
WORD EMBEDDING BASED NEW CORPUS FOR LOW-RESOURCED LANGUAGE: SINDHI

A PREPRINT

Wazir Ali, Jay Kumar, Junyu Lu, Zenglin Xu

School of Computer Science and Engineering
University of Electronic Science and Technology of China

January 20, 2022

ABSTRACT

Representing words and phrases into dense vectors of real numbers which encode semantic and syntactic properties is a vital constituent in natural language processing (NLP). The success of neural network (NN) models in NLP largely rely on such dense word representations learned on the large unlabeled corpus. Sindhi is one of the rich morphological language, spoken by large population in Pakistan and India lacks corpora which plays an essential role of a test-bed for generating word embeddings and developing language independent NLP systems. In this paper, a large corpus of more than 61 million words is developed for low-resourced Sindhi language for training neural word embeddings. The corpus is acquired from multiple web-resources using web-scrapy. Due to the unavailability of open source preprocessing tools for Sindhi, the prepossessing of such large corpus becomes a challenging problem specially cleaning of noisy data extracted from web resources. Therefore, a preprocessing pipeline is employed for the filtration of noisy text. Afterwards, the cleaned vocabulary is utilized for training Sindhi word embeddings with state-of-the-art GloVe, Skip-Gram (SG), and Continuous Bag of Words (CBoW) word2vec algorithms. The intrinsic evaluation approach of cosine similarity matrix and WordSim-353 are employed for the evaluation of generated Sindhi word embeddings. Moreover, we compare the proposed word embeddings with recently revealed Sindhi fastText (SdfastText) word representations. Our intrinsic evaluation results demonstrate the high quality of our generated Sindhi word embeddings using SG, CBoW, and GloVe as compare to SdfastText word representations.

Keywords Corpus acquisition · Sindhi language · Neural networks · Word embeddings · Continuous Bag of Words · Skip gram · GloVe

1 Introduction

Sindhi is a rich morphological, multitscript, and multidialectal language. It belongs to the Indo-Aryan language family [1], with significant cultural and historical background. Presently, it is recognized as is an official language [2] in Sindh province of Pakistan, also being taught as a compulsory subject in Schools and colleges. Sindhi is also recognized as one of the national languages in India. Ulhasnagar, Rajasthan, Gujarat, and Maharashtra are the largest Indian regions of Sindhi native speakers. It is also spoken in other countries except for Pakistan and India, where native Sindhi speakers have migrated, such as America, Canada, Hong Kong, British, Singapore, Tanzania, Philippines, Kenya, Uganda, and South, and East Africa. Sindhi has rich morphological structure [3] due to a large number of homogeneous words. Historically, it was written in multiple writing systems, which differ from each other in terms of orthography and morphology. The Persian-Arabic is the standard script of Sindhi, which was officially accepted in 1852 by the British government¹. However, the Sindhi-Devanagari is also a popular writing system in India being written in left to right direction like the Hindi language. Formerly, Khudabadi, Gujrati, Landa, Khojki, and Gurumukhi were also adopted as its writing systems. Even though, Sindhi has great historical and literal background, presently spoken by nearly

¹<https://www.britannica.com/topic/Sindhi-language>

75 million people [2]. The research on SNLP was coined in 2002², however, IT grabbed research attention after the development of its Unicode system [4]. But still, Sindhi stands among the low-resourced languages due to the scarcity of core language processing resources of the raw and annotated corpus, which can be utilized for training robust word embeddings or the use of machine learning algorithms. Since the development of annotated datasets requires time and human resources.

The Language Resources (LRs) are fundamental elements for the development of high quality NLP systems based on automatic or NN based approaches. The LRs include written or spoken corpora, lexicons, and annotated corpora for specific computational purposes. The development of such resources has received great research interest for the digitization of human languages [5]. Many world languages are rich in such language processing resources integrated in their software tools including English [6] [7], Chinese [8] and other languages [9] [10]. The Sindhi language lacks the basic computational resources [11] of a large text corpus, which can be utilized for training robust word embeddings and developing language independent NLP applications including semantic analysis, sentiment analysis, parts of the speech tagging, named entity recognition, machine translation [12], multitasking [13], [14]. Presently Sindhi Persian-Arabic is frequently used for online communication, newspapers, public institutions in Pakistan, and India [2]. But little work has been carried out for the development of LRs such as raw corpus [15], [16], annotated corpus [17], [18], [2], [19]. In the best of our knowledge, Sindhi lacks the large unlabelled corpus which can be utilized for generating and evaluating word embeddings for Statistical Sindhi Language Processing (SSL)

One way to break out this loop is to learn word embeddings from unlabelled corpora, which can be utilized to bootstrap other downstream NLP tasks. The word embedding is a new term of semantic vector space [20], distributed representations [21], and distributed semantic models. It is a language modeling approach [22] used for the mapping of words and phrases into n -dimensional dense vectors of real numbers that effectively capture the semantic and syntactic relationship with neighboring words in a geometric way [23] [24]. Such as “Einstein” and “Scientist” would have greater similarity compared with “Einstein” and “doctor.” In this way, word embeddings accomplish the important linguistic concept of “a word is characterized by the company it keeps”. More recently NN based models yield state-of-the-art performance in multiple NLP tasks [25] [26] with the word embeddings. One of the advantages of such techniques is they use unsupervised approaches for learning representations and do not require annotated corpus which is rare for low-resourced Sindhi language. Such representations can be trained on large unannotated corpora, and then generated representations can be used in the NLP tasks which uses a small amount of labelled data.

In this paper, we address the problems of corpus construction by collecting a large corpus of more than 61 million words from multiple web resources using the web-scrappy framework. After the collection of the corpus, we carefully preprocessed for the filtration of noisy text, e.g., the HTML tags and vocabulary of the English language. The statistical analysis is also presented for the letter, word frequencies and identification of stop-words. Finally, the corpus is utilized to generate Sindhi word embeddings using state-of-the-art GloVe [27] SG and CBoW [28] [21] [25] algorithms. The popular intrinsic evaluation method [21] [29] [30] of calculating cosine similarity between word vectors and WordSim353 [31] are employed to measure the performance of the learned Sindhi word embeddings. We translated English WordSim353³ word pairs into Sindhi using bilingual English to Sindhi dictionary. The intrinsic approach typically involves a pre-selected set of query terms [24] and semantically related target words, which we refer to as query words. Furthermore, we also compare the proposed word embeddings with recently revealed Sindhi fastText (SdfastText)⁴ [26] word representations. To the best of our knowledge, this is the first comprehensive work on the development of large corpus and generating word embeddings along with systematic evaluation for low-resourced Sindhi Persian-Arabic. The synopsis of our novel contributions is listed as follows:

- We present a large corpus of more than 61 million words obtained from multiple web resources and reveal a list of Sindhi stop words.
- We develop a text cleaning pipeline for the preprocessing of the raw corpus.
- Generate word embeddings using GloVe, CBoW, and SG Word2Vec algorithms also evaluate and compare them using the intrinsic evaluation approaches of cosine similarity matrix and WordSim353.
- We are the first to evaluate SdfastText word representations and compare them with our proposed Sindhi word embeddings.

The remaining sections of the paper are organized as, Section 2 presents the literature survey regarding computational resources, Sindhi corpus construction, and word embedding models. Afterwards, Section 3 presents the employed

²“Sindhia lai Kampyutar jo Istemalu” (Use of computer for Sindhi), an article published in Sindhu yearly, Ulhasnagar. 2002

³ Available online at <https://rdrr.io/cran/wordspace/man/WordSim353.html>

⁴We denote Sindhi word representations as (SdfastText) recently revealed by fastText, available at (<https://fasttext.cc/docs/en/crawl-vectors.html>) trained on Common Crawl and Wikipedia corpus of Sindhi Persian-Arabic.

methodology, Section 4 consist of statistical analysis of the developed corpus. Section 5 present the experimental setup. The intrinsic evaluation results along with comparison are given in Section 6. The discussion and future work are given in Section 7, and lastly, Section 8 presents the conclusion.

2 Related work

The natural language resources refer to a set of language data and descriptions [32] in machine readable form, used for building, improving, and evaluating NLP algorithms or softwares. Such resources include written or spoken corpora, lexicons, and annotated corpora for specific computational purposes. Many world languages are rich in such language processing resources integrated in the software tools including NLTK for English [6], Stanford CoreNLP [7], LTP for Chinese [8], TectoMT for German, Russian, Arabic [9] and multilingual toolkit [10]. But Sindhi language is at an early stage for the development of such resources and software tools.

The corpus construction for NLP mainly involves important steps of acquisition, preprocessing, and tokenization. Initially, [15] discussed the morphological structure and challenges concerned with the corpus development along with orthographical and morphological features in the Persian-Arabic script. The raw and annotated corpus [2] for Sindhi Persian-Arabic is a good supplement towards the development of resources, including raw and annotated datasets for parts of speech tagging, morphological analysis, transliteration between Sindhi Persian-Arabic and Sindhi-Devanagari, and machine translation system. But the corpus is acquired only from Wikipedia-dumps. A survey-based study [5] provides all the progress made in the Sindhi Natural Language Processing (SNLP) with the complete gist of adopted techniques, developed tools and available resources which show that work on resource development on Sindhi needs more sophisticated efforts. **The raw corpus is utilized for Sindhi word segmentation [33]**. More recently, an initiative towards the development of resources is taken [17] by open sourcing annotated dataset of Sindhi Persian-Arabic obtained from news and social blogs. The existing and proposed work is presented in Table 1 on the corpus development, word segmentation, and word embeddings, respectively.

The power of word embeddings in NLP was empirically estimated by proposing a neural language model [22] and multitask learning [13], but recently usage of word embeddings in deep neural algorithms has become integral element [34] for performance acceleration in deep NLP applications. The CBoW and SG [28] [21] popular word2vec neural architectures yielded high quality vector representations in lower computational cost with integration of character-level learning on large corpora in terms of semantic and syntactic word similarity later extended [34] [25]. Both approaches produce state-of-the-art accuracy with fast training performance, better representations of less frequent words and efficient representation of phrases as well. [35] proposed NN based approach for generating morphemic-level word embeddings, which surpassed all the existing embedding models in intrinsic evaluation. A count-based GloVe model [27] also yielded state-of-the-art results in an intrinsic evaluation and downstream NLP tasks.

The performance of Word embeddings is evaluated using intrinsic [24] [30] and extrinsic evaluation [29] methods. The performance of word embeddings can be measured with intrinsic and extrinsic evaluation approaches. The intrinsic approach is used to measure the internal quality of word embeddings such as querying nearest neighboring words and calculating the semantic or syntactic similarity between similar word pairs. A method of direct comparison for intrinsic evaluation of word embeddings measures the neighborhood of a query word in vector space. The key advantage of that method is to reduce bias and create insight to find data-driven relevance judgment. An extrinsic evaluation approach is used to evaluate the performance in downstream NLP tasks, such as parts-of-speech tagging or named-entity recognition [24], but the Sindhi language lacks annotated corpus for such type of evaluation. Moreover, extrinsic evaluation is time consuming and difficult to interpret. Therefore, we **opt** intrinsic evaluation method [29] to get a quick insight into the quality of proposed Sindhi word embeddings by measuring the cosine distance between similar words and using WordSim353 dataset. A study reveals that the choice of optimized hyper-parameters [36] has a great impact on the quality of pretrained word embeddings as compare to desing a novel algorithm. Therefore, we optimized the hyperparameters for generating robust Sindhi word embeddings using CBoW, SG and GloVe models. The embedding visualization is also useful to visualize the similarity of word clusters. Therefore, we use t-SNE [37] dimensionality reduction algorithm for compressing high dimensional embedding into 2-dimensional x,y coordinate pairs with PCA [38]. The PCA is useful to combine input features by dropping the least important features while retaining the most valuable features.

3 Methodology

This section presents the employed methodology in detail for corpus acquisition, preprocessing, statistical analysis, and generating Sindhi word embeddings.

Paper	Related works	Resource
[26]	Word embedding	Wiki-dumps (2016)
[15]	Text Corpus	4.1M tokens
[2]	Corpus development	Wiki-dumps (2016)
[17]	Labelled corpus	6.8K
[18]	Sentiment analysis	31.5K tokens
[33]	Text Segmentation	1575K
Proposed work	Raw Corpus	61.39 M tokens
	Word embeddings	61.39M tokens

Table 1: Comparison of existing and proposed work on Sindhi corpus construction and word embeddings

3.1 Task description

We initiate this work from scratch by collecting large corpus from multiple web resources. After preprocessing and statistical analysis of the corpus, we generate Sindhi word embeddings with state-of-the-art CBOW, SG, and GloVe algorithms. The generated word embeddings are evaluated using the intrinsic evaluation approaches of cosine similarity between nearest neighbors, word pairs, and WordSim-353 for distributional semantic similarity. Moreover, we use t-SNE with PCA for the comparison of the distance between similar words via visualization.

3.2 Corpus acquisition

The corpus is a collection of human language text [32] built with a specific purpose. However, the statistical analysis of the corpus provides quantitative, reusable data, and an opportunity to examine intuitions and ideas about language. Therefore, the corpus has great importance for the study of written language to examine the text. In fact, realizing the necessity of large text corpus for Sindhi, we started this research by collecting raw corpus from multiple web resource using web-scrappy framework⁵ for extraction of news columns of daily Kawish⁶ and Awami Awaz⁷ Sindhi newspapers, Wikipedia dumps⁸, short stories and sports news from Wichaar⁹ social blog, news from Focus Word press blog¹⁰, historical writings, novels, stories, books from Sindh Salamat¹¹ literary websites, novels, history and religious books from Sindhi Adabi Board¹² and tweets regarding news and sports are collected from twitter¹³.

3.3 Preprocessing

The preprocessing of text corpus obtained from multiple web resources is a challenging task specially it becomes more complicated when working on low-resourced language like Sindhi due to the lack of open-source preprocessing tools such as NLTK [6] for English. Therefore, we design a preprocessing pipeline depicted in Figure 1 for the filtration of unwanted data and vocabulary of other languages such as English to prepare input for word embeddings. Whereas, the involved preprocessing steps are described in detail below the Figure 1. Moreover, we reveal the list of Sindhi stop words [39] which is labor intensive and requires human judgment as well. Hence, the most frequent and least important words are classified as stop words with the help of a Sindhi linguistic expert. The partial list of Sindhi stop words is given in 4. We use python programming language for designing the preprocessing pipeline using regex and string functions.

- **Input:** The collected text documents were concatenated for the input in UTF-8 format.
- **Replacement symbols:** The punctuation marks of a full stop, hyphen, apostrophe, comma, quotation, and exclamation marks replaced with white space for authentic tokenization because without replacing these symbols with white space the words were found joined with their next or previous corresponding words.
- **Filtration of noisy data:** The text acquisition from web resources contain a huge amount of noisy data. Therefore, we filtered out unimportant data such as the rest of the punctuation marks, special characters, HTML tags, all types of numeric entities, email, and web addresses.

⁵<https://github.com/scrapy/scrapy>

⁶<http://kawish.asia/Articles1/index.htm>

⁷<http://www.awamiaawaz.com/articles/294/>

⁸<https://dumps.wikimedia.org/sdwiki/20180620/>

⁹<http://wichaar.com/news/134/>, accessed in Dec-2018

¹⁰<https://thefocus.wordpress.com/> accessed in Dec-2018

¹¹<http://sindhosalamat.com/>, accessed in Jan-2019

¹²http://www.sindhiadabiboard.org/catalogue/History/Main_History.HTML

¹³<https://twitter.com/dailysindhtimes>

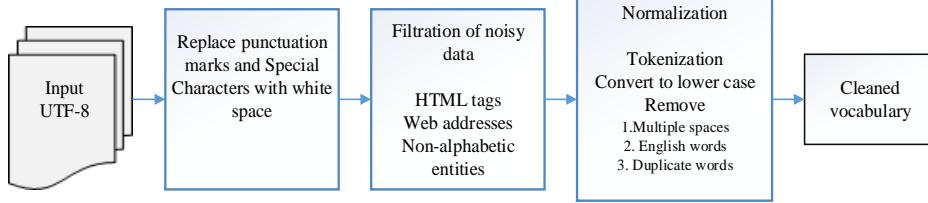


Figure 1: Employed preprocessing pipeline for text cleaning

- **Normalization:** In this step, We tokenize the corpus then normalize to lower-case for the filtration of multiple white spaces, English vocabulary, and duplicate words. The stop words were only filtered out for preparing input for GloVe. However, the sub-sampling approach in CBoW and SG can discard most frequent or stop words automatically.

3.4 Word embedding models

The NN based approaches have produced state-of-the-art performance in NLP with the usage of robust word embeddings generated from the large unlabelled corpus. Therefore, word embeddings have become the main component for setting up new benchmarks in NLP using deep learning approaches. Most recently, the use cases of word embeddings are not only limited to boost statistical NLP applications but can also be used to develop language resources such as automatic construction of WordNet [40] using the unsupervised approach.

The word embedding can be precisely defined as the encoding of vocabulary V into N and the word w from V to vector \vec{w} into N -dimensional embedding space. They can be broadly categorized into predictive and count based methods, being generated by employing co-occurrence statistics, NN algorithms, and probabilistic models. The GloVe [27] algorithm treats each word as a single entity in the corpus and generates a vector of each word. However, CBoW and SG [28] [21], later extended [34] [25], well-known as word2vec rely on simple two layered NN architecture which uses linear activation function in hidden layer and softmax in the output layer. The word2vec model treats each word as a bag-of-character n-gram.

3.5 GloVe

The GloVe is a log-bilinear regression model [27] which combines two methods of local context window and global matrix factorization for training word embeddings of a given vocabulary in an unsupervised way. It weights the contexts using the harmonic function, for example, a context word four tokens away from an occurrence will be counted as $\frac{1}{4}$. The Glove's implementation represents word $w \in V_w$ and context $c \in V_c$ in D -dimensional vectors \vec{w} and \vec{c} in a following way,

$$\vec{w} \cdot \vec{c} + b_w + b_c = \log(\#(w, c)) \forall (w, c) \in D \quad (1)$$

Where b_w and b_c represent word and context biases also to be learned as parameters of \vec{w} and \vec{c} . The objective of GloVe algorithm is to learn word embeddings by taking the log-count matrix [36] shifted by the bias terms of entire vocabulary as,

$$M^{\log(\#(w, c))} \approx \mathbf{W} \cdot \mathbf{C}^T + \mathbf{b}^{\vec{w}} + \mathbf{b}^{\vec{c}} \quad (2)$$

where, $\mathbf{b}^{\vec{w}}$ is row vector $|V_w|$ and $\mathbf{b}^{\vec{c}}$ is $|V_c|$ is column vector.

3.6 Continuous bag-of-words

The standard CBoW is the inverse of SG [28] model, which predicts input word on behalf of the context. The length of input in the CBoW model depends on the setting of context window size which determines the distance to the left and right of the target word. Hence the context is a window that contain neighboring words such as by giving $w = \{w_1, w_2, \dots, w_t\}$ a sequence of words T , the objective of the CBoW is to maximize the probability of given neighboring words such as,

$$\sum_{t=0}^T \log p(w_t | c_t) \quad (3)$$

Where c_t is context of t^{th} word for example with window $w_{t-c}, \dots, w_{t-1}, w_{t+1} \dots w_{t+c}$ of size $2c$.

3.7 Skip gram

The SG model predicts surrounding words by giving input word [21] with training objective of learning good word embeddings that efficiently predict the neighboring words. The goal of skip-gram is to maximize average log-probability of words $w = \{w_1, w_2, \dots, w_t\}$ across the entire training corpus,

$$J(\theta) \frac{1}{T} \sum_{t=0}^T \left(\sum_{-c \leq j \leq c, j} \log p(w_{t+j} | c_t) \right) \quad (4)$$

Where c_t denotes the context of words indices set of nearby w_t words in the training corpus.

3.8 Hyperparameters

3.8.1 Sub-sampling

The sub-sampling [21] approach is useful to dilute most frequent or stop words, also accelerates learning rate, and increases accuracy for learning rare word vectors. Numerous words in English, e.g., ‘the’, ‘you’, ‘that’ do not have more importance, but these words appear very frequently in the text. However, considering all the words equally would also lead to over-fitting problem of model parameters [25] on the frequent word embeddings and under-fitting on the rest. Therefore, it is useful to count the imbalance between rare and repeated words. The sub-sampling technique randomly removes most frequent words with some threshold t and probability p of words and frequency f of words in the corpus.

$$P(w_i) = 1 - \sqrt{\frac{t}{f(w_i)}} \quad (5)$$

Where each word w_i is discarded with computed probability in training phase, $f(w_i)$ is frequency of word w_i and $t > 0$ are parameters.

3.8.2 Dynamic context window

The traditional word embedding models usually use a fixed size of a context window. For instance, if the window size $ws=6$, then the target word apart from 6 tokens will be treated similarly as the next word. The scheme is used to assign more weight to closer words, as closer words are generally considered to be more important to the meaning of the target word. The CBoW, SG and GloVe models employ this weighting scheme. The GloVe model weights the contexts using a harmonic function, for example, a context word four tokens away from an occurrence will be counted as $\frac{1}{4}$. However, CBoW and SG implementation equally consider the contexts by dividing the ws with the distance from target word, e.g. $ws=6$ will weigh its context by $\frac{6}{6} \frac{5}{6} \frac{4}{6} \frac{3}{6} \frac{2}{6} \frac{1}{6}$

3.8.3 Sub-word model

The sub-word model [25] can learn the internal structure of words by sharing the character representations across words. In that way, the vector for each word is made of the sum of those character $n - gram$. Such as, a vector of a word “table” is a sum of $n - gram$ vectors by setting the letter $n - gram$ size $min = 3$ to $max = 6$ as, $<ta, tab, tabl, table, table>, abl, able, able>, ble, ble>, le>$, we can get all sub-words of “table” with minimum length of $minn = 3$ and maximum length of $maxn = 6$. The $<$ and $>$ symbols are used to separate prefix and suffix words from other character sequences. In this way, the sub-word model utilizes the principles of morphology, which improves the quality of infrequent word representations. In addition to character $n - grams$, the input word w is also included in the set of character $n - gram$, to learn the representation of each word. We obtain scoring function using a input dictionary of $n - grams$ with size K by giving word w , where $K_w \subset \{1, \dots, K\}$. A word representation Z_k is associated to each $n - gram$ Z . Hence, each word is represented by the sum of character $n - gram$ representations, where, s is the scoring function in the following equation,

$$s(w, c) = \sum_{k \in K_j} z_k^T v_c \quad (6)$$

3.8.4 Position-dependent weights

The position-dependent weighting approach [41] is used to avoid direct encoding of representations for words and their positions which can lead to over-fitting problem. The approach learns positional representations in contextual word representations and used to reweight word embedding. Thus, it captures good contextual representations at lower computational cost,

$$v_C V = \sum_{p \in P} d_p \odot u_{t+p} \quad (7)$$

Where p is individual position in context window associated with d_p vector. Afterwards the context vector reweighted by their positional vectors is average of context words. The relative positional set is P in context window and v_C is context vector of w_t respectively.

3.8.5 Shifted point-wise mutual information

The use sparse Shifted Positive Point-wise Mutual Information (SPPMI) [42] word-context matrix in learning word representations improves results on two word similarity tasks. The CBoW and SG have k (number of negatives) [28] [21] hyperparameter, which affects the value that both models try to optimize for each $(w, c) : PMI(w, c) - \log k$. Parameter k has two functions of better estimation of negative examples, and it performs as before observing the probability of positive examples (actual occurrence of w, c).

3.8.6 Deleting rare words

Before creating a context window, the automatic deletion of rare words also leads to performance gain in CBoW, SG and GloVe models, which further increases the actual size of context windows.

3.9 Evaluation methods

The intrinsic evaluation is based on semantic similarity [24] in word embeddings. The word similarity measure approach states [36] that the words are similar if they appear in the similar context. We measure word similarity of proposed Sindhi word embeddings using dot product method and WordSim353.

3.9.1 Cosine similarity

The cosine similarity between two non-zero vectors is a popular measure that calculates the cosine of the angle between them which can be derived by using the Euclidean dot product method. The dot product is a multiplication of each component from both vectors added together. The result of a dot product between two vectors isn't another vector but a single value or a scalar. The dot product for two vectors can be defined as: $\vec{a} = (a_1, a_2, a_3, \dots, a_n)$ and $\vec{b} = (b_1, b_2, b_3, \dots, b_n)$ where a_n and b_n are the components of the vector and n is dimension of vectors such as,

$$\vec{a} \cdot \vec{b} = \sum_{i=1}^n a_i b_i = a_1 b_1 + a_2 b_2 + \dots + a_n b_n \quad (8)$$

However, the cosine of two non-zero vectors can be derived by using the Euclidean dot product formula,

$$\vec{a} \cdot \vec{b} = \|\vec{a}\| \|\vec{b}\| \cos(\theta) \quad (9)$$

Given a_i two vectors of attributes a and b , the cosine similarity, $\cos(\theta)$, is represented using a dot product and magnitude as,

$$\text{similarity} = \cos(\theta) = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \|\vec{b}\|} = \frac{\sum_{i=1}^n a_i b_i}{\sqrt{\sum_{i=1}^n a_i^2} \sqrt{\sum_{i=1}^n b_i^2}} \quad (10)$$

where a_i and b_i are components of vector \vec{a} and \vec{b} , respectively.

3.9.2 WordSim353

The WordSim353 [43] is popular for the evaluation of lexical similarity and relatedness. The similarity score is assigned with 13 to 16 human subjects with semantic relations [31] for 353 English noun pairs. Due to the lack of annotated datasets in the Sindhi language, we translated WordSim353 using English to Sindhi bilingual dictionary¹⁴ for the evaluation of our proposed Sindhi word embeddings and SdfastText. We use the Spearman correlation coefficient for the semantic and syntactic similarity comparison which is used to discover the strength of linear or nonlinear relationships if there are no repeated data values. A perfect Spearman's correlation of +1 or -1 discovers the strength of a link between two sets of data (word-pairs) when observations are monotonically increasing or decreasing functions of each other in a following way:

$$r_s = 1 - \frac{6 \sum_i d_i^2}{n(n^2 - 1)} \quad (11)$$

where r_s is the rank correlation coefficient, n denote the number of observations, and d_i is the rank difference between i^{th} observations.

4 Statistical analysis of corpus

The large corpus acquired from multiple resources is rich in vocabulary. We present the complete statistics of collected corpus (see Table 2 with number of sentences, words and unique tokens.

¹⁴<http://dic.sindhila.edu.pk/index.php?txtsrch=>

Source	Category	Sentences	Vocabulary	Unique words
Kawish	News columns	473,225	13,733,379	109,366
Awami awaz	News columns	107,326	7,487,319	65,632
Wikipedia	Miscellaneous	844,221	8,229,541	245,621
Social Blogs	Stories, sports	7,018	254,327	10,615
	History, News	3,260	110,718	7,779
Focus word press	Short Stories	63,251	968,639	28,341
	Novels	36,859	998,690	18,607
	Safarnama	138,119	2,837,595	53,193
Sindh Salamat	History	145,845	3,493,020	61,993
	Religion	96,837	2,187,563	39,525
	Columns	85,995	1,877,813	33,127
	Miscellaneous	719,956	9,304,006	168,009
Sindhi Adabi Board	History books	478,424	9,757,844	57,854
	Twitter	10,752	159,130	9,794
Total		3,211,088	61,399,584	908,456

Table 2: Complete statistics of collected corpus from multiple resources

n-grams	Frequency	% in corpus
Uni-gram	936,301	1.52889
Bi-gram	19,187,314	31.3311
Tri-gram	11,924,760	19.472
4-gram	14,334,444	23.4068
5-gram	9,459,657	15.4467
6-gram	3,347,907	5.4668
7-gram	1,481,810	2.4196
8-gram	373,417	0.6097
9-gram	163,301	0.2666
10-gram	21,287	0.0347
11-gram	5,892	0.0096
12-gram	3,033	0.0049
13-gram	1,036	0.0016
14-gram	295	0.0004
Total	61,240,454	100

Table 3: Length of letter n-grams in words, distinct words, frequency and percentage in corpus

4.1 Letter occurrences

The frequency of letter occurrences in human language is not arbitrarily organized but follow some specific rules which enable us to describe some linguistic regularities. The Zipf's law [44] suggests that if the frequency of letter or word occurrence ranked in descending order such as,

$$F_r = \frac{a}{r^b} \quad (12)$$

Where, F_r is the letter frequency of r^{th} rank, a and b are parameters of input text. The comparative letter frequency in the corpus is the total number of occurrences of a letter divided by the total number of letters present in the corpus. The letter frequencies in our developed corpus are depicted in Figure 2; however, the corpus contains 187,620,276 total number of the character set. Sindhi Persian-Arabic alphabet consists of 52 letters but in the vocabulary 59 letters are detected, additional seven letters are modified uni-grams and standalone honorific symbols.

4.2 Letter n-grams frequency

We denote the combination of letter occurrences in a word as n-grams, where each letter is a gram in a word. The letter n-gram frequency is carefully analyzed in order to find the length of words which is essential to develop NLP systems, including learning of word embeddings such as choosing the minimum or maximum length of sub-word for character-level representation learning [25]. We calculate the letter n-grams in words along with their percentage in the developed corpus (see Table 3). The bi-gram words are most frequent, mostly consists of stop words and secondly, 4-gram words have a higher frequency.

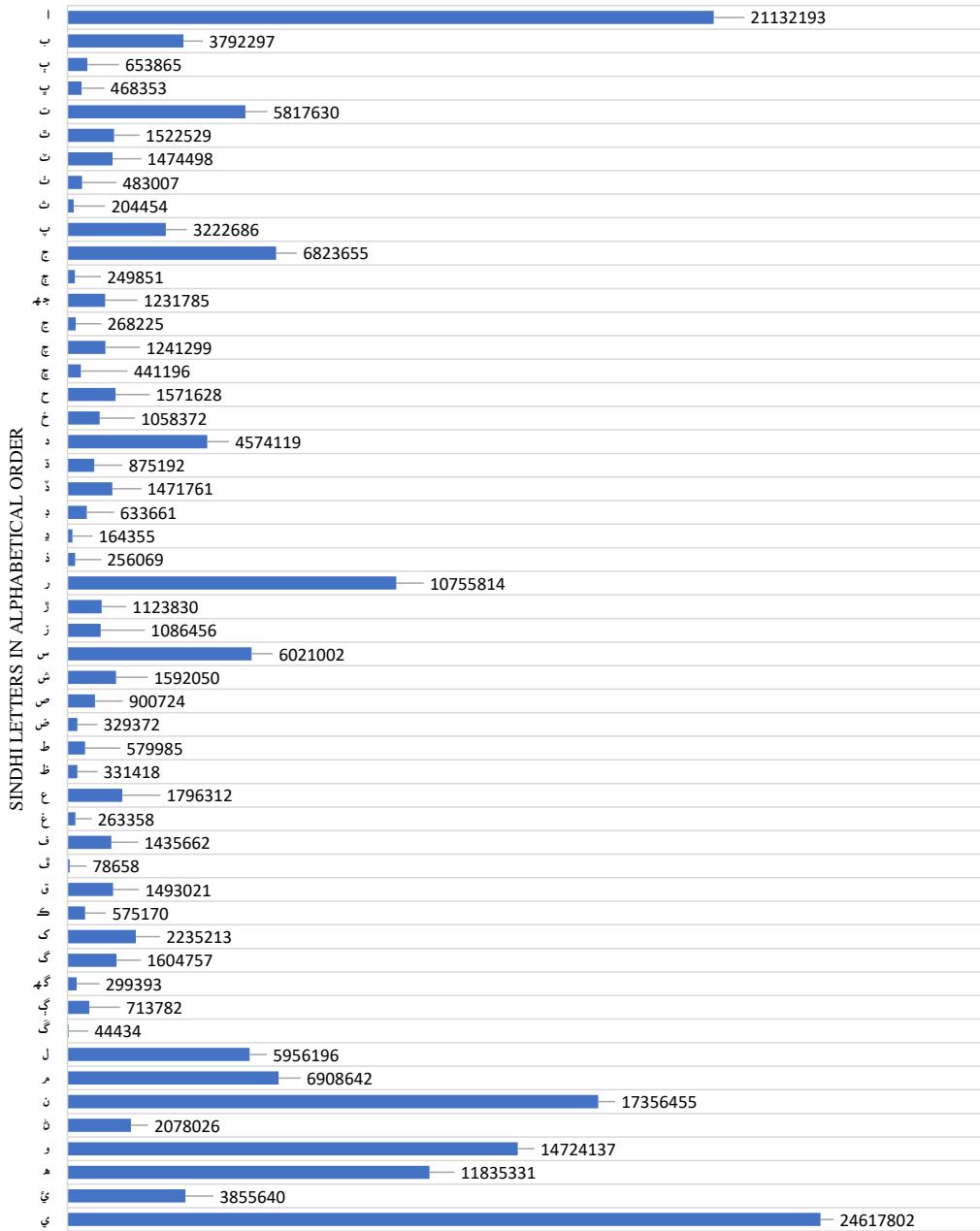


Figure 2: Frequency distribution of letter occurrences

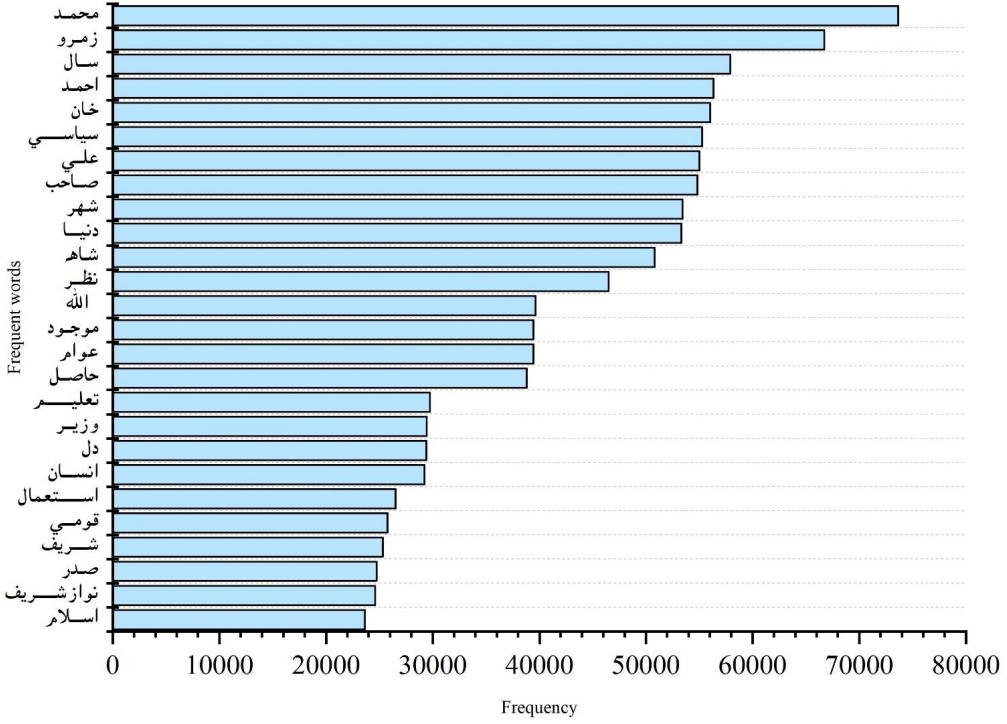


Figure 3: Most frequent words after filtration of stop words.

4.3 Word Frequencies

The word frequency count is an observation of word occurrences in the text. The commonly used words are considered to be with higher frequency, such as the word “the” in English. Similarly, the frequency of rarely used words to be lower. Such frequencies can be calculated at character or word-level. We calculate word frequencies by counting a word w occurrence in the corpus c , such as,

$$\text{freq}(w) = \sum_{k=0}^k w_k \in c \quad (13)$$

Where the frequency of w is the sum of every occurrence k of w in c .

4.4 Stop words

The most frequent and least important words in NLP are often classified as stop words. The removal of such words can boost the performance of the NLP model [39], such as sentiment analysis and text classification. But the construction of such words list is time consuming and requires user decisions. Firstly, we determined Sindhi stop words by counting their term frequencies using Eq. 13, and secondly, by analysing their grammatical status with the help of Sindhi linguistic expert because all the frequent words are not stop words (see Figure 3). After determining the importance of such words with the help of human judgment, we placed them in the list of stop words. The total number of detected stop words is 340 in our developed corpus. The partial list of most frequent Sindhi stop words is depicted in Table 4 along with their frequency. The filtration of stop words is an essential preprocessing step for learning GloVe [27] word embeddings; therefore, we filtered out stop words for preparing input for the GloVe model. However, the sub-sampling approach [34] [25] is used to discard such most frequent words in CBoW and SG models.

5 Experiments and results

Hyperparameter optimization [24] is more important than designing a novel algorithm. We carefully choose to optimize the dictionary and algorithm-based parameters of CBoW, SG and GloVe algorithms. Hence, we conducted a large number of experiments for training and evaluation until the optimization of most suitable hyperparameters depicted in Table 5 and discussed in Section 5.1. The choice of optimized hyperparameters is based on The high cosine similarity

Word	Freq:	Word	Freq:	Word	Freq:	Word	Freq:	Word	Freq:
جي	2068791	سندن	50219	دوران	21351	الف	10070	تهنجو	4838
جو	980764	۽	49856	ويندا	20829	باب	9962	آ	4820
ٿه	895430	ويندو	49060	وغيره	20214	بنا	9834	اور	4741
تي	528745	رهيا	46609	پوءِ	20132	رهن	9593	هيں	4680
ٻه	538301	هوندو	45822	جاری	192667	ايندي	9154	بنجي	4533
سان	409252	منهنجي	43204	هلي	19200	اتان	9052	هليا	4516
ان	397751	ها	42837	ورتو	18996	بن	8947	اول	4493
ڏه	397393	بي	42508	لفظ	18026	هليو	8890	هلندو	4092
هو	393941	هنن	38121	۾	17784	ترجمو	8787	مليا	4064
جا	291307	توهان	37991	مطلوب	16773	ون	8222	هيل	4516
هن	283280	بعد	37032	هجن	16602	رهندا	8101	عنوان	4820
مان	231711	هوندي	36961	اسين	16308	نهایت	7966	اور	4741
اهو	192158	ثالو	35774	مثال	15405	هوس	7838	هيں	4680
جهنن	183937	صفحو	35711	ويون	15278	ورتي	7617	بنجي	4533
ويو	165138	اتي	34695	ايندو	14814	نالا	7556	آءِ	4518
انهن	156051	تمار	33882	خود	14702	پهراحال	7546	تاين	4496
اسان	151600	وارون	33303	ايترو	13837	علاوه	7523	وج	4426
يا	150984	نالي	32736	جيابان	13797	ناهن	7310	تن	4419
سندس	129866	هتي	32449	آنمن	13514	آيو	7218	ڪجن	4407
واري	120907	تهنن	32343	بس	12823	هيا	7206	اهي	4389
مون	108985	تان	31957	باوجود	12656	هتان	7398	نيند	4378
اهما	101799	سو	31562	ثابت	12620	رهندي	6550	آيا	4357
اهي	100484	بات	30835	تو	12601	وجهي	6526	هوند	4342
هي	87537	يعني	30146	چو	12592	جيڪر	6493	ملڻ	4331
هئي	82867	ويندي	28525	معني	12408	هونديون	6453	هئں	4318
جن	80650	اوهان	27497	رهيون	10941	احوال	6346	سواء	4302
رهيو	77724	وارا	27030	عنوان	10889	لاءِ	6029	آند	4206
هر	74937	هيون	26425	باري	10875	واريون	5457	تنهن	4193
وري	62225	صرف	25442	مليو	10844	ملن	5321	ڪري	4165
رهي	60474	سي	25271	جلد	10791	ملندو	5320	آيس	4138
ٿه	59968	تون	24903	منهنجي	10790	جتان	5318	جول	4123
ويا	58308	هيyo	24871	وي	10339	وارين	5292	اٿهوند	4058
هجي	51691	اي	24656	ويهي	10337	حوالو	5182	آبل	4037
وارو	51501	فقط	22149	ايندا	10167	هيڪر	5039	ره	4013

Table 4: Partial list of most frequent Sindhi stop words along with frequency in the developed corpus.

Parameter	CBoW, SG	GloVe
Epoch	40	40
lr	0.25	0.25
D	300	300
minn char	02	—
maxn char	07	—
ws	7	7
NS	20	—
minw	4	4

Table 5: Optimized parameters for CBoW, SG and GloVe models

score in retrieving nearest neighboring words, the semantic, syntactic similarity between word pairs, WordSim353, and visualization of the distance between twenty nearest neighbours using t-SNE respectively. All the experiments are conducted on GTX 1080-TITAN GPU.

5.1 Hyperparameter optimization

The state-of-the-art SG, CBoW [28] [34] [21] [25] and Glove [27] word embedding algorithms are evaluated by parameter tuning for development of Sindhi word embeddings. These parameters can be categories into dictionary and algorithm based, respectively. The integration of character n-gram in learning word representations is an ideal method especially for rich morphological languages because this approach has the ability to compute rare and misspelled words. Sindhi is also a rich morphological language. Therefore more robust embeddings became possible to train with the hyperparameter optimization of SG, CBoW and GloVe algorithms. We tuned and evaluated the hyperparameters of three algorithms individually which are discussed as follows:

- **Number of Epochs:** Generally, more epochs on the corpus often produce better results but more epochs take long training time. Therefore, we evaluate 10, 20, 30 and 40 epochs for each word embedding model, and 40 epochs constantly produce good results.
- **Learning rate (lr):** We tried lr of 0.05, 0.1, and 0.25, the optimal lr (0.25) gives the better results for training all the embedding models.
- **Dimensions (D):** We evaluate and compare the quality of $100 - D$, $200 - D$, and $300 - D$ using WordSim353 on different ws , and the optimal $300 - D$ are evaluated with cosine similarity matrix for querying nearest neighboring words and calculating the similarity between word pairs. The embedding dimensions have little affect on the quality of the intrinsic evaluation process. However, the selection of embedding dimensions might have more impact on the accuracy in certain downstream NLP applications. The lower embedding dimensions are faster to train and evaluate.
- **Character n-grams:** The selection of minimum (minn) and the maximum (maxn) length of character $n - grams$ is an important parameter for learning character-level representations of words in CBoW and SG models. Therefore, the n-grams from 3 – 9 were tested to analyse the impact on the accuracy of embedding. We optimized the length of character n-grams from $minn = 2$ and $maxn = 7$ by keeping in view the word frequencies depicted in Table 3.
- **Window size (ws):** The large ws means considering more context words and similarly less ws means to limit the size of context words. By changing the size of the dynamic context window, we tried the ws of 3, 5, 7 the optimal $ws=7$ yield consistently better performance.
- **Negative Sampling (NS):** : The more negative examples yield better results, but more negatives take long training time. We tried 10, 20, and 30 negative examples for CBoW and SG. The best negative examples of 20 for CBoW and SG significantly yield better performance in average training time.
- **Minimum word count (minw):** We evaluated the range of minimum word counts from 1 to 8 and analyzed that the size of input vocabulary is decreasing at a large scale by ignoring more words similarly the vocabulary size was increasing by considering rare words. Therefore, by ignoring words with a frequency of less than 4 in CBoW, SG, and GloVe consistently yields better results with the vocabulary of 200,000 words.
- **Loss function (ls):** we use hierarchical softmax (hs) for CBoW, negative sampling (ns) for SG and default loss function for GloVe [27].
- The recommended verbosity level, number of buckets, sampling threshold, number of threads are used for training CBoW, SG [25], and GloVe [27].

6 Word similarity comparison of Word Embeddings

6.1 Nearest neighboring words

The cosine similarity matrix [36] is a popular approach to compute the relationship between all embedding dimensions of their distinct relevance to query word. The words with similar context get high cosine similarity and geometrical relatedness to Euclidean distance, which is a common and primary method to measure the distance between a set of words and nearest neighbors. Each word contains the most similar top eight nearest neighboring words determined by the highest cosine similarity score using Eq. 10. We present the English translation of both query and retrieved words also discuss with their English meaning for ease of relevance judgment between the query and retrieved words. To take a closer look at the semantic and syntactic relationship captured in the proposed word embeddings, Table 6 shows the top eight nearest neighboring words of five different query words **Friday**, **Spring**, **Cricket**, **Red**, **Scientist** taken from the vocabulary. As the first query word *Friday* returns the names of days *Saturday*, *Sunday*, *Monday*, *Tuesday*, *Wednesday*, *Thursday* in an unordered sequence. The SdfastText returns five names of days *Sunday*, *Thursday*, *Monday*, *Tuesday* and *Wednesday* respectively. The GloVe model also returns five names of days. However, CBoW and SG gave six names of days except *Wednesday* along with different writing forms of query word *Friday* being written in the Sindhi language which shows that CBoW and SG return more relevant words as compare to SdfastText and GloVe. The CBoW returned *Add* and GloVe returns *Honorary* words which are little similar to the query word but SdfastText resulted two irrelevant words *Kameeso (N)* which is a name (N) of person in Sindhi and *Phrase* is a combination of three Sindhi words which are not tokenized properly. Similarly, nearest neighbors of second query word *Spring* are retrieved accurately as names and seasons and semantically related to query word *Spring* by CBoW, SG and GloVe but SdfastText returned four irrelevant words of *Dilbahar (N)*, *Pharase*, *Ashbahar (N)* and *Farzana (N)* out of eight. The third query word is *Cricket*, the name of a popular game. The first retrieved word in CBoW is *Kabadi (N)* that is a popular national game in Pakistan. Including *Kabadi (N)* all the returned words by CBoW, SG and GloVe are related to *Cricket* game or names of other games. But the first word in SdfastText contains a punctuation mark in retrieved word *Gone.Cricket* that are two words joined with a punctuation mark (.), which shows the tokenization error in preprocessing step, sixth retrieved word *Misspelled* is a combination of three words not related to query word, and *Played*, *Being played* are also irrelevant and stop words. Moreover, fourth query word *Red* gave results that contain names of closely related to query word and different forms of query word written in the Sindhi language. The last returned word *Unknown* by SdfastText is irrelevant and not found in the Sindhi dictionary for translation. The last query word *Scientist* also contains semantically related words by CBoW, SG, and GloVe, but the first *Urdu word* given by SdfastText belongs to the Urdu language which means that the vocabulary may also contain words of other languages. Another *unknown* word returned by SdfastText does not have any meaning in the Sindhi dictionary. More interesting observations in the presented results are the diacritized words retrieved from our proposed word embeddings and The authentic tokenization in the preprocessing step presented in Figure 1. However, SdfastText has returned tri-gram words of *Phrase* in query words **Friday**, **Spring**, a *Misspelled* word in **Cricket** and **Scientist** query words. Hence, the overall performance of our proposed SG, CBoW, and GloVe demonstrate high semantic relatedness in retrieving the top eight nearest neighbor words.

6.2 Word pair relationship

Generally, closer words are considered more important to a word's meaning. The word embeddings models have the ability to capture the lexical relations between words. Identifying such relationship that connects words is important in NLP applications. We measure that semantic relationship by calculating the dot product of two vectors using Eq. 10. The high cosine similarity score denotes the closer words in the embedding matrix, while less cosine similarity score means the higher distance between word pairs. We present the cosine similarity score of different semantically or syntactically related word pairs taken from the vocabulary in Table 7 along with English translation, which shows the average similarity of 0.632, 0.650, 0.591 yields by CBoW, SG and GloVe respectively. The SG model achieved a high average similarity score of **0.650** followed by CBoW with a 0.632 average similarity score. The GloVe also achieved a considerable average score of 0.591 respectively. However, the average similarity score of SdfastText is 0.388 and the word pair **Microsoft-Bill Gates** is not available in the vocabulary of SdfastText. This shows that along with performance, the vocabulary in SdfastText is also limited as compared to our proposed word embeddings.

Moreover, the average semantic relatedness similarity score between countries and their capitals is shown in Table 8 with English translation, where SG also yields the best average score of **0.663** followed by CBoW with 0.611 similarity score. The GloVe also yields better semantic relatedness of 0.576 and the SdfastText yield an average score of 0.391. The first query word **China-Beijing** is not available in the vocabulary of SdfastText. However, the similarity score between **Afghanistan-Kabul** is lower in our proposed CBoW, SG, GloVe models because the word *Kabul* is the name of the capital of *Afghanistan* as well as it frequently appears as an adjective in Sindhi text which means *able*.

Query	SdfastText	Eng. Trans.	CBoW	Eng. Trans.	SG	Eng. Trans.	GloVe	Eng. Trans
Friday	جمو	آجْر	Sunday	سومر	Monday	جمون	Friday	جمرات
	بروز	On the day	جمعون	Friday	جمعي	Friday	اربع	Wednesday
	خميسو	Kameeso (N)	آچْر	Sunday	جعما	Fridays	شیخ	Saturday
	خمیس	Thursday	اگارو	Tuesday	خمیس	Thursday	بروز	On the day
	سومر	Monday	جماعی	On Friday	سومر	Monday	ماناتی	Honorary
	اثارو	Tuesday	خمیس	Thursday	چنچو	Saturday	خمیس	Thursday
	اربع	Wednesday	چنچ	Saturday	آچْر	Sunday	جمعي	Friday
	هي . ابڑو آهي	Phrase	آچْر	Sunday	اگارو	Tuesday	جمون	Friday
بهار	بهارجو	of spring	بهار	Springs	بهاران	Springs	بهاري	Comfort
	بهار	spring	خران	Autumn	بهار	Springs	سدا	Ever
	دلبهار	Dilbahar (N)	پُربهار	Mid-autumn	خران	Autumn	خوبصورتیہ	Beauty
	بهارن	Springs	سُرهان	Fragrance	تَنْدِيز	Bloom	بهار	Spring
	جومشہور گلو	Phrase	بھارن	On Springs	پُربهار	mid spring	بھارن	Springs
	اشبهار	Ashbahar (N)	سدابهار	Ever spring	سیارو	Winter	خشبو	Fragrance
	بودلو	Bodlo (N)	سیارو	winter	اونھارو	summer	خران	Autumn
	فرزان	Farzana (N)	اوئھارو	summer	بھار	spring	پُربهار	Mid spring
کرکیت	کرکیت ویو	Gone.cricket	ڪڪبی	Kabadi (N)	ڪڪپېرن	Cricketers	ڪڪپېر	Cricketer
	کرکیترز	Cricketers	توُرٽامینت	Tournament	ڪڪپېر	Cricketers	توئنتیي	Twenty
	ڪڪپېرن	Cricketers	ڪڪپېرن	Cricketers	ھاڪی	Hockey	گرانوئندن	Grounds
	ڪڪپېر	Cricketer	مئج	Match	توئنتیي	Twenty	ڪڪپېرن	Cricketers
	توئنتی ۲۰	T-Twenty	راندیگر	Players	راند	Game	تیست	Test
	عڪسلونڪرڪ	Misspelled	راند	Game	مئج	Match	مئج	Match
	کیٽپر	Played	بشت	Bat	فڪسنج	Fixing	توئنتی	Twenty
	کیدل	Being played	ھاڪی	Hockey	بال	Bat	هارائی	Lost
Red	ڳاڙهو	Reddish	ڳاڙھي	Red	ڳاڙھوي	Reddish	لاتين	Red lamp
	ڳاڙهه	Red	ڳاڙھوي	Reddish	ڳاڙھي	Red	ڳاڙهائش	Light red
	ڳاڙهه	Red	اچو	White	ڳاڙهه	Red	ھيپرو	Yellowish
	ڳاڙھسو	Reddish	پيلو	Yellowish	ڳاڙھيرڙو	Reddish	ڳاڙھوي	Reddish
	ڳاڙهائش	Light red	ھيپرو	Yellowish	ڦڪر	Yellow	ڳاڙھسو	Reddish
	ڳاڙھين	Red's	ڳاڙهه	Red	ڳاڙھسو	Reddish	ڳاڙهه	Red
	ڳاڙهاڻ	Light red	ڳاڙھسري	Reddish	ڳاڙهاڻ	Light red	ڳاڙهائش	Light red
	هڙھو	Unknown	ڳاڙھسو	Reddish	ڳاڙهائش	Light red	ڳاڙھي	Red
Scientist	سائنسدان	Urdu word	سائنسدان	Scientists	ڪيميادان	Chemist	سوشل	Social
	سائنسدان	Scientists	مفڪر	Thinker	سائنسدان	Scientists	سائنسدان	Scientist
	سائنسن	Sciences	فلاسافر	philosopher	آئنسٽائين	Einstein	سٽڪالا جست	Psychologist
	سائنسدان	Misspelled	اینگریمینبر	Anaximander	سائنتست	Scientist	ڪائڻه	Kaira (N)
	سائنسدان	Misspelled	ڪيميادان	Chemist	استائن	Stein	گيلوانى	Gailwani
	نيڪوت	Unknown	ماهر	Expert	سائنسدان	Scientist	فلاسافر	Philosopher
	سائنسي	Scientific	سائنس	Science	آئنسٽائين	Einstein	مفڪر	Thinker

Table 6: Eight nearest neighboring words of each query word with English translation

Word pair	English Translation	SdfastText	CBoW	SG	GloVe
شـاگرـد-استـاد	Teacher-Student	0.306	0.635	0.558	0.633
سـائـنسـدان-آـئـنـسـائـن	Einstein-Scientist	0.432	0.610	0.673	0.621
ڪـرـسيـ-تـيـبـلـ	Table-chair	0.284	0.520	0.539	0.492
ڪـلـابـ-ڪـلـ	Flower-Rose	0.347	0.796	0.638	0.588
ڇـوـڪـريـ-عـورـتـ	Woman-Girl	0.264	0.601	0.573	0.543
ڏـاـڏـيـ-ڏـاـڏـوـ	Grandfather-Grandmother	0.486	0.787	0.800	0.691
ڇـوـڪـروـ-مـرـدـ	Man-boy	0.223	0.451	0.511	0.472
ڪـرـاـچـيـ-سـنـدـ	Sindh-Karachi	0.472	0.567	0.669	0.647
لاـهـورـ-پـنجـابـ	Panjab-Lahore	0.386	0.528	0.569	0.513
چـينـ-چـينـيـ	China-Chinese	0.508	0.654	0.746	0.614
آـمـريـكاـ-آـمـريـكـيـ	America-American	0.566	0.804	0.875	0.687
ماـنـڪـروـسـافـتـ-بـلـيـگـيـسـ	Microsoft-Bill Gates	NA	0.527	0.502	0.483
Average		0.388	0.632	0.650	0.591

Table 7: Word pair relationship using cosine similarity (higher is better)

Word pair	English Translation	SdfastText	CBoWs	SG	GloVe
چـائـنـ-بيـجـنـگـ	China-Beijing	N.A	0.594	0.743	0.542
آـمـريـكاـ-نيـويـارـكـ	America-New York	0.371	0.635	0.689	0.518
جاـپـانـ-توـكـيـوـ	Japan-Tokyo	0.451	0.610	0.643	0.806
انـديـاـ-مبـكـيـ	India-Mumbai	0.266	0.651	0.759	0.628
بنـگـلـادـيشـ-داـكاـ	Bangladesh-Dhaka	0.428	0.629	0.633	0.593
اـيرـانـ-تـهـرانـ	Iran-Tehran	0.431	0.673	0.769	0.561
اـفـغـانـسـ坦ـ-قـابـلـ	Afghanistan-Kabul	0.103	0.267	0.283	0.215
عـراـقـ-بغـدادـ	Iraq-Baghdad	0.450	0.695	0.712	0.542
سـعـودـيـ-ريـاضـ	Saudi-Riyadh	0.454	0.576	0.686	0.616
ملـائـيشـياـ-ڪـرـالـامـپـورـ	Malaysia-Kuala Lumpur	0.573	0.786	0.721	0.712
Average		0.391	0.611	0.663	0.576

Table 8: Cosine similarity score between country and capital

6.3 Comparison with WordSim353

We evaluate the performance of our proposed word embeddings using the WordSim353 dataset by translation English word pairs to Sindhi. Due to vocabulary differences between English and Sindhi, we were unable to find the authentic meaning of six terms, so we left these terms untranslated. So our final Sindhi WordSim353 consists of 347 word pairs. Table 9 shows the Spearman correlation results using Eq. 11 on different dimensional embeddings on the translated WordSim353. The Table 9 presents complete results with the different ws for CBoW, SG and GloVe in which the $ws=7$ subsequently yield better performance than ws of 3 and 5, respectively. The SG model outperforms CBoW and GloVe in semantic and syntactic similarity by achieving the performance of **0.629** with $ws=7$. In comparison with English [28] achieved the average semantic and syntactic similarity of 0.637, 0.656 with CBoW and SG, respectively. Therefore, despite the challenges in translation from English to Sindhi, our proposed Sindhi word embeddings have efficiently captured the semantic and syntactic relationship.

Model	ws	Accuracy
CBoW	3	0.568
	5	0.582
	7	0.596
Skip gram	3	0.617
	5	0.621
	7	0.629
GloVe	3	0.542
	5	0.563
	7	0.568
SdfastText		0.374

Table 9: Comparison of semantic and syntactic accuracy of proposed word embeddings using WordSim-353 dataset on $300 - D$ embedding choosing various window size (ws).

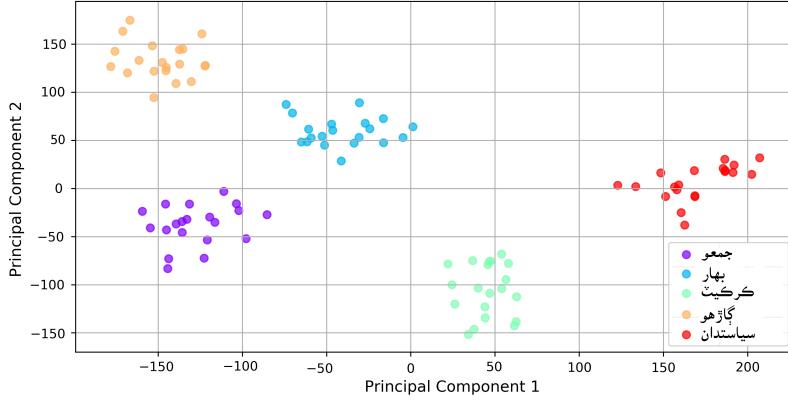


Figure 4: Visualization of Sindhi CBoW word embeddings

6.4 Visualization

We use t-Distributed Stochastic Neighboring (t-SNE) dimensionality [37] reduction algorithm with PCA [38] for exploratory embeddings analysis in 2-dimensional map. The t-SNE is a non-linear dimensionality reduction algorithm for visualization of high dimensional datasets. It starts the probability calculation of similar word clusters in high-dimensional space and calculates the probability of similar points in the corresponding low-dimensional space. The purpose of t-SNE for visualization of word embeddings is to keep similar words close together in 2-dimensional x, y coordinate pairs while maximizing the distance between dissimilar words. The t-SNE has a perplexity (PPL) tunable parameter used to balance the data points at both the local and global levels. We visualize the embeddings using PPL=20 on 5000-iterations of 300-D models. We use the same query words (see Table 6) by retrieving the top 20 nearest neighboring word clusters for a better understanding of the distance between similar words. Every query word has a distinct color for the clear visualization of a similar group of words. The closer word clusters show the high similarity between the query and retrieved word clusters. The word clusters in SG (see Fig. 5) are closer to their group of semantically related words. Secondly, the CBoW model depicted in Fig. 4 and GloVe Fig. 6 also show the better cluster formation of words than SdfastText Fig. 7, respectively.

7 Discussion and future work

In this era of the information age, the existence of LRs plays a vital role in the digital survival of natural languages because the NLP tools are used to process a flow of un-structured data from disparate sources. It is imperative to mention that presently, Sindhi Persian-Arabic is frequently used in online communication, newspapers, public institutions in Pakistan and India. Due to the growing use of Sindhi on web platforms, the need for its LRs is also increasing for the development of language technology tools. But little work has been carried out for the development of resources which is not sufficient to design a language independent or machine learning algorithms. The present work is a first comprehensive initiative on resource development along with their evaluation for statistical Sindhi language processing.

More recently, the NN based approaches have produced a state-of-the-art performance in NLP by exploiting unsupervised word embeddings learned from the large unlabelled corpus. Such word embeddings have also motivated

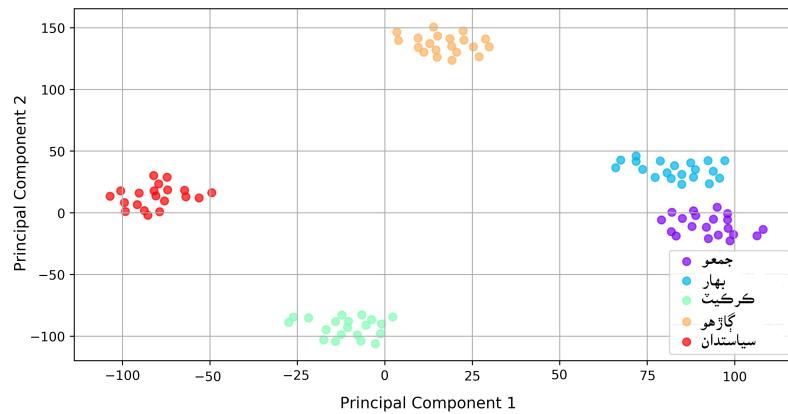


Figure 5: Visualization of Sindhi SG word embeddings

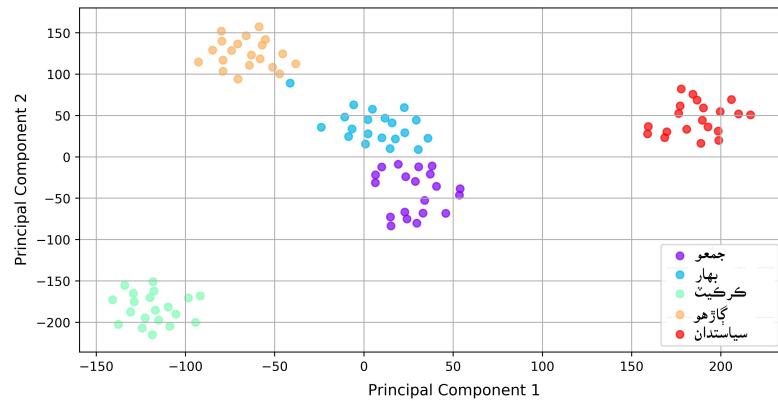


Figure 6: visualization of Sindhi GloVe word embeddings

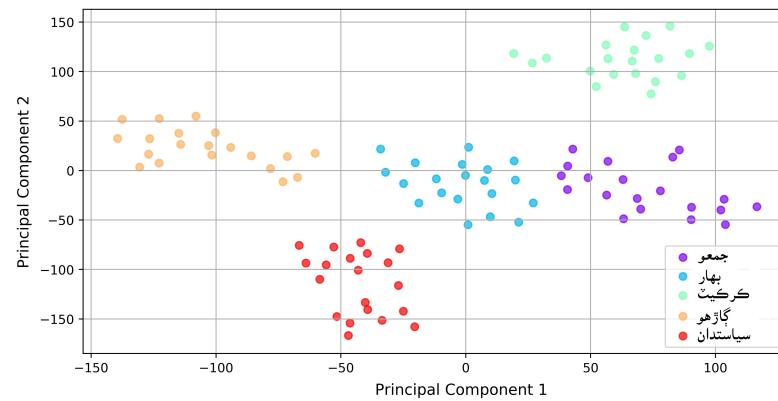


Figure 7: Visualization of SdfastText word embeddings

the work on low-resourced languages. Our work mainly consists of novel contributions of resource development along with comprehensive evaluation for the utilization of NN based approaches in SNLP applications. The large corpus obtained from multiple web resources is utilized for the training of word embeddings using SG, CBoW and Glove models. The intrinsic evaluation along with comparative results demonstrates that the proposed Sindhi word embeddings have accurately captured the semantic information as compare to recently revealed SdfastText word vectors. The SG yield best results in nearest neighbors, word pair relationship and semantic similarity. The performance of CBoW is also close to SG in all the evaluation matrices. The GloVe also yields better word representations; however SG and CBoW models surpass the GloVe model in all evaluation matrices.

Hyperparameter optimization is as important as designing a new algorithm. The choice of optimal parameters is a key aspect of performance gain in learning robust word embeddings. Moreover, We analysed that the size of the corpus and careful preprocessing steps have a large impact on the quality of word embeddings. However, in algorithmic perspective, the character-level learning approach in SG and CBoW improves the quality of representation learning, and overall window size, learning rate, number of epochs are the core parameters that largely influence the performance of word embeddings models. Ultimately, the new corpus of low-resourced Sindhi language, list of stop words and pretrained word embeddings along with empirical evaluation, will be a good supplement for future research in SSLP applications.

In the future, we aim to use the corpus for annotation projects such as parts-of-speech tagging, named entity recognition. The proposed word embeddings will be refined further by creating custom benchmarks and the extrinsic evaluation approach will be employed for the performance analysis of proposed word embeddings. Moreover, we will also utilize the corpus using Bi-directional Encoder Representation Transformer [14] for learning deep contextualized Sindhi word representations. Furthermore, the generated word embeddings will be utilized for the automatic construction of Sindhi WordNet.

8 Conclusion

In this paper, we mainly present three novel contributions of large corpus development contains large vocabulary of more than 61 million tokens, 908,456 unique words. Secondly, the list of Sindhi stop words is constructed by finding their high frequency and least importance with the help of Sindhi linguistic expert. Thirdly, the unsupervised Sindhi word embeddings are generated using state-of-the-art CBoW, SG and GloVe algorithms and evaluated using popular intrinsic evaluation approaches of cosine similarity matrix and WordSim353 for the first time in Sindhi language processing. We translate English WordSim353 using the English-Sindhi bilingual dictionary, which will also be a good resource for the evaluation of Sindhi word embeddings. Moreover, the proposed word embeddings are also compared with recently revealed SdfastText word representations.

Our empirical results demonstrate that our proposed Sindhi word embeddings have captured high semantic relatedness in nearest neighboring words, word pair relationship, country, and capital and WordSim353. The SG yields the best performance than CBoW and GloVe models subsequently. However, the performance of GloVe is low on the same vocabulary because of character-level learning of word representations and sub-sampling approaches in SG and CBoW. Our proposed Sindhi word embeddings have surpassed SdfastText in the intrinsic evaluation matrix. Also, the vocabulary of SdfastText is limited because they are trained on a small Wikipedia corpus of Sindhi Persian-Arabic. We will further investigate the extrinsic performance of proposed word embeddings on the Sindhi text classification task in the future. The proposed resources along with systematic evaluation will be a sophisticated addition to the computational resources for statistical Sindhi language processing.

References

- [1] Jennifer Cole. Sindhi. encyclopedia of language & linguistics volume8, 2006.
- [2] Raveesh Motlani. Developing language technology tools and resources for a resource-poor language: Sindhi. In *Proceedings of the NAACL Student Research Workshop*, pages 51–58, 2016.
- [3] Hidayatullah Shaikh, Javed Ahmed Mahar, and Mumtaz Hussain Mahar. Instant diacritics restoration system for sindhi accent prediction using n-gram and memory-based learning approaches. *INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS*, 8(4):149–157, 2017.
- [4] Abdul-Majid Bhurgri. Enabling pakistani languages through unicode. *Microsoft Corporation white paper at <http://download.microsoft.com/download/1/4/2/142ae9f-1a74-4a24-b1f4-782d48d41a6d/PakLang.pdf>*, 2006.
- [5] Wazir Ali Jamro. Sindhi language processing: A survey. In *2017 International Conference on Innovations in Electrical Engineering and Computational Technologies (ICIEECT)*, pages 1–8. IEEE, 2017.

- [6] Edward Loper and Steven Bird. Nltk: the natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics-Volume 1*, pages 63–70. Association for Computational Linguistics, 2002.
- [7] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60, 2014.
- [8] Wanxiang Che, Zhenghua Li, and Ting Liu. Ltp: A chinese language technology platform. In *Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations*, pages 13–16. Association for Computational Linguistics, 2010.
- [9] Martin Popel and Zdeněk Žabokrtský. Tectomt: modular nlp framework. In *International Conference on Natural Language Processing*, pages 293–304. Springer, 2010.
- [10] Lluís Padró, Miquel Collado, Samuel Reese, Marina Lloberes, and Irene Castellón. Freeling 2.1: Five years of open-source language processing tools. In *7th International Conference on Language Resources and Evaluation*, 2010.
- [11] Waqar Ali Narejo and Javed Ahmed Mahar. Morphology: Sindhi morphological analysis for natural language processing applications. In *2016 International Conference on Computing, Electronic and Electrical Engineering (ICE Cube)*, 2016.
- [12] Yang Li and Tao Yang. Word embedding for understanding natural language: a survey. In *Guide to Big Data Applications*, pages 83–104. Springer, 2018.
- [13] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.
- [14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.
- [15] Mutee U Rahman. Towards sindhi corpus construction. In *Conference on Language and Technology, Lahore, Pakistan*, 2010.
- [16] Fida Hussain Khoso, Mashooque Ahmed Memon, Haque Nawaz, and Sayed Hyder Abbas Musavi. To build corpus of sindhi. 2019.
- [17] Mazhar Ali Dootio and Asim Imdad Wagan. Unicode-8 based linguistics data set of annotated sindhi text. *Data in brief*, 19:1504–1514, 2018.
- [18] Mazhar Ali Dootio and Asim Imdad Wagan. Development of sindhi text corpus. *Journal of King Saud University-Computer and Information Sciences*, 2019.
- [19] Mazhar Ali and Asim Imdad Wagan. Sentiment summarization and analysis of sindhi text. *Int. J. Adv. Comput. Sci. Appl.*, 8(10):296–300, 2017.
- [20] Kevin Lund and Curt Burgess. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior research methods, instruments, & computers*, 28(2):203–208, 1996.
- [21] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.
- [22] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.
- [23] Jacob Andreas and Dan Klein. How much do word embeddings encode about syntax? In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 822–827, 2014.
- [24] Tobias Schnabel, Igor Labutov, David Mimno, and Thorsten Joachims. Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 298–307, 2015.

- [25] Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. Advances in pre-training distributed word representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, 2018.
- [26] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. Learning word vectors for 157 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, 2018.
- [27] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.
- [28] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- [29] Neha Nayak, Gabor Angeli, and Christopher D Manning. Evaluating word embeddings using a representative suite of practical tasks. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 19–23, 2016.
- [30] Bénédicte Pierrejean and Ludovic Tanguy. Towards qualitative word embeddings evaluation: measuring neighbors variation. 2018.
- [31] Eneko Agirre, Enrique Alfonsena, Keith Hall, Jana Kravalova, Marius Paşa, and Aitor Soroa. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 19–27. Association for Computational Linguistics, 2009.
- [32] Roland Schäfer and Felix Bildhauer. Web corpus construction. *Synthesis Lectures on Human Language Technologies*, 6(4):1–145, 2013.
- [33] Zeeshan Bhatti, Imdad Ali Ismaili, Waseem Javaid Soomro, and Dil Nawaz Hakro. Word segmentation model for sindhi text. *American Journal of Computing Research Repository*, 2(1):1–7, 2014.
- [34] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.
- [35] Siyu Qiu, Qing Cui, Jiang Bian, Bin Gao, and Tie-Yan Liu. Co-learning of word representations and morpheme representations. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 141–150, 2014.
- [36] Omer Levy, Yoav Goldberg, and Ido Dagan. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225, 2015.
- [37] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.
- [38] Rémi Lebret and Ronan Collobert. Word emdeddings through hellinger pca. *arXiv preprint arXiv:1312.5542*, 2013.
- [39] Amaresh Kumar Pandey and Tanvver J Siddiqui. Evaluating effect of stemming and stop-word removal on hindi text retrieval. In *Proceedings of the First International Conference on Intelligent Human Computer Interaction*, pages 316–326. Springer, 2009.
- [40] Mikhail Khodak, Andrej Risteski, Christiane Fellbaum, and Sanjeev Arora. Automated wordnet construction using word embeddings. In *Proceedings of the 1st Workshop on Sense, Concept and Entity Representations and their Applications*, pages 12–23, 2017.
- [41] Andriy Mnih and Koray Kavukcuoglu. Learning word embeddings efficiently with noise-contrastive estimation. In *Advances in neural information processing systems*, pages 2265–2273, 2013.
- [42] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*, pages 2177–2185, 2014.
- [43] Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. Placing search in context: The concept revisited. *ACM Transactions on information systems*, 20(1):116–131, 2002.

- [44] Alvaro Corral, Gemma Boleda, and Ramon Ferrer-i Cancho. Zipf's law for word frequencies: Word forms versus lemmas in long texts. *PloS one*, 10(7):e0129031, 2015.