/10.1007/s10844-020-00597-7