VFAST Transactions on Software Engineering
https://vfast.org/journals/index.php/VTSE@ 2025, ISSN(e): 2309-3978, ISSN(p): 2411-6246
Volume 13, Number 1, January-June 2025 *pp: 20-39*

Advancing NLP for Shahmukhi Punjabi: Word Embedding and Text Classification with a Novel Dataset

Muhammad Shabbir [1*], **Shereen Fatima Bhatti** [2], **Raja Sohail Ahmed Larik** [3], **Ali Orangzeb Panhwar** [4], **Muhammad Saif** [5], **Asadullah Kehar** [6]

[1]Department of Computer Science, Sindh Madresstual Islam University, Pakistan; [2]Department Of Computer Science, The Shaikh Ayaz University, Shikarpur.; [3]Department of Computer Science, Ilma University, Karachi, Sindh, Pakistan; [4]Faculty of Computer and Engineering Sciences, SZABIST Gharo Campus; [5] Department of Computer Science, Sindh Madresstual Islam University, Pakistan ; [6]Institute of Computer Science, Shah Abdul Latif University, Khairpur

Keywords:*Punjabi, Shahmukhi, NLP, Word Embedding, Classification, Machine Learning.*

Journal Info:
Submitted:
November 5,2024
Accepted:
January 10, 2025
Published:
January 29, 2025,

Abstract

The Punjabi language occupies a large pool in today's era; millions speak it. Only in Pakistan, 80 million people speak the Punjabi language in the province of Punjab. However, with this Big Market Cap, there has yet to be any proper research available. This research focuses on the Punjabi Language, especially the Shahmukhi Punjabi language, famous in Pakistan and Asia. However, it needs to be given more attention in the existing research. There needs to be an adequately supervised established dataset available with large data. Till now, there has yet to be any proper research on Word Embedding and Classification. This paper introduced the crafted dataset for the Shahmukhi Punjabi language dataset. It is also based on advanced NLP techniques like Word2Vec and SDfasttext for Word Embedding to capture the semantic relation within the language. In addition, we investigated the applications of six distinct classification models to analyze four different categories: News, Ghazal, Dohra, and Poetry. The notable success of the Naive Bayes classifier with other classification models lays the groundwork for future research and applications in natural language processing for the Punjabi language. The study encourages further exploration and the development of tailored solutions to meet the linguistic diversity in digital environments and apply deep learning models.

***Correspondence author email address:** m.shabbir1047@gmail.com
DOI: 10.21015/vtcs.v13i1.2093

1 Introduction

Punjabi is a rich morphological, multiscript, multidialectal language [1] and is the most widely spoken Indo-Aryan language. In the 21st century, there were about 33 million speakers of the Punjabi language in India. It is the official language of the Punjab state of India, and the Indian constitution recognizes it. Also, Punjabi is the official language of the Punjab province of Pakistan and the native language of 32 percent of the country's population. There are about 80M Punjabi speakers in the largest province in the country. It is the most spoken language after Urdu in Pakistan. There are also overseas communities who are Punjabi speakers in the UK, Canada, Malaysia, and Saudi Arabia, which is growing in Australia. It is written in India's Gurmukhi script and Pakistan's Shahmukhi script. Despite this, Punjabi has a rich historical and literal heritage [2]. The term "SNPL study" was coined in 2002, but it gained popularity with the development of its Unicode system. However, due to a lack of fundamental NLP ways, for example, unprocessed data may help train the best word embedding and algorithm for machine learning; Shahmukhi language is less resourced language now because their dataset is not created yet, and it requires Human made effort because it is in raw form now.

Language Materials are a priority component for making the best-quality natural language systems using auto-processing. LR has written and spoken datasets, monolingual dictionaries, and noted datasets for detailed computational reasons. Making such material has extended the exposure of researchers working on natural language. Many world languages, including English, Chinese, and others, have similar language processing facilities built into their software products. Shahmukhi language needs more starting resources to train this word embedding and make the stand-alone NLP application, for example, semantic analysis [3], sentiment modeling POS tagging, NER and machine translation, or multitasking. Punjabi is now widely used in Pakistan and India for online communication, publications, and public organizations. However, more work needs to be done to build LRs, such as raw corpus and annotated corpus. To our knowledge, Punjabi does not have a large unlabeled dataset from which to construct and assess word embeddings for Statistical Punjabi Language Processing (SPLP).

Learn word embeddings from unlabeled corpora to break out of this cycle, which may be utilized to bootstrap various downstream NLP processes. In the context of semantic directional size and partition presentation and partition of models like semantics [4] and word embeddings, a unique termite is an LM technique for joint the words and sentences into n-directional sreal-number directions that graphically showing the semantic and syntactic relationship with your following words. Einstein" and "Scientist" sound more alike than "Einstein" and "doctor." Word pre-processing achieves the essential morphological indication of "a word is categorized by the corporation it preserves" in this mode. NN-built prototypes [5] have recently shown cutting-edge performance in various NLP tasks employing word embeddings. One advantage of such methods is that they employ hearsay ways to knowledge demonstrations and do not need a marked amount, which is unusual for the low-resource Punjabi language. These representations may be learnt on large unannotated corpora and then used in NLP tasks that need only a small amount of tagged data.

We address corpus creation concerns in this study by constructing a huge corpus of over 3.9 million arguments from diverse websites using the scrapper tools. After acquiring it, we extensively pre-processed the dataset to filter out noisy data such as HTML elements and English terms. The statistical analysis for the small words and letters, word scores, and stop-word identification is also given. Finally, the dataset makes Shahmukhi Punjabi word embedding using cutting-edge GloVe SG and CBoW algorithms [6].

The inherent methodology frequently comprises an already-selected collection of question keywords and semantically comparable target terms known as query words. The recommended one-by-one embedding of words is also linked to the newly found Punjabi fast Text (SD fast Text) word demonstrations [7]. To the top of our information, this is the initial systematic work on building large corpora and producing word embeddings for low-resource Punjabi. Our novel contributions are summarized below:

Punjabi is a rich morphological, multiscript, multidialectal language [1] and is the most widely spoken Indo-Aryan language. In the 21st century, there were about 33 million speakers of the Punjabi language in India. It is the official language of the Punjab state of India, and the Indian constitution recognizes it. Also, Punjabi is the official language of the Punjab province of Pakistan and the native language of 32 percent of the country's population. There are about 80M Punjabi speakers in the largest province in the country. It is the most spoken language after Urdu in Pakistan. There are also overseas communities who are Punjabi speakers in the UK, Canada, Malaysia, and Saudi Arabia, which is growing in Australia. It is written in India's Gurmukhi script and Pakistan's Shahmukhi script. Despite this, Punjabi has a rich historical and literal heritage [2].

Language Materials are a priority component for making the best-quality natural language systems using auto-processing. LR has written and spoken datasets, monolingual dictionaries, and noted datasets for detailed computational reasons. Making such material has extended the exposure of researchers working on natural language. Many world languages, including English, Chinese, and others, have similar language processing facilities built into their software products.

We address corpus creation concerns in this study by constructing a huge corpus of over 3.9 million arguments from diverse websites using the scrapper tools. After acquiring it, we extensively pre-processed the dataset to filter out noisy data such as HTML elements and English terms. The statistical analysis for the small words and letters, word scores, and stop-word identification is also given. Finally, the dataset makes Shahmukhi Punjabi word embedding using cutting-edge GloVe SG and CBoW algorithms [6].

1. We provide a large corpus of over 3.9 million words culled from numerous online sites, as well as a list of Punjabi stop words.
2. Use the GloVe, CBoW, and SG Word2Vec algorithms to create word embeddings, then calculate and relate them via the essential valuation techniques of the similarity models.
3. We are the first to compare SdfastText word representations to the Punjabi word embeddings we proposed.
4. We apply seven different classification models, compare them, and Predict the data with 95 percent accuracy.

The left-over work is divided into monitors: Division 2 summarises the literature on computing resources, Punjabi corpus creation, and word embedding models. Section 3 covers the methods used, and Section 4 includes statistical analysis of the created corpus. The experiments and results are set up in Section 5. Section 6 contains the findings of the intrinsic evaluation and comparison. Section 7 contains the discussion and future work; Section 8 concludes the paper.

2 POS Tagging

PUNJABI	ENGLISH	POS
تحریک انصاف دے شرپسنداں نال کسی قسم دی رعایت نہ ورتی جائے مریم نواز	marium nawaz said that do not make any concession to the miscreants of the Tehreek-e-Insaf community	**Noun:** Mariam Nawaz, concession, miscreants, Tehreek-e-Insaf **Verb:** said, make **Adjective:** any, kind, miscreants **Adverb:** Not **Preposition:** to, of **Conjunction:** That **Interjection:** None in this sentence

PUNJABI	ENGLISH	POS
عمران خان دی رہائی تک تحریک انصاف پُرامن احتجاج جاری رکھے گی پرویزالہٰی	Tehreek-e-Insaf will continue peaceful protest until Imran Khan is released.	**Noun:** Tehreek-e-Insaf, protest, Imran Khan **Verb:**will continue, is released **Adjective:**peaceful **Adverb:** until **Preposition:** during **Conjunction:** and **Interjection:** None in this sentence
پنجاب وچ فوج بُلا لئی گئی	Army has been called in Punjab.	**Noun**:Army, Punjab **Verb:**has called **Adjective:** None in this sentence **Adverb:** None in this sentence **Preposition:** In **Conjunction:** None in this sentence **Interjection:** None in this sentence

Table 1. Part Of Speech Tagging Table

3 Related Work

The Shahmukhi Punjabi raw and annotated corpus is a helpful addition to constructing resources such as under-done and marked corpus for POS labeling and Grammarly analysis. However, the corpus was gathered from many news websites and genres. For Punjabi word segmentation, the raw corpus is used. Many news and social blog websites have recently gathered a resource development initiative. The command of word embeddings in natural language processing was assessed by as long as a neural linguistic prototype and more than one task learning.

However, word embeddings in Deep NA [8] have recently become a key component for deep NLP performance acceleration. CBoW and SG, two famous word2vec neural constructions [9], make the best quality directional illustrations at a low-priced hardware cost by joining words-level knowledge on immense datasets in relation to semantic and syntactic word comparison [10]. The evaluation of word embeddings may be addressed in two ways: internally and externally. A simple link approach for intrinsic assessment of word embeddings to determine the vector space next to a query word. This method can minimize pre-judice and offer insights for data-driven decision-making. Meanwhile, an extrinsic assessment technique [11] calculates recital in downstream Natural Language Processing tasks such as named-entity recognition and parts-of-speech tagging for classification [12]; we apply different models, but the best and the highest accuracy comes to the naïve byson model [13]. However, the Punjabi language lacks an annotated corpus for such assessments, making extrinsic evaluations difficult and time-consuming. As a result, we chose an intrinsic assessment approach to assess the quality of recommended Punjabi word embeddings quickly. The cosine distance between related terms in a dataset is calculated. We fine-tuned hyperparameters using the CBoW, SG, and Glove models [13] to construct robust Punjabi word embedding [14].

Table 2. Table For Related Work

REFRENCE	PROPOSED	FINDING	LIMITATION
m.ahmed and m.ilyas at 2019	Relation Between The Word of Shahmukhi Punjabi Language	Find the semantic relation among shahmukhi Punjabi Words	There Is Not Any Wordnet Use For Finding Relation
Gurpreet Singh And Tejinder Singh at 2012	Converation Between Script Of Gurumukhi Punjabi	Make transliteration system between Gurmukhi and Shahmukhi scripts of Punjabi language	The Translation System is only specific for Language
Andrea Lynn Bowden at 2012	Finding Punjabi Tonemics on the Gurumukhi Scripts	Review the Issue Of Lexical Tone Of Punjabi Language	Its Just A review On Punjabi Language
Farukh Arsalan and Dr.M.asim Mehmood at 2023	Finding The comprehensive morphological analysis of Shahmukhi Punjabi Nouns	Analyze the morphological patterns of nouns in Punjabi language	Its Just a Analysis No Proper Algorithum Or Models Were Applied
M.G Abbas Malik At 2005	Unicode Compatible Punjabi Dataset	Adaption Of Arabic Script For Shahmukhi Language Analyze letters, special symbols, diacritical marks	Its Just A Unicode Study Of Character Set
M.G Abbas at 2006	Apply Machine Translation On Punjabi Language	Converting a Shahmukhi Word Into Gurumukhi word Using PMT	The Translation System is only specific for Language

Punjabi	English	POS	SOP
Beant Singh And Gursharn Singh at 2017	Talk About Punjabi Language And Its Main Dialects	Study About Different Dialects Of Punjabi Language	No Practical Work Perform
Simmi Luthra And Parminder Singh at 2012	Make Punjabi Speech Generation System On Gurumukhi Punjabi Language	Making Speech System On The Basis Of Phoneme	The Work Based On Specific Gurumukhi Language
Hamza Khali And Ghulam Murtaza at 2023	Imptoving NER On Shahmukhi Punjabi Language	Using Bert And Data Augmentation To Improve NER	The Work Is not For Classification
Jasleen Kaur and Jatinder Kujmar at 2016	Punjabi Stop Words On Gurmukhi And Shahmukhi	Find Stop Word of 256 for Gurumukhi And 184 For Shahmukhi	Its Just For Data Preprocessing Not Practical Work Done On This Paper
Anne Murphy 2018	Writing Punjabi Across Border	Review And Study On Gurumukhi And Shahmukhi Punjabi Language	Its Just A Study Not Practical Work
Ejaz Hassan And Munawar Iqbal at 2015	Online Punjabi Lexical Resources	A web interface that facilitates the updation and query-based results retrieval of entries from lexical database.	There Is No NLP Model Use For This
Amina Tehseen Toqeer Ahsan at 2023	Neural POS Tagging Of Shahmukhi	This paper presents the development and evaluation of the first POS tagged corpus along with a Bi-directional long-short memory (BiLSTM) network based POS tagger	syntactic parsing, text summarization, text to speech systems and information extraction

4 Methodology

We begin our work by gathering an extensive dataset from numerous internet assets. Following corpus preparation and arithmetical study, we use cutting-edge CBoW, SG, and GloVe approaches to generate Punjabi word embeddings [15]. The generated word embeddings are tested for distributional semantic similarity using cosine similarity in the middle of the closest next one and word sets. In addition, we use t-SNE in conjunction with PCA to graphically analyze the distance between related words.

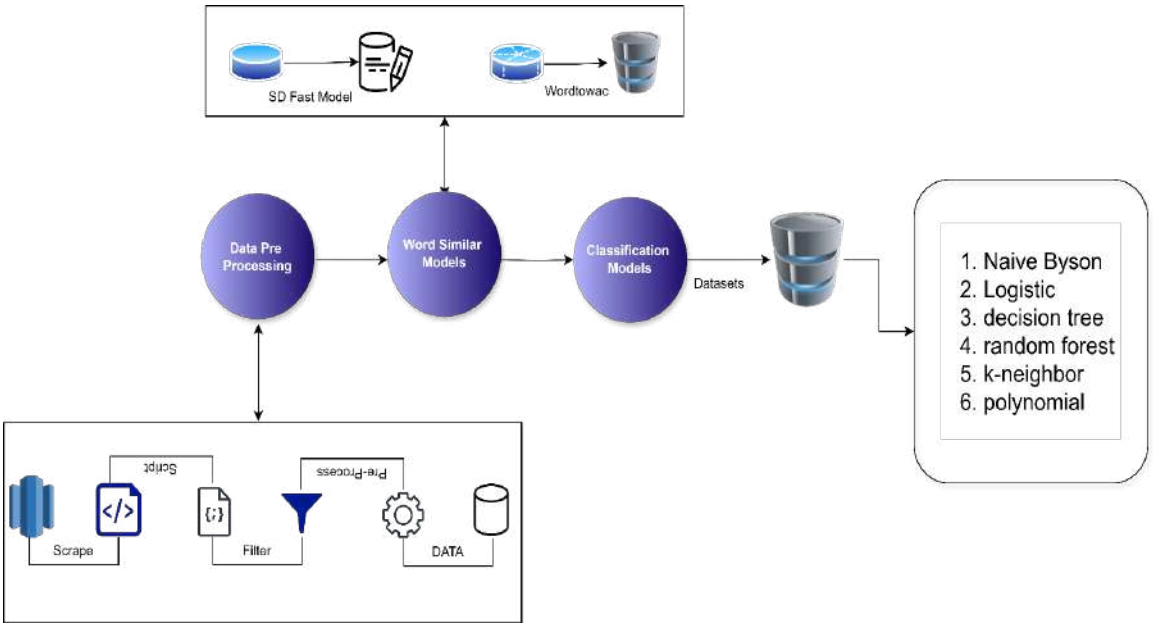

Figure 1. The methodology of all process

4.1 Dateset Acquirement:

The dataset gathers humanoid linguistic text prepared for a specific resolution. On the other hand, corpus arithmetical research delivers measurable, recyclable records and a chance to examine intuitions and conceptions about language. As a result, to examine the text, the corpus is necessary to study written language. Although there were few resources for learning Punjabi, we discovered a new website called Bhulekhatv Figure 3 that included several different categories, including sports and Islam.

4.2 Pre-processing:

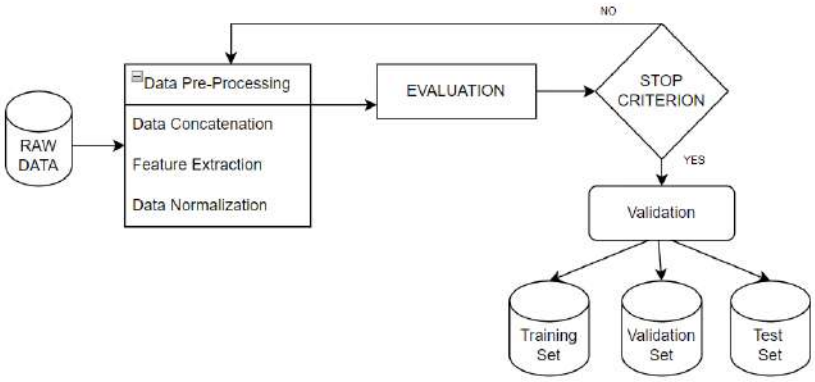

Figure 3. Data Pre-processing Architecture

Figure 2. The Scrapping Process

Pre-processing text corpora acquired from various websites is a tough job, and it turns even more difficult

when functioning on a low-resource linguistic like Punjabi since open-source pre-processing tools like NLTK [16] for English are unavailable. As a result, we build the pretreatment process depicted in Figure 1 to remove unsolicited information and dictionaries from other lingos, for example, English, to make involvement for word embedding [17]. On the other hand, the pre-processing methods required are explained in below diagram 1. In addition, we give a list of Shahmukhi language stop words, which take time and need humanoid decision

1. **Data Concatenation: :** Collected data from different Excel sheets were concatenated [18], as shown in Figure 3
2. **Removal of English:** The second thing was that the data had English words in it, so those words were removed.
3. **Removal of special characters and numbers:** The numerals and punctuation signals of an end dot like a full stop, hyphen, apostrophe, comma, quotation mark, and exclamation mark remained changed with white space for proper break in words since deprived of these signs [19], the words were discovered connected with their next or previous comparable terms.
4. **Normalization:**Tokenization separates text into words; the lowercase text is converted to guarantee consistency, and difficulties such as multiple whitespaces see Figure 3, English vocabulary, and duplicate words are dealt with. We also use stop-word removal; however, remember that stop-words are maintained for some tasks, such as GloVe preparation. Furthermore, in Skip-Gram (SG) models [21], sub-sampling techniques are used to eliminate common or stop words automatically, resulting in more effective word embeddings.
5. **Removing other noise:**Text collected from web resources contains much noise [20]. Therefore, we deleted all lingering punctuation marks, special characters, numeric entities, and even excess spaces.

4.3 Word embedding models.

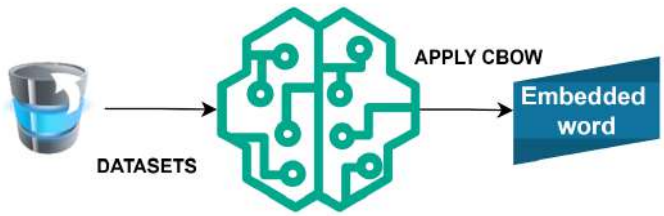

Figure 4. Cbow Architecture

Using powerful word embeddings derived from a massive unlabeled corpus, as shown in the Figure 4, NN-based NLP approaches have proven cutting-edge performance. As a result, word embeddings [22] have emerged as the primary section for establishing fresh standards in NLP utilizing deep learning methodologies. Newly, word embeddings have been utilized to generate language resources such as automated WordNet creation via an unsupervised technique and boost statistical NLP applications. Word embedding is clearly described as the encrypting of lexis V into N and the term w from V to vector w into N-dimensional embedding size [23]. They are widely classified as predictive and count-based methods, with Statistics of co-occurrence, NN methods, and probabilistic models used to construct them.

4.4 Skip Gram

By giving input words, the SG model [24] predicts surrounding words with the training goal of knowledge-effective word embeddings that guess adjacent arguments quickly [25]. Skip-gram seeks to get the most out of the middling log-probability of words w = w1,w2,.....wt throughout the whole train of the corpus.

$$J(\theta) = \frac{1}{T}\sum_{k=0}^{T}\sum_{\substack{-c\leq j\leq c \\ j\neq 0}} \log p(w_{t+j} \mid c_t) \tag{1}$$

Where ct Represents the meaning or category of words in (Equation 1) and wt denotes the collection of Next to wt arguments in (Equation 1) in the working out dataset [26].

4.5 Hyper Parameters

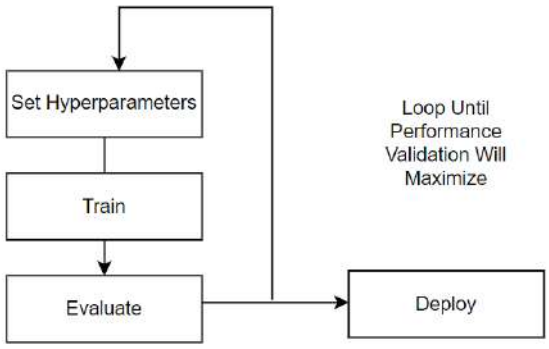

Figure 5. Architecture of Hyperparameter

4.5.1 Sub-Sampling

The subsampling strategy is useful for diluting the most frequent or stop words and accelerating and increasing accuracy when learning unusual Vectors of words. Many words in English, such as 'the,' 'you,' and 'that,' are not more important, yet they appear Several times in the text. However, treating all words similarly would result in model parameters being over-fitted on the most common word embeddings [27] and under-fitted on the rest. As a result, evaluating the disparity between rare and repeated phrases is advantageous. The subsampling technique removes the most frequently occurring words [28] in the corpus at random using some inception t, likelihood p of arguments, and occurrence f of words in (Equation 2).

$$p(w_i) = 1 - \frac{\sqrt{t}}{f(w_i)} \tag{2}$$

Where f(wi) is the occurrence of term wi and t > 0 are constraints in (Equation 2), and during the training phase, each wordwi is rejected with a computed probability.

4.5.2 Dynamic context menu

A fixed-size context window, such as a window size of 6, is widely used in word embedding models. This implies that, except for the first six tokens, the target word is processed in the same way as the neighbouring words inside the provided frame. This method prioritizes surrounding phrases that are more related to the meaning of the target word. This focus is aided by the weighing technique used by models such as CBoW [29], SG, and GloVe. The GloVe approach, for instance, uses a harmonic function to allocate weights to contexts. For example, a context word four tokens away from an occurrence is given a specified weight. SG implementations, on the other hand, assign equal significance to all contexts by dividing the window size by the distance from the target word. For

example, if the window size is set to 6, SG will consider the surroundings equally, ensuring a fair assessment of the situation.

4.5.3 Sub-word model

By sharing character representations across several words, the sub-word model is good at recognizing the underlying structure of words. As a result, the vector for each word is constructed by adding the vectors of its component character n-grams [30]. Take, for example, the word "table." We may extract sub-words such as ta, tab, tabl, table, table >, abl, able, able >, ble, ble >, le > by using a minimum length (minn) of 3 and a maximum length (max) of 6. Prefix and suffix words [31] are distinguished from other character sequences by symbols such as Morphological concepts are included in this sub-word model, improving the quality of representations for less common terms. The input word "w" is added to the collection of character representations in the (Equation 3), which aids in the learning process for each word.

$$(w, c) = X_{Z_{(kT)}} v_c \tag{3}$$

4.5.4 Weights based on position.

A positional weighting method reduces overfitting induced by encoding direct representations for words and their places. Contextual word representations use positional information to change the weighting of word embeddings. This method captures high-quality contextual representations at a low computational cost in (Equation 4).

$$(x + a)^n = \sum d_p \odot u_{(i+p)} \tag{4}$$

The context window point is represented by the symbol "p" in the provided formulation in (Equation 4), and it is coupled with the positional vector "dp." These positional vectors then reweight the context vector to produce the average of context words [32]. "P" represents in (Equation 4). the relative positional set in the context window, and "vC" represents the context vector of the word "wt." in Eq 4.

Using symbols and vectors distinguishes this mathematical statement to reflect the model's spatial and contextual properties in (Equation 4).

4.5.5 Mutual information has been shifted point-by-point

Using a word-context matrix with sparsely Shifted Positive Point-wise Mutual Information (SPPMI) for gaining word representations improves performance on two-word similarity evaluations. In both the Continuous Bag of Words (CBoW) and Skip-Gram (SG) models, the hyperparameter [33] "k," which represents the number of negatives, determines the value that both models strive to optimize for each word-context pair (w,c): PMI(w,c)logk. Similar to its previous purpose, the parameter "k" serves as a way to assess the likelihood of positive cases (the actual occurrence of w,c). It has a dual purpose in improving the estimate of negative cases.

4.5.6 Deleting rare words

The automatic deletion of rare words before building a context window improves efficiency in CBoW, SG, and GloVe models, increasing the actual size of context windows

4.6 Evaluation Techniques

The semantic similarity of word embeddings is crucial to the intrinsic assessment. Words are comparable in the word similarity measuring approach if they occur in similar situations. We use the dot product technique to assess the word similarity of the recommended Punjabi word embeddings.

4.6.1 similarity in cosine

The cosine similarity, a popular similarity measure between two non-zero vectors, uses the Euclidean dot product approach to calculate the cosine of the angle between them. To calculate the dot product, multiply the components of both vectors together. Notably, the dot product of two vectors yields a scalar value rather than another vector.

a = (a1,a2,a3,...,an) and only one value or scalar in (Equation 5). The dot product of two vectors is defined as follows: b = (b1,b2,b3,...,bn) where a and bn are vector components and n is the dimension of vectors in (Equation 5) such as:

$$\mathbf{a} \cdot \mathbf{b} = Xa_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n \tag{5}$$

On the other hand, the cosine of two non-zero vectors in (Equation 6) can be calculated using the Euclidean dot product formula.

$$a \cdot b = \|a\|\|b\| \cos(\theta) \tag{6}$$

Given ai and two attribute vectors a and b, the cosine similarity, cos(), is given by a dot product with magnitude as

$$\text{Similarity} = \cos(\theta) = \frac{a \cdot b}{\|a\|\|b\| \sqrt{\sum_{i=1}^{n} b_i^2}} \tag{7}$$

Where ai and bi are vector a and vector b components, respectively.

4.7 Classification of models

1. NLP is classified and available in the systematic approach of already defined labels or categories for numerical and textual based on inherent features. We can do category detection for titles, news descriptions and many more from sentiment analysis. The priority objective is to train or develop a model to accurately discern linguistic structure patterns to predict unseen text.
2. NLP classification goes beyond simple word classification; it enables machines to comprehend and interpret human language, creating opportunities for automation and augmentation across various industries. NLP categorization is at the forefront of technical breakthroughs, from interpreting social media sentiment to extracting vital information from medical texts. The development of NLP classification algorithms promises to be a significant step forward in the evolution of natural language understanding as the language continues to change. It will improve our capacity to engage with and learn from the immense ocean of textual data.
3. Classification for Sentiment analysis is identifying the emotional tone in the text. It identifies whether the sentiment is positive, negative, or neutral, like customer feedback. Spam Detection classify the messages for spam detection in the context of email or text messages. Filtering unwanted content from communication platforms. Topic modeling is another type in which the text and title are trained to model them, but it requires a large corpus for efficient analysis. NER classification can be classifying the names' location and organization. Documents classification and intent recognition are used for virtual assistants.

5 Results and Discussion

The occurrence of words in human language is not indiscriminately created but relatively follows precise predetermined rules that consent us to identify some language regularities. Zipf's law states that if the occurrence of words is presented in decreasing order, for example, Where Fr is the 4th rank letter frequency and a and b are input text arguments. In the Datasets, the relative letter occurrence is calculated by dividing the total number of

Table 3. Complete statistics on the corpus gathered from several sources

Source	Category	Paragraphs	Vocabulary	Unique Words
bhulekhatv	News	5167	885,862	553,411
bhulekhatv	Columns	2554	2,796,305	1,179,876
bhulekhatv	Ghazal	1847	139,666	112,552
bhulekhatv	Dohra	1789	140,430	112,000
Total		11,357	3,962,263	1,957,839

occurrences of a letter by the total number of letters. U can see in Figure 6. depicts the letter frequencies in our produced corpus; nonetheless, the corpus has a total of 14,566,396 characters.

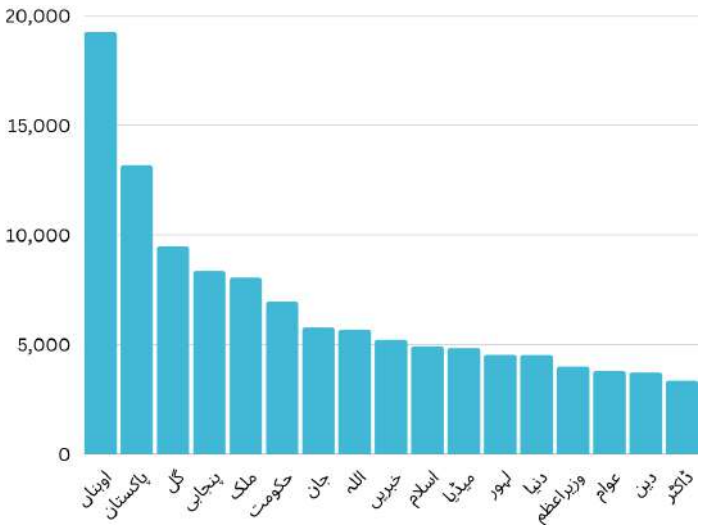

Figure 6. After filtering out stop words, the most frequent terms remain

N-grams are word combinations of letter occurrences, where every word is a gram. The occurrence of word n-grams is prudently examined in mandate to find the length of letters, which is essential to create Natural Language Programming systems, including word embedding knowledge, such as formative the most petite or supreme length of sub-word for character-level demonstration learning. The segment of words n-grams in words in the developed corpus is computed (Figure 3). Although 4-gram Letters are more common, bi-gram words are the most common and are mainly composed of stop words.

The word occurrence sum counts the number of times a letter appears in the text. The most frequently used words, such as "the" in English, are deemed to have a greater frequency. Likewise, the frequency of rarely used words is smaller. These occurrences can be estimated at the character or word level. We calculate word frequencies by counting the occurrences of a word w in the corpus c, as can be seen in (Equation 8), such as,

$$\text{freq}^{(w)} = (x + a)^n = \sum_{k=0}^{n} w_k \in c \tag{8}$$

In NLP, stop words are the most prevalent and least important terms. Removing such phrases may improve the NLP model's performance in sentiment analysis and text categorization tasks. However, creating such a word list takes time and involves user decisions. We first discovered Punjabi stop words by calculating their phrase frequencies using the equation above, and then we examined their grammatical status with the help of a Punjabi linguistic specialist. In our produced corpus, there are 340 recognized stop words. (Table 3) briefly lists the

most common Punjabi stop words and their frequency. The optimization of hyperparameters is more significant than the development of a novel algorithm. To optimize, we carefully select the vocabulary and algorithm-based constraints of the CBoW, SG, and GloVe algorithms. As a result, we conducted many training and evaluation studies until we arrived at the optimal set of hyperparameters shown in (Table 5) and discussed in Section 5.1. The optimum hyperparameters are chosen based on the high degree of cosine similarity.

Table 4. Stop Words and Their Frequencies

Stop Word	Frequency	Stop Word	Frequency	Stop Word	Frequency
دے	6224	تک	347	حے	162
جس	419	ہی	659	سبھ	38
ے	4663	جد	91	دیی	11
کرن	921	ہیں	22	کدی	53
دا	3302	ہویا	297	آپ	26
گیا	846	اسیں	511	کی	38
اے	5608	تاں	62	نالے	8
جی	199	کوئی	419	جاون	4
نوں	3685	بن	106	تسی	9
گی	486	ہونا	80	ایہو	36
ے	4659	لا	50	جدوں	100
کہ	3601	آئی	405	جاوے	82
جاندا	115	جو	123	چوں	4
توں	1880	بنائے	30	کولوں	39
نیں	1797	رہی	259	آکھ	13
دوران	186	کروائی	21	آوے	7
لی	1374	ورگا	8	ہووے	90
وی	1521	کرکے	61	بھاویں	4
ولوں	258	کئی	93	کاؤ	15
نال	2134	ویکھ	51	جھڑا	32
ہو	549	ویلے	213	طرحاں	33
سارا	22	جاں	34	بھی	8
کر	827	ایتھے	69	سدا	3
سی	1239	ہے	51	پاسوں	1
ہی	279	آ	63	جیکر	3
ہاں	489	رکھ	55	تیرا	3
سن	437	اوے	2	وانگ	3
میں	357	لگ	46	دیاں	1
ساں	54	لیا	148	کری	1
ایس	1100	اندر	88	وچ	84
بعد	314	ایہہ	329	پھیر	3
نہیں	1660	سن	7	تد	1
سکدے	92	لایا	33	کُجھ	2
مگر	84	چلا	27	جنہاں	1
اپنا	122	پر	62	میرا	50
بارے	189	پیا	14	چاہے	10
ہوئے	671	اگلی	6	جیہا	4
سارے	178	تینوں	7	چکے	6
رہے	571	ن	272	کے	971

Parameter adjustment is used to evaluate the state-of-the-art SG, SdFasttext word embedding algorithms for generating Punjabi word embeddings. These constraints can be classified as Vocabulary-based or algorithm-based. Incorporating character n-grams in knowledge letter demonstrations is an appropriate method, particularly for popular morphological languages, because it can compute unusual and misspelt words.

Table 5. Parameters use in Model

Parameter	Value
Epochs	40
Learning Rate	0.1
N-grams (minn)	3
N-grams (maxn)	6
Window Size (ws)	5
Negative Sampling (NS)	20

Punjabi is likewise a morphologically rich language. As a result, more robust embeddings could be trained using the hyperparameter optimization of the SG and SdFasttext methods. We independently tweaked and evaluated the hyperparameters of three algorithms, which are presented below:

1. Number of Epochs: see (Table 3). More epochs on the corpus generally offer better results, but they take more training time. As a result, we tested each word embedding model at 10, 20, 30, and 50 epochs, and results at 40 epochs were consistently good. Learning rate (lr): We experimented with lr values of 0.05, 0.1, and 0.25; in Table 3, the optimal lr (0.1) produces the best results for training all embedding models
2. may have a higher impact on accuracy. Lower embedding dimensions facilitate faster training and evaluation
3. Alphabet or Character n-grams: Choosing the minimum (min-n) and maximum (max-n) character n-gram lengths, seen in (Table 3), is an essential constraint for knowledge character-level representations of words in SG models. As a result, the 3 9 n-grams were analyzed to see how they affected embedding accuracy. We enhanced the measurement of the Alphabet. n-grams from minn = 3 to maxn = 6.
4. Window size (ws): A larger ws means examining more context words, whereas a smaller ws indicates limiting the quantity of context words, as seen in (Table 3). By varying the size of the dynamic context window, we tried ws of 3, 5, 7, and found that ws=5 consistently produces superior results.
5. Negative Sampling (NS): Longer negative instances produce better results but require longer training time. We tried 5, 10, 20, and 5 negative instances for SG. The top negative examples of 20 for SG considerably outperform in terms of average training time.

Discovering the nearest nearby words is critical in natural language processing and word embedding approaches for detecting semantically similar phrases within a given corpus. See (Table 4). This method uses the mathematical metric of cosine similarity to determine the proximity of words in a multidimensional space. Consider a word embedding model, such as Word2Vec or FastText, which converts words into high-dimensional vectors, as seen in (Table 4) with each dimension representing a semantic component. The angle between two-word vectors is then measured using cosine similarity, which reflects their similarity. When the cosine similarity is 1, the vectors are the same, and when it approaches -1, it indicates a significant dissimilarity. Finding terms with the maximum cosine similarity to a particular target word, showing semantic proximity, yields the closest nearby words. This method serves as the foundation for various natural language processing tasks, including word recommendation, sentiment analysis, and document classification, where it aids in discovering semantic links between words and improves comprehension of textual material.

Table 6: Word similarity using sdfasttext and word2vec

PUNJABI	SdfastText	Word2Vec
غیرقنونی	نجائز	اکاونٹس
	وزیر قانون	قنونی
	غیرآئیی	غیرآئیی
	دفعات	منافع
بنجر	زمیناں	سیراب
	سیراب	زمیناں
	ایکڑ	زمیں
	ویران	نہراں
مینوں	میرے	تہانوں
	اوہ	تینوں
	تہانوں	اوہنوں
	تسیں	اسنوں
اوتھے	جتھے	جتھے
	ای	ایتھے
	وی	اوتھوں
	جدوں	ے
لنگھیا	لنگھیاں	لنگھی
	ویلا	لنگھ
	گزریا	لنگھندا
	دھوڑ	لنگھندے

In general, nearby terms are seen to be more relevant to a word's meaning. The word embedding models can capture the lexical relationships between words, as in (Table 4). Identifying such word-to-word relationships is critical in NLP applications. We calculate the dot product of two vectors to determine the semantic link. The higher the cosine similarity score, the closer the words in the embedding matrix, whereas the lower The cosine similarity value, the more significant the gap between word pairs.

Table 7: Punjabi Word Similarity Scores

PUNJABI	SdfastText	Word2Vec
غیرقنونی	0.3474	0.5003
غیرآئیی		
بنجر	0.3776	0.6286
سیراب		
اوتھے	0.7167	0.7828
جتھے		
مینوں	0.4201	0.6504
تینوں		
لنگھیا	0.2718	0.5361
لنگھی		

Different strengths appear in (Table 7) when comparing different categorization models using measures like

recall, accuracy, and F1 score (Table 7). Naive Bayes is a dependable option for classification jobs since it exhibits a pleasing mix of precision and recall. Because of its ensemble approach, Random Forest works well at managing intricate interactions and reducing false positives and false negatives.

Table 8: Different Models F1, Precision and Recall With 4 Category

Class	Methods	F1	Precision	Recall
خبریں	Naïve Byson	0.99	0.98	0.98
	Logistic Regression	0.99	0.99	0.98
	Decision Tree	0.96	0.95	0.95
	Random Forest	0.98	0.98	0.98
	polynomial equations	0.93	0.89	0.86
	K-Nearest Neighbors	1	0.2	0.11
دوہڑا	Naïve Byson	0.91	0.91	0.92
	Logistic Regression	0.5	0.47	0.49
	Decision Tree	0.47	0.47	0.47
	Random Forest	0.51	0.48	0.46
	polynomial equations	0.48	0.48	0.48
	K-Nearest Neighbors	0.18	0.3	0.94
غزل	Naïve Byson	0.93	0.92	0.91
	Logistic Regression	0.93	0.92	0.91
	Decision Tree	0.49	0.5	0.5
	Random Forest	0.52	0.54	0.57
	polynomial equations	0.56	0.51	0.46
	K-Nearest Neighbors	0.52	0.11	0.06
کالمز	Naïve Byson	0.97	0.97	0.98
	Logistic Regression	0.95	0.97	0.98
	Decision Tree	0.92	0.92	0.91
	Random Forest	0.97	0.97	0.97
	polynomial equations	0.7	0.79	0.9
	K-Nearest Neighbors	0.97	0.72	0.57

Even though the technique is not traditional, Polynomial Regression shows flexibility in identifying non-linear correlations in the dataset. Reputable for its interpretability, Decision Tree minimizes mistakes while maintaining excellent accuracy. The simplest and most accurate method is K-Nearest Neighbors (KNN), whereas Logistic Regression is dependable and easy to understand. Distinct benefits, offering a varied toolbox for classification jobs based on specific dataset requirements and features.

This thorough assessment enhances knowledge of each categorization model's capabilities and offers insights into how well it performs individually. The various strengths and flexibility that are shown in different models provide a strong basis for choosing suitable approaches in the field of classification jobs, providing customized solutions for the complex requirements of various datasets and applications.

This collection of categorization models (Table 8) provides a broad range of tools to address various problems. Every model offers a useful answer, regardless of whether dependability, flexibility, interpretability, or simplicity are the top priorities. The thorough assessment based on corpus scores, accuracy, weighted average, and macro average guarantees a sophisticated comprehension of their suitability in various categorization scenarios. However, the naïve bayson is the best approach for classification because it has given multi-classification support on the complex and new corpus.

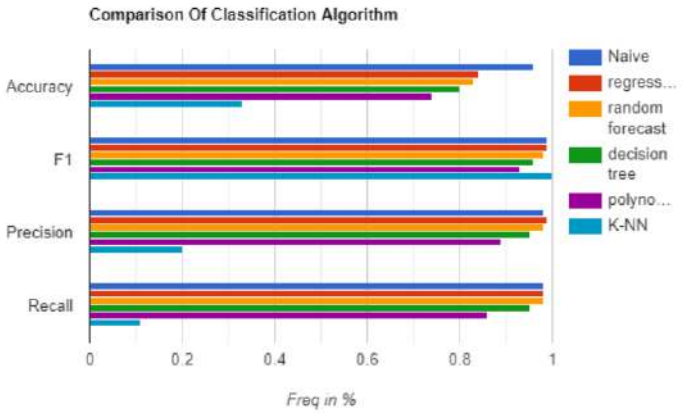

Figure 7. Graph Of Compare Algorithms

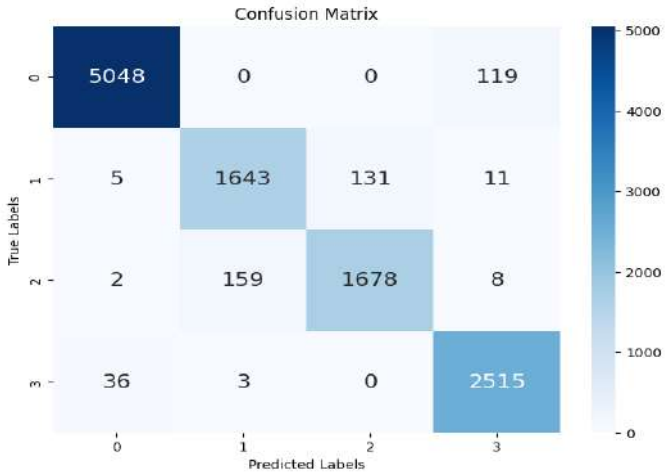

Figure 8. Confusion Matrix

Table 9. Accuracy and Corpus Report

Models	Accuracy	Macro Avg	Weighted Avg	Corpus
Naïve Bayes	0.96	0.95	0.96	11385
Logistic Regression	0.84	0.75	0.85	2272
Decision Tree	0.80	0.71	0.80	2272
Random Forest	0.83	0.74	0.83	2272
Polynomial Equations	0.74	0.68	0.74	9086
K-Nearest Neighbors	0.33	0.42	0.33	2272

The confusion matrix breaks out the model's predictions for every class in detail. This instance has four labels (0, 1, 2, 3). Label 0 is highlighted because it is considered the "best" label. The confusion matrix shows the model's accuracy in classifying cases for label 0 and differentiating between genuine positives, false positives, true negatives, and false negatives.

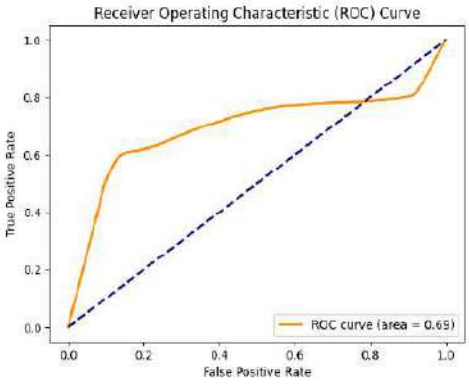

Figure 9. Graph Of precision

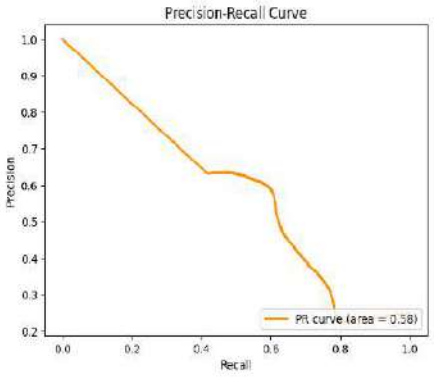

Figure 10. Graph Of recall

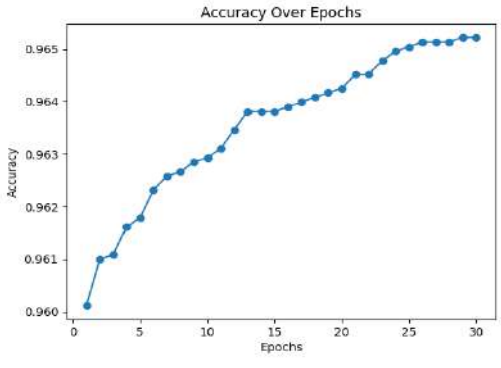

Figure 11. Accuracy Graph Over Epochs

The trade-off between genuine positive rate (sensitivity) and false positive rate (one-specificity) is depicted by the Receiver Operating Characteristic (ROC) curve. With an area under the ROC curve (AUC-ROC) of 0.69, this model may be used to differentiate between several classes. The discriminating power of the model is better the higher its AUC. The result of 0.69 indicates modest ability to discriminate. The accuracy-recall curve sheds light on the trade-off between precision and recall. The model demonstrates Moderate precision-recall performance, with an area under the curve (AUC-PR) of 0.58. A higher AUC-PR value indicates better accuracy and recall balance, whereas a lower value indicates difficulties in concurrently obtaining high precision and recal
The accuracy graph across epochs visually represents the model's performance during training. It illustrates how accuracy changes throughout the training phase using 30 epochs. Analyzing this graph might facilitate finding patterns, convergence, or variations in accuracy over time. Understanding the model's learning behavior and convergence rate may be gained by looking at the curve's form

6 Conclusion

In a linguistic field that has long been neglected, combining word embeddings and classification models for the Shahmukhi language offers a potential path for natural language processing. Using these methods, it has been possible to classify and analyze Shahmukhi's text effectively and gain insightful knowledge about its subtleties. However, it is essential to recognize this project's constraints and difficulties. Word embeddings have greatly improved our capacity to capture and interpret the semantic subtleties of Shahmukhi literature. Improved classification accuracy is made possible by the embedding, which serves as a link between machine learning models and language difficulties. By capturing context, connections, and meaning inside the language, the model can provide more sophisticated and contextually aware predictions thanks to this synergy. The categorization models have performed admirably, using methods like Random Forest, Naive Bayes, etc. They propose an organized framework for classifying text written in Shahmukhi, with uses ranging from subject categorization to sentiment analysis. It facilitates information retrieval and advances our knowledge of the linguistic and cultural background of Shahmukhi language datasets. Prospects for enhancing Shahmukhi language processing in the future are promising. A critical area of research that stands out is semantic analysis, which would move the emphasis from syntactic structures to a deeper comprehension of the meaning and intent of the text. By utilizing sophisticated approaches for semantic analysis, the models will be able to identify nuances, colloquialisms, and cultural allusions, improving the precision and applicability of classifications. Exploration into deep learning models provides an additional avenue due to their ability to learn complex patterns and representations. Neural network structures, such as transformers and recurrent neural networks (RNNs), may be able to decipher more complex linkages seen in the Shahmukhi language. These models can capture semantic subtleties and long-range relationships missed by more conventional machine-learning techniques.

7 Conflict of Interest

It is declare that all authors don't have any conflict of interest. It is also declare that this article does not contain any studies with human participants or animals performed by any of the authors. Furthermore, informed consent was obtained from all individual participants included in the study.

Author Contributions

Muhammad Shabbir: Conceptualization, Methodology, Software **Shafiq Ahmed Awan**: Data curation, Writing-Original draft preparation. **Anwar Ali Sathio** : Methodlogy, Visualization, Investigation. **Asadullah Burdi**: Supervision.: **Waheed Khan**: Software, Validation. **Waheed Khan & Anwar Ali Sathio** : Writing- Reviewing and Editing

Compliance with Ethical Standards

It is declared that all authors don't have any conflict of interest.

References

[1] E. Hasan, M. M. Iqbal, et al., "An online Punjabi Shahmukhi lexical resource," Sci Int Lahore, 2015.

[2] V. Goyal, G. S. Lehal, "Comparative study of Hindi and Punjabi language scripts," Nepal. Linguist., 2008.

[3] M. A. Hashmi, M. A. Mahmood, and M. I. Mahmood, "Analysis of lecxico-semantic relations of Punjabi Shahmukhi nouns: A corpus based study," Int. J. Engl. Linguist., vol. 13, 2019, Accessed: Jan. 23, 2024. [Online]. Available: https://pdfs.semanticscholar.org/ec1f/ea5d736f108bf89f465575f6de9845d08a24.pdf

[4] H. Singh, "Analyzing the Punjabi Language Stemmers: A Critical Approach.," in ISIC, 2021, pp. 250–258. Accessed: Jan. 23, 2024. [Online]. Available: https://ceur-ws.org/Vol-2786/Paper33.pdf

[5] M. F. Arslan, M. A. Mahmood, M. Shoaib, S. Idrees, and Z. Tariq, "Morphological Description Of Nouns In Shahmukhi Punjabi; A Corpus Based Study," J. Posit. Sch. Psychol., pp. 1259–1269, 2023.

[6] M. T. Ahmad et al., "Named Entity Recognition and Classification for Punjabi Shahmukhi," ACM Trans. Asian Low-Resour. Lang. Inf. Process., vol. 19, no. 4, pp. 1–13, Jul. 2020, doi: https://doi.org/10.1145/3383306,10.1145/3383306

[7] A. L. Bowden, Punjabi tonemics and the Gurmukhi script: A preliminary study. Brigham Young University, Accessed: Jan. 23, 2024.

[8] S. J. Johnson, M. R. Murty, and I. Navakanth, "A detailed review on word embedding techniques with emphasis on word2vec," Multimed. Tools Appl., Oct. 2023, doi: https://doi.org/10.1007/s11042-023-17007-z10.1007/s11042-023-17007-z

[9] S. Selva Birunda and R. Kanniga Devi, "A Review on Word Embedding Techniques for Text Classification," in Innovative Data Communication Technologies and Application, vol. 59, J. S. Raj, A. M. Iliyasu, R. Bestak, and Z. A. Baig, Eds., in Lecture Notes on Data Engineering and Communications Technologies, vol. 59. , Singapore: Springer Singapore, 2021, pp. 267–281.

[10] V. P. Singh and P. Kumar, "Sense disambiguation for Punjabi language using supervised machine learning techniques," Sādhanā, vol. 44, no. 11, p. 226, Nov. 2019, doi: https://doi.org/10.1007/s12046-019-1206-x10.1007/s12046-019-1206-x.

[11] Y. Li and T. Yang, "Word Embedding for Understanding Natural Language: A Survey," in Guide to Big Data Applications, vol. 26, S. Srinivasan, Ed., in Studies in Big Data, vol. 26. , Cham: Springer International Publishing, 2019, pp. 83–104.

[12] S. S. Sahu, D. Dutta, S. Pal, and I. Rasheed, "Effect of Stopwords and Stemming Techniques in Urdu IR," SN Comput. Sci., vol. 4, no. 5, p. 547, Jul. 2023, doi: 10.1007/s42979-023-01953-4.

[13] E. Loper and S. Bird, "NLTK: The Natural Language Toolkit." arXiv, May 17, 2002. Accessed: Jan. 23, 2024. [Online]. Available: http://arxiv.org/abs/cs/0205028

[14] S. Takase and S. Kobayashi, "All word embeddings from one embedding," Adv. Neural Inf. Process. Syst., vol. 33, pp. 3775–3785, 2020.

[15] S. S. Sahu and S. Pal, “Effect of stopwords in Indian language IR,” Sādhanā, vol. 47, no. 1, p. 17, Mar. 2022, doi: 10.1007/s12046-021-01731-z.

[16] H. Singh, “GPStemmer—A Gurmukhi Punjabi Stemmer,” in Advances in Data and Information Sciences, vol. 318, S. Tiwari, M. C. Trivedi, M. L. Kolhe, K. K. Mishra, and B. K. Singh, Eds., in Lecture Notes in Networks and Systems, vol. 318. , Singapore: Springer Singapore, 2022, pp. 493–503.

[17] D. Hadžiosmanović, L. Simionato, D. Bolzoni, E. Zambon, and S. Etalle, “N-Gram against the Machine: On the Feasibility of the N-Gram Network Analysis for Binary Protocols,” in Research in Attacks, Intrusions, and Defenses, vol. 7462, D. Balzarotti, S. J. Stolfo, and M. Cova, Eds., in Lecture Notes in Computer Science, vol. 7462. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2022, pp. 354–373.

[18] A. Sak, “Using cosine similarity classifier for NLP analysis in construction field texts,” in AIP Conference Proceedings, AIP Publishing, 2023. Accessed: Jan. 23, 2024. [Online]. Available: https://pubs.aip.org/aip/acp/article/2791/1/040010/2905402

[19] A. Bražinskas, S. Havrylov, and I. Titov, “Embedding Words as Distributions with a Bayesian Skip-gram Model.” arXiv, Jun. 10, 2018. Accessed: Jan. 23, 2024. [Online]. Available: http://arxiv.org/abs/1711.11027

[20] Z. Xiong, Q. Shen, Y. Xiong, Y. Wang, and W. Li, “New Generation Model of Word Vector Representation Based on CBOW or Skip-Gram,” Comput. Mater. Contin., vol. 60, no. 1, 2019. Accessed: Jan. 23, 2024.

[21] P. Preethi Krishna and A. Sharada, “Word Embeddings - Skip Gram Model,” V. K. Gunjan, V. Garcia Diaz, M. Cardona, V. K. Solanki, and K. V. N. Sunitha, Eds., Singapore: Springer Singapore, 2020, pp. 133–139.

[22] Y. Wang, L. Cui, and Y. Zhang, “Improving skip-gram embeddings using BERT,” vol. 29, pp. 1318–1328, 2021.

[23] P. Mehndiratta and D. Soni, “Identification of sarcasm using word embeddings and hyperparameters tuning,” *J. Discrete Math. Sci. Cryptogr.*, vol. 22, no. 4, pp. 465–489, May 2019, doi: 10.1080/09720529.2019.1637152.

[24] T. Menon, “Empirical analysis of CBOW and skip gram NLP models,” 2020, Accessed: Jan. 23, 2024.

[25] W. B. Cavnar and J. M. Trenkle, “N-gram-based text categorization,” in *Proceedings of SDAIR-94, 3rd annual symposium on document analysis and information retrieval*, Las Vegas, NV, 1994, p. 14. Accessed: Jan. 23, 2024.

[26] A. Basile, G. Dwyer, M. Medvedeva, J. Rawee, H. Haagsma, and M. Nissim, “N-GrAM: New Groningen Author-profiling Model.” arXiv, Jul. 12, 2019. Accessed: Jan. 23, 2024.

[27] T. P. Adewumi, F. Liwicki, and M. Liwicki, “Word2Vec: Optimal Hyper-Parameters and Their Impact on NLP Downstream Tasks.” arXiv, Apr. 17, 2021. Accessed: Jan. 23, 2024.

[28] S. Pattnaik and A. K. Nayak, “Summarization of odia text document using cosine similarity and clustering,” in *2019 International Conference on Applied Machine Learning (ICAML)*, IEEE, 2019, pp. 143–146. Accessed: Jan. 23, 2024.

[29] W. Chen et al., “A comparative study of logistic model tree, random forest, and classification and regression tree models for spatial prediction of landslide susceptibility,” *Catena*, vol. 151, pp. 147–160, 2019.

[30] K. Park, J. S. Hong, and W. Kim, “A Methodology Combining Cosine Similarity with Classifier for Text Classification,” *Appl. Artif. Intell.*, vol. 34, no. 5, pp. 396–411, Apr. 2020, doi: 10.1080/08839514.2020.1723868.

[31] W. Loh, “Fifty Years of Classification and Regression Trees,” *Int. Stat. Rev.*, vol. 82, no. 3, pp. 329–348, Dec. 2014, doi: 10.1111/insr.12019.

[32] D. Alberg, M. Last, and A. Kandel, “Knowledge discovery in data streams with regression tree methods,” *WIREs Data Min. Knowl. Discov.*, vol. 2, no. 1, pp. 69–78, Jan. 2022, doi: 10.1002/widm.51.

[33] V. Gregg, “Word frequency, recognition and recall.,” 1976, Accessed: Jan. 23, 2024.