Sense disambiguation for Punjabi language using supervised machine learning techniques

VARINDER PAL SINGH* and PARTEEK KUMAR

Computer Science and Engineering Department, Thapar Institute of Engineering and Technology,
Patiala 147004, India
e-mail: varinderpal@thapar.edu; parteek.bhatia@thapar.edu

MS received 21 June 2017; revised 28 May 2019; accepted 13 August 2019

Abstract. Automatic identification of a meaning of a word in a context is termed as Word Sense Disambiguation (WSD). It is a vital and hard artificial intelligence problem used in several natural language processing applications like machine translation, question answering, information retrieval, etc. In this paper, an explicit WSD system for Punjabi language using supervised techniques has been analysed. The sense tagged corpus of 150 ambiguous Punjabi noun words has been manually prepared. The six supervised machine learning techniques Decision List, Decision Tree, Naive Bayes, K-Nearest Neighbour (K-NN), Random Forest and Support Vector Machines (SVM) have been investigated in this proposed work. Every classifier has used same feature space encompassing lexical (unigram, bigram, collocations, and co-occurrence) and syntactic (part of speech) count based features. The semantic features of Punjabi language have been devised from the unlabelled Punjabi Wikipedia text using *word2vec* continuous bag of word and skip gram shallow neural network models. Two deep learning neural network classifiers multilayer perceptron and long short term memory have also been applied for WSD of Punjabi words. The word embedding features have experimented on six classifiers for the Punjabi WSD task. It has been observed that the performance of the supervised classifiers applied for the WSD task of Punjabi language has been enhanced with the application of word embedding features. In this work, an accuracy of 84% has been achieved by LSTM classifier using word embedding feature.

Keywords. Lexical features; syntactic features; word embedding; supervised learning techniques; word sense disambiguation.

1. Introduction

The inference of the right sense of the word in a given context by a computer is known as Word Sense Disambiguation (WSD). Natural languages are ambiguous, and the words of a language can take different meaning with respect to its context in which they have been used. For example, the word “*bark*” can refer to the sound made by a dog in the sentence “*the dog barks loudly*” and in the sentence “*the bark of mango tree is hard*” refers to covering of a tree. Clearly, in the first sentence the words “*dog*” and “*loudly*” indicate the ‘sound made by a dog’ sense of word “*bark*”. While in the second sentence, words “*tree*”, “*mango*”, “*hard*” refer to the ‘tree covering’ sense of the word. Human beings can understand these ambiguities because of cognizance of syntax and semantics of language along with world knowledge. The computers need the complete acquisition of knowledge about language to perform WSD task. This makes WSD an artificial intelligence problem [1]. WSD is considered as an intermediate step in

almost all Natural Language Processing (NLP) applications like machine translation, question answering, information retrieval, etc. The removal of ambiguity is the primary concern of all the NLP applications to achieve higher accuracy.

There are two variants of WSD, i.e., lexical (targeted) and all-word WSD. The lexical sample WSD disambiguates a single targeted word of the sentence while all-word WSD has wide coverage and it disambiguates every word in the text. But all-word WSD, suffers from data sparseness problem as training of classifiers does not cover the whole text of the language. The lexical WSD performs better than all-word WSD as it considers limited ambiguous words.

The WSD has been performed using supervised, unsupervised and knowledge base methods. The supervised methods of WSD use the manually sense tagged data set to train the classifier. The classifiers evaluate the context of the target ambiguous word to find its sense. The manual sense tagging is language-specific and time-consuming task. However, the supervised methods are the best performing methods for WSD [1, 2]. The unsupervised methods create annotated data sets by assuming similar

*For correspondence
Published online: 31 October 2019

senses occur in a similar context. Nevertheless, these methods also need manual intervention to map the induced clusters of words with senses of sense inventory. The knowledge base methods use an external lexical resource like a dictionary, tagged corpus, thesauri and semantic networks. The knowledge base methods have wide coverage and are rivalling the supervised methods [3, 4].

The supervised WSD for the English language has been successfully implemented using different classification algorithms like Naïve Bayes (NB) [1], Decision Trees (DT) [1, 4], Support Vector Machine (SVM) [4, 5], K-Nearest Neighbour (K-NN) [4]. The classifiers have been trained on different lexical and syntactic features extracted from the sense tagged context of the ambiguous words. IMS (It Make Sense) WSD system of English language has applied surrounding words and their Part-Of-Speech (POS) as features to yield state-of-the-art performance [5]. The manually crafted lexical and syntactic features are count based and sparse. The sparsity of text features also known as the curse of dimensionality leads to degradation of performance of supervised WSD.

The sparsity and non-linearity problems of features have been handled by Baingo *et al* [6] using Neural Network (NN) classifier for the different NLP tasks. The NN has been used to extract the semantic information from the large unsupervised text called embedding. These embedding features when used by deep neural network techniques namely Multi-Layer Perceptron (MLP), Long Short Term Memory (LSTM) and by traditional classifiers namely SVM and K-NN have shown an improvement in the performance of WSD task of English language [7].

This paper examines the effectiveness of the lexical, syntactic and word embedding features for supervised WSD task for the Punjabi language. The major contributions toward these issues are as follows.

- The lexical supervised WSD task for the Punjabi language has been performed, evaluated and compared using supervised approaches, namely, Decision List (DL), DT, NB, K-NN, Random Forest (RF), SVM, MLP, and LSTM.
- The supervised classifiers have been trained on the sense tagged data set for 150 most ambiguous Punjabi noun words extracted from Punjabi WordNet [8]. The fine senses of extracted words have been manually merged to generate coarse senses. The sense tagged data set of coarse senses of 150 ambiguous words have been prepared manually.
- The lexical features like unigram, bigram with different window sizes (2, 3 and 4), collocations and co-occurrence have been devised and explored by six different classification approaches. Similarly, the syntactic feature such as POS of surrounding words of target word has experimented on six different classification techniques. The open source Punjabi shallow parser (developed by Consortium of Institutions headed

by IIIT Hyderabad, India) has been applied to yield the POS tags of the sense tagged data set.

- The distributional word embedding features have been prepared from Wikipedia the unlabelled corpus of Punjabi language using two efficient NN models of *word2vec* CBOW and skip-gram [9]. Three different dimensions of CBOW and skip-gram embedding have been evaluated on six ML and two deep learning classifiers. To further fine tune the WSD for Punjabi language the pre-trained embedding by *FastText* model [10] has been evaluated on two deep learning classifiers.
- The proposed WSD system using eight classifiers has been applied individually on 150 ambiguous words. The WSD architecture evaluates the performance of the individual classifier on lexical, syntactic and embedding features.

In this paper, it has been observed that the WSD for the Punjabi language is complex due to non-availability of lexical resource. It has been ascertained that the performance of Punjabi WSD using traditional lexical features is exceedingly better than the syntactic features. The performance of WSD Punjabi language has shown improvement when the distributional word embedding features are employed.

This paper has been divided into eight sections. Section 2 gives an insight into the related work. Section 3 provides a brief introduction to the Punjabi language. Corpus creation using Punjabi WordNet has been elaborated in section 4. Lexical, Syntactic and Word embedding features used in this paper are given in section 5. Section 6 outlines the basic WSD architecture of Punjabi language. The analysis of experimental results has been reported in section 7 and conclusions are provided in section 8.

2. Related work

The researchers have explored the different statistics and artificial intelligence methods for WSD for various languages of the world [4]. This section reviews the supervised WSD using different set of influential features devised and experimented on the different machine learning and deep machine learning classifiers for WSD.

2.1 Supervised WSD

Supervised methods leverage the sense tagged data sets. The availability of lexical resources like sense tagged corpus, semantic corpus [1, 4] and sense inventory like WordNet [11], has led to a successful implementation of supervised approaches for WSD. SENSEVAL (now SEMEVAL) competitions have been established to determine the strengths and weaknesses of WSD programs with

respect to different words and different languages [1, 2, 4]. The different supervised learning algorithms NB [4], RF [12], DT [13], Rule Based [4] and K-NN [1] have been evaluated for WSD of Chinese, Japanese, Korean, Hindi and English language SENSEVAL data sets [2, 3]. Florian *et al* [15] worked on lexical WSD of English, Spanish, Swedish and Basque languages sample data sets using supervised learning techniques. The NN with multilayers have re-emerged and successfully applied on supervised WSD task [14]. However, the supervised WSD models have bottlenecks like availability of sense tagged data set and language specificity.

2.2 Different features used by supervised WSD

The lexical supervised WSD methods have successfully used context words of the target word. The unigrams, bigrams, collocations, POS relations, headword features of English language have shown state-of-the-art accuracy when used by DT classifier for WSD task [13]. The IMS WSD system has experimented both the lexical (different collections of words) and syntactic (grammatical relations among words) features on SVM classifier [5]. Narouei *et al* [12] have demonstrated the application of iterative patterns extracted as monolingual features from standard English language corpus on SVM and RF supervised classifiers. The more influential features based on the frequency of words in the corpus has been applied for supervised WSD [15].

Recently, the word embedding features created by NN feedforward network have been successfully applied for WSD of English language [16]. These predictions based features capture the semantic relations among the words of unlabelled corpus. The efficient word embedding algorithm *word2vec* by Mikilov *et al* has yielded the best performance for various NLP tasks [9]. The *word2vec* embedding has been successfully used by different deep learning and traditional classifiers for WSD task of the English language [7, 17, 18]. The word embedding feature formed by mixing the concepts with words has been applied for the supervised WSD [17]. The WSD has further being fine-tuned by applying the pre-trained word embedding [19].

2.3 Deep learning NN for supervised WSD

Recently, the use of embedding features by upsurge deep learning NN classifiers have shown an improvement in the performance of supervised WSD [20]. The LSTM and window probability maximization methods using word embedding methods have improved the performance of WSD [21]. Bi-directional LSTM using word embedding has improved the performance of English language WSD [18]. LSTM model using word embedding and WordNet has improved the performance of all-word WSD [22]. The

use of sequence learning along with the word embedding has enhanced the performance of LSTM classifiers for all-word WSD models [23]. However, this research is limited to resource-rich languages like English, Chinese, Spanish, etc.

2.4 Motivations

The main motivation of this study is to evaluate the predictive performance of the supervised classifiers for the resource-limited Punjabi language by exploiting the hand-crafted and prediction based features. The presented study aims to identify the lexical and syntactic optimal features for the supervised classifier of WSD. The generated embedding features have been explored and evaluated for the supervised WSD of Punjabi language.

The existing studies of WSD for the Punjabi language have primarily based on knowledge based method. Rana and Kumar [24] have applied simple LESK algorithm to perform Punjabi WSD; and Singh and Singh [25] have applied enhanced LESK algorithm to perform Punjabi WSD on 15 ambiguous noun words. The *n-gram* approaches have been applied to determine the right sense of the word in machine translation [26] and transliteration [27] of Punjabi language. Unfortunately, for the Punjabi language no supervised approaches have been applied due to the dearth of sense annotated corpus. The availability of computational resources like Punjabi WordNet [8] in Gurmukhi script has motivated us to build the proposed supervised WSD system. From the literature review, it has been observed that different supervised and deep learning classifiers have been successfully applied for the WSD of different languages. It has encouraged us to investigate the performance classifiers over the diverse range of the feature sets for disambiguation of Punjabi language noun words.

3. Introduction to Punjabi language

The Punjabi language is classified as a member of the Indo-Aryan subgroup of the Indo-European languages. It is the world's 10th most widely spoken language [28]. The Punjabi language is the official language of the Indian state of Punjab. It is also the Canadian parliament's third language after English and French. There are four scripts of Punjabi language, i.e., Perso-Arabic Shahmukhi, Gurmukhi, Devnagri and LaNDA [28]. The most widely used script of the Punjabi language is Gurmukhi script. The availability of Punjabi language text in Gurmukhi script has motivated us to use it. In Punjabi language, ambiguity exists at many levels like semantic, syntactic, internal word structure and anaphoric. A core issue of WSD for the Punjabi language is the absence of computational resources like a standard online dictionary, sense tagged and labelled data sets. Further

Table 1. Free word order examples.

Sample instance	English translation
P: ਰਾਜ ਘੋੜਾ ਲੈ ਗਿਆ। T: rāj ghōṛā lai giā	Raj took the horse
P: ਘੋੜਾ ਰਾਜ ਲੈ ਗਿਆ। T: ghōṛā rāj lai giā	The horse took Raj

Table 2. Null-subject examples.

Sample instance	English translation
P: ਬਾਹਰ ਮੀਂਹ ਪੈ ਰਿਹਾ ਹੈ। T: bāhar mīṃh pai rihā hai	It is raining outside
P: ਲੜਾਈ ਸ਼ੁਰੂ ਹੋ ਸਕਦੀ ਹੈ। T: laṛāī shurū hō sakadī hai	There could be a war

Table 3. Gender overlap example.

Sample instance	English translation
P: ਰਾਮ ਨੇ ਅਖਬਾਰ ਪੜ੍ਹ ਲਿਆ ਹੈ। T: rām nē akhbār paṛh liā hai	Ram has read the paper
P: ਰਾਮ ਨੇ ਅਖਬਾਰ ਪੜ੍ਹ ਲਈ ਹੈ। T: rām nē akhbār paṛh lai hai	Ram has read the paper

Punjabi language has special characteristics that need special consideration to perform supervised WSD.

3.1 Diversion of Punjabi over English language

There are many differences between Punjabi and English languages and some of the key points observed are listed below.

- The sentence structure of Punjabi is SOV (Subject-Object-Verb) as compared to English SVO (Subject-Verb-Object) sentence structure.
- The Punjabi has postpositions rather than prepositions of English.
- Punjabi language does not include capitalization.
- Punjabi is relatively a free word order language while English is fixed order language as shown in table 1, Punjabi sentences (prefixed with P: and its respective transliteration with T:), with different word order, have the same meaning.
- Punjabi is a null-subject language where the subject is not required for completion of a sentence while the English language usually uses “It” and “There” to complete the sentences as shown in table 2.
- Punjabi language is highly inflectional language [29] and has rich morphological variations while using verbs, adjectives, and nouns than the English language.

For example, Punjabi verbs like ਗਈ (gāī) {WordClass: Main Verb; Gender: Feminine; Number: Singular;}, ਗਏ (gāē) {WordClass: Main Verb; Gender: Masculine; Number: Plural;}, ਗਿਆ (giā) {WordClass: Main Verb; Gender: Masculine; Number: Singular} have only one equivalent ‘gone’ in English.

Adjectives in Punjabi are inflected for number, case and gender of nouns they qualify, like in ਛੋਟੀ ਬੱਚੀ (chōṭī baccī) where ਛੋਟੀ (chōṭī) {WordClass: Inflected Adjective; Gender: Feminine; Number: Singular; Case: Direct;} is an adjective for ‘girl’ and ਛੋਟਾ ਬੱਚਾ (chōṭā baccā) where ਛੋਟਾ (chōṭā) {WordClass: Inflected Adjective; Gender: Masculine; Number: Singular; Case: Direct;} is an adjective for ‘boy’. In the case of English same inflected adjective, i.e., ‘little’ has been used for both ‘boy’ and ‘girl’.

Nouns of Punjabi language are inflected for case and number. For example, word ਮੁੰਡਾ (muṇḍā) {WordClass: Noun; Gender: Masculine; Number: Singular; Case: Direct;} has four inflectional forms ਮੁੰਡੇ (muṇḍē) {WordClass: Noun; Gender: Masculine; Number: Singular; Case: Oblique and Direct;}, ਮੁੰਡੀਆ (muṇḍiā) {WordClass: Noun; Gender: Masculine; Number: Singular; Case: Vocative;}, ਮੁੰਡਿਆਂ (muṇḍiām) {WordClass: Noun; Gender: Masculine; Number: Plural; Case: Oblique;} and ਮੁੰਡਿਓ (muṇḍiō) {WordClass: Noun; Gender: Masculine; Number: Plural; Case: Vocative;} whereas it has only one English equivalent ‘boy’.

- The use of two gender forms masculine and feminine for the same noun word is frequent in the Punjabi language as shown in table 3. The same word ਅਖਬਾਰ (akhbār) ‘newspaper’ of Punjabi language has been treated as masculine and feminine. On the other hand, English words have distinct gender forms.
- The Punjabi language has two-word verbs called double and conjunct verbs. For example, ਤੁਰ-ਤੁਰ ਕੇ (tur-tur kē), ਖਾਧਾ-ਪੀਤਾ (khādhā-pītā), ਖੇਡ-ਖੇਡ ਕੇ (khēḍ-khēḍ kē) are double verbs.
- The particles of Punjabi language like ਜੀ (jī), and ਹੀ (hī) do not have any equivalent in the English language.
- The Punjabi language has a severe spelling variation problem. This enhances the complexity of the preprocessing task of cleaning the corpus data with standard spelling.
- Punjabi language users frequently include the English language words in the context. This makes the analysis of the Punjabi language context a complex task. The English words ‘percent’ and ‘sir’ are used in the respective two example sentences as shown in table 4.

These different structural and grammatical challenges of Punjabi language make the WSD harder. To understand these rich variations of attributes of the Punjabi context words, the system needs robust morph analyser so that the

Table 4. Multi lingual examples.

Sample instance	English translation
P: ਸੈਟ ਪ੍ਰਸੈਟ ਅੱਜ ਮੀਂਹ ਪਉਗਾ। T: saṭṭ prasaiṭ aṭṭ mīmḥ paugā	It will rain today 100 percent
P: ਸਰ ਜੀ ਮੇਰੀ ਬੇਨਤੀ ਤਾਂ ਸੁਣੋ। T: sar jī mērī bēntī tāṁ suṇō	Sir please listen to my request

Table 5. Newspaper and same noun verb examples.

Issues	Sample instance	English translation
Ambiguous newspaper headline	P: ਆਮ ਆਦਮੀ ਦੀ ਹੈ ਆਮ ਆਦਮੀ ਪਾਰਟੀ।	Aam Aadmi party is people's party
	T: ām ādmī dī hai ām ādmī pāraṭī	
Same verb and noun	P: ਪੂਜਾ ਨੇ ਪੂਜਾ ਦੀ ਥਾਲੀ ਨਾਲ ਪੂਜਾ ਕੀਤੀ। T: pūjā nē pūjā dī thālī nāl pūjā kītī	Pooja has worshiped using a plate

Table 6. Punjabi WordNet Coverage.

Noun	Verb	Adjective	Adverb	Total
23225	2836	5830	443	32334

right sense of the word can be mined. Notably good pre-processor for the Punjabi language is required, to take care of spelling and multi-lingual issues.

Besides the differences from the English language, there are some common issues like an ambiguous newspaper headline and distinctions of the same verb and noun as shown in example instance of table 5.

3.2 Need of WSD for Punjabi language

To accomplish most of the NLP tasks, sense disambiguation of a word is an important intermediate step. Chan *et al* [30] have shown that WSD can improve machine translation (MT). WSD for the Punjabi language can benefit MT

by identifying the right sense of the word and can also help in improving the quality of different NLP applications like information retrieval, sentiment analysis, Named Entity Recognition (NER), semantic web, content analysis, and speech processing, etc. Currently, no explicit WSD system using supervised techniques for Punjabi language exists. In this paper, different supervised classifiers have been evaluated on lexical, syntactic and word embedding features extracted from sense tagged and unlabelled data sets.

4. Punjabi WordNet: lexical resource used to build corpus

The sense inventory Punjabi WordNet has been used to prepare sense tagged data set of 150 most ambiguous Punjabi language noun words. WordNet is a large lexical database of words and their respective meanings. Synonymous words are grouped together to form synonym sets called synsets. WordNet provides the distinct senses and semantic relations of the words. Punjabi WordNet is one of the constituents of Indo WordNet which is a linked structure of WordNets of major Indian languages from Indo-Aryan, Dravidian and Sino-Tibetan families [1, 31]. The Punjabi WordNet has total of 32334 words of four different part of speech as shown in table 6.

The Punjabi WordNet defines the fine senses of words, i. e., overlapping senses of the words. There are 23225 nouns in Punjabi WordNet out of which 2460 nouns have more than single fine sense. The average of fine senses of 2460 nouns is 2.16. The 150 most ambiguous Punjabi nouns have selected out of 2460 nouns. The maximum fine sense of the selected ambiguous word is 15. While the minimum sense is 3. The average fine senses of 150 ambiguous words are 6.45. It has a standard deviation of 1.92 which shows the widespread distribution of senses in the data set.

The sense granularity is a major challenge in the development of the WSD system [4]. The overlapped fine senses are merged to form clear and coarse senses which are more useful in WSD system [4, 32]. The coarse senses of an ambiguous word have been created by merging the overlapping fine senses of an ambiguous word extracted from Punjabi WordNet. The different relationships (hypernymy, hyponymy, etc.) of all 150 most ambiguous nouns

Table 7. Five most ambiguous Punjabi noun words.

Word	Coarse senses	#Instances per sense								Total
		S1	S2	S3	S4	S5	S6	S7	S8	
ਜੋੜ(jōr)	8	15	76	8	34	46	22	69	92	370
ਗੱਠ(gaṭṭh)	7	26	35	5	5	33	128	9	–	248
ਮੇਲ(mēl)	7	5	19	5	10	6	5	39	–	96
ਤਲਾ(talā)	7	5	17	19	5	5	46	5	–	109
ਚਾਲ(cāl)	7	40	14	13	17	5	23	21	–	140

Figure 1. Fine and coarse sense of 150 Punjabi ambiguous words.

Table 8. Coarse sense tagged data set.

#Word	#Senses per word			#Instances per sense			Total instance
	Max.	Min.	Avg.	Max.	Min.	Avg.	
150	8	2	4.03	435	5	36.6	22153

described in Punjabi WordNet have been manually analysed to merge fine senses into coarse senses. The relation between fine and coarse senses of all 150 ambiguous Punjabi noun words is shown in figure 1. The average coarse senses of 150 ambiguous words are 4.03.

Based on the coarse senses of the ambiguous words, the corpus for training data set has been prepared manually from resources like online newspapers, articles, blogs and Punjabi Wikipedia. The coarse sense training data set consists of 22153 instances of different senses of 150 ambiguous Punjabi words as shown in table 8. The maximum and minimum coarse sense of an ambiguous word is 8 and 2. There are maximum of 435 instances of the most frequent sense of the word ਭਾਵ (bhāv) in the coarse sense tagged data set. The minimum number of instances for any coarse sense of ambiguous words is 5. The number of average instances per sense of the target ambiguous words is 36.6.

Table 7 shows the statistics of the five ambiguous words selected from the coarse sense tagged data set. From the context of ambiguous words, the important features that suggest its specific sense are discussed in the next section.

5. Lexical, syntactic and word embedding features

The lexical features are the words surrounding the target ambiguous word whereas, the syntactic features capture the specific syntactic relations among target word and its neighbourhood words, based on grammar rules of a language [13]. The semantic relations among the words are captured using word embedding. The description of these features extracted from Punjabi language corpus is as follows.

Table 9. Extracted lexical features.

Example instance	P: ਸਾਡੇ ਭਾਰਤ ਦੇ ਉੱਤਰ ਵਿਚ ਹਿਮਾਲਿਆ ਪਰਬਤ ਹੈ
Translation	T: sādē bhārat dē uttar vic himālīā parbat hai ET: The Himalayan mountain is situated in north of our India.
Stop words removed	P: ਦੇ, ਵਿਚ, ਹੈ T: dē, vic, hai
Word/ Tokens	P: ਸਾਡੇ, ਭਾਰਤ, ਉੱਤਰ, ਹਿਮਾਲਿਆ, ਪਰਬਤ T: sādē, bhārat, uttar, himālīā, parbat
Unigrams	P: ਸਾਡੇ, ਭਾਰਤ, ਉੱਤਰ, ਹਿਮਾਲਿਆ, ਪਰਬਤ T: sādē, bhārat, uttar, himālīā, parbat
Bigram Window Size 2	P: ਸਾਡੇ-ਭਾਰਤ, ਭਾਰਤ-ਉੱਤਰ, ਉੱਤਰ-ਹਿਮਾਲਿਆ, ਹਿਮਾਲਿਆ-ਪਰਬਤ T: sādē-bhārat, bhārat-uttar, uttar- himālīā, himālīā-parbat
Bigram Window Size 3	P: ਸਾਡੇ-ਭਾਰਤ, ਸਾਡੇ-ਉੱਤਰ, ਭਾਰਤ-ਉੱਤਰ, ਭਾਰਤ-ਹਿਮਾਲਿਆ, ਉੱਤਰ-ਹਿਮਾਲਿਆ, ਉੱਤਰ- ਪਰਬਤ, ਹਿਮਾਲਿਆ-ਪਰਬਤ T: sādē-bhārat, sādē- uttar, bhārat- uttar, bhārat- himālīā, uttar - himālīā, uttar - parbat, himālīā -parbat
Bigram Window Size 4	P: ਸਾਡੇ-ਭਾਰਤ, ਸਾਡੇ-ਉੱਤਰ, ਸਾਡੇ-ਹਿਮਾਲਿਆ, ਭਾਰਤ-ਉੱਤਰ, ਭਾਰਤ-ਹਿਮਾਲਿਆ, ਭਾਰਤ- ਪਰਬਤ, ਉੱਤਰ-ਹਿਮਾਲਿਆ, ਉੱਤਰ-ਪਰਬਤ, ਹਿਮਾਲਿਆ-ਪਰਬਤ T: sādē-bhārat, sādē-uttar, sādē-himālīā, bhārat-uttar, bhārat-himālīā, bhārat-parbat, uttar-himālīā, uttar -parbat, himālīā -parbat
Collocation	P: ਭਾਰਤ, ਉੱਤਰ, ਹਿਮਾਲਿਆ T: bhārat, uttar, himālīā

5.1 Lexical features

The lexical features like unigram, bigram with window size 2, 3, 4, collocations and co-occurrence have been explored in this paper. Table 9 illustrates the lexical features extracted for an ambiguous word ਉੱਤਰ (uttar). Initially, the stop words, for example, ਦੇ (dē), ਵਿਚ (vic), ਹੈ (hai) have been excluded with the help of a manually prepared dictionary. The details of lexical features that have been used in this proposed work are described below.

- Unigram: The context words around the target ambiguous word are known as unigrams. The unigrams of

target word w_t having m senses are extracted by applying equation (1). The unigram features have been extracted from the individual sense file of the target ambiguous word.

$$\text{unigram}(w_t) = \cup_{i=1}^m \{w_1, w_2, \dots, w_n \in S_i\} \quad (1)$$

The unigrams ਭਾਰਤ(bhārat), ਹਿਮਾਲਿਆ (himālīā), and ਪਰਬਤ (parbat) of sentence ‘ਸਾਡੇ ਭਾਰਤ ਦੇ ਉੱਤਰ ਵਿਚ ਹਿਮਾਲਿਆ ਪਰਬਤ ਹੈ’ (T: sādē bhārat dē uttar vic himālīā parbat hai) (ET: The Himalayan mountain is situated in north of our india) as shown in table 9 indicate the ‘direction’ sense of the target word ਉੱਤਰ (utarr). While, in the sentence ‘ਹਰ ਸਵਾਲ ਦਾ ਢੁਕਵਾਂ ਉੱਤਰ ਹੈ’ (T: har savāl dā ḍhukkvām uttar hai) (ET: Every question has appropriate answer) the words ਸਵਾਲ (savāl), ਢੁਕਵਾਂ (ḍhukkvām) indicate the ‘answer’ sense of the target word ਉੱਤਰ (utarr).

- Bigrams: The sequences of two-words in the context of an instance are known as bigrams. Bigrams include the target ambiguous word along with all surrounding words in an instance. Equation (2) shows the bigram features of target word w_t extracted from individual sense file with window size ranging from 2 to k .

$$\text{bi}(w_t) = \cup_{i=1}^m \{ \cup_{win=2}^k (w_1, w_2, \dots, w_k) \in S_i \} \quad (2)$$

Bigram features are based on the window size. The common window size for selecting bigrams from context is two. In the context indicative words of target ambiguous word may not occur consecutively and have words between them. To handle such issues, the window size of 3 and 4 have been applied to include the bigrams which have 3 or 4 words in between them.

- Collocation features: Collocations are the words in an ordered sequence. For example, the ordered words ਉੱਤਰ ਦਿਸ਼ਾ (utarr dishā) indicate the direction sense of an ambiguous word ਉੱਤਰ (utarr). The collocation features have been extracted from the training data set as shown in equation (3).

$$\text{Coll}(w_t) = \cup_{i=1}^m \{ (w_{-1}, w_t, w_{+1}) \in S_i \} \quad (3)$$

The collocation feature contains target ambiguous word w_t along with one-word before (w_{-1}) and one-word after (w_{+1}) it. The unique collocations are considered as features of the target word.

- Co-occurrence features: The indicative words which occur in the context of an ambiguous word with a variable number of intermediary words between them are known as co-occurrence words. For example, the occurrence of word ਪ੍ਰਸ਼ਨ (prashan) ‘question’ in the context of an ambiguous word ਉੱਤਰ (utarr) clearly indicates its ‘answer’ sense. Co-occurrence is determined by estimating the count of individual word w_i present in the respective sense S_k of the target ambiguous word w_t as shown in equation (4).

$$\text{Coccur}(w_i, w_t) = \sum_{S_k} w_i \quad (4)$$

The common co-occurring words with unique frequencies are selected, while the words with same frequencies are not considered. The words that occur more than the minimum value (determined for individual word) are selected as co-occurrence features. There are variable numbers of co-occurring words for an individual sense of the ambiguous word. The number of unique co-occurring words occurring in the individual sense of four selected Punjabi ambiguous words are given in table 10.

5.2 Syntactic features

The syntactic features have been extracted with the help of Punjabi Shallow parser [29]. The shallow parser assigns POS tags to the context words of the training and test data sets. Figure 2 shows the POS tags of context words around the target ambiguous word of an example instance given in table 9. These tags help in narrowing down the search of right sense for an ambiguous word in the sentence. The noun POS tag of target word ਉੱਤਰ (utarr) clearly points toward its noun sense and rule out verb sense. The target ambiguous word and its neighbouring context words of example instance are represented as P_0 and P_{-2}, P_{-1}, P_{+1} ,

Table 10. Co-occurrence word for per sense.

Words	Co-occurring words per sense			
	S1	S2	S2	S4
ਤਾਲ (tāl)	24	10	27	289
ਫਲ (phaḷ)	383	42	8	15
ਹਾਰ (hār)	163	72	623	–
ਯੋਗ (yōg)	136	171	269	–

Figure 2. Extracted syntactic features.

P_{+2} respectively. The 0th index indicates the target word while negative index indicates words on left of the target word and positive index indicates the words on the right of the target word. Five different combinations of local POS tags (P_{-1}, P_0) {NNP, NN}, (P_0, P_{+1}) {NN, VM}, (P_{-1}, P_0, P_{+1}) {NNP, NN, VM}, (P_{-1}, P_{+1}) {NNP, VM} and $P_{-2}, P_{-1}, P_0, P_{+1}, P_{+2}$ {PRP, NNP, NN, VM, NN} have been extracted as syntactic features.

5.3 Word embedding features

The word embedding features have been the most successful technique of distributional semantics and have been successfully integrated with supervised WSD system for the English language [7]. These NN based approaches embed words into low dimensional space. The d-dimensional vector of embedded words is the collection of real numbers. The closeness of vectors depicts the semantic relations between them. The *word2vec* and *FastText* two different word embedding techniques have been applied in this work. The details of these techniques are as follows.

- **WORD2VEC:** The *word2vec* is an efficient three-layer NN based algorithm for yielding word embedding. There are two different models of *word2vec*, Continuous Bag of Word (CBOW) and Skip-Gram (SG). The CBOW model generates a single target word by looking around its context words. The skip-gram model determines the context words of the target word within a symmetric window [33]. Both the CBOW and SG models have been used to yield word embedding features from the unlabelled Punjabi Wikipedia data set. During the preparation of CBOW and SG word embedding features all words of the Punjabi Wikipedia has been included. Three different dimensional (100/200/300) word vectors of the Punjabi language using a context window size of 5 have been generated by CBOW and SG *word2vec* models. Python *gensim* library has been used to extract both the CBOW and SG word vectors [34].

- **FASTTEXT:** The *word2vec* vectors represent an individual word as the unique vector ignoring the internal structure of the word. The morphological rich languages that have many word forms have poor word representation. The *FastText* approach uses character level information to generate the word vector. The *FastText* method treats *n-gram* of characters of each word and applies the continuous skip-gram method to generate word vectors of the subwords [19]. The pre-trained vector of Punjabi language has been applied for this supervised Punjabi WSD [10]. This pre-trained word vector has been prepared by using CBOW approach of *FastText* method having position-weights. This vector has 300 dimensions. The character *n-grams* of length 5 have been applied to words in window of size 5 with 10 negative samples [10]. It has been trained on the Punjabi Wikipedia and the raw web data fetched by common crawl method.

6. Working of WSD System for Punjabi language

The complete process of disambiguation of Punjabi language ambiguous noun words has been depicted in figure 3. The scope of this WSD is single target ambiguous word in the context. The following data collection, text pre-processing, parsing, feature extraction, and classification steps work in tandem to classify the right sense of an ambiguous word in an instance.

6.1 Data collection

In data collection step the sense tagged data set has been prepared for the 150 ambiguous Punjabi noun words. The selection of ambiguous noun words from Punjabi WordNet, merging of their fine senses to coarse senses and the instance collection for a coarse sense of ambiguous words has been discussed in section 4 of this paper. The sense tagged corpus consists of individual context files for each

Figure 3. Proposed WSD system for Punjabi language.

coarse sense of an ambiguous word. This sense tagged data set has been used to extract the lexical and syntactic features. The semantic word embedding features have been extracted from Wikipedia which is the largest and the most contemporized unlabelled data set of Punjabi language.

6.2 Pre-processing

The sense tagged data set collected in the first step contains a lot of noise (like abbreviations, different fonts, spelling variants). Pre-processing phase converts the abbreviations to normalized text by using a dictionary. Different fonts of the trained data set have been transformed to the standard Unicode font. The spellings variations of the same word are also corrected in this phase by using a standard dictionary. Similarly, the unlabelled Punjabi Wikipedia text has been cleaned by removing web links, punctuations and numeric digits. After eliminating the noise from the sense tagged corpus, the normalized corpus is processed in the parsing phase.

6.3 Parsing

The parsing step extracts the POS tags of the text of sense tagged data set. The Punjabi shallow parser has been applied to generate the different base form and respective POS tags of words in an input instance. The parsing process is subdivided into seven different stages. The first stage of parser is tokenization and it generates the atomic units of the input text. Morphology step determines the root words of an instance. Grammatical classes of text are identified by POS tagger. Chunking stage categorizes a group of words into noun, verb, adjective and adverb phrase. Pruning determines the feature structure of the text. The POS tag of the computed headword to the chunk is assigned by head computation step. The final output is generated by Vibhakti computation of shallow parser [35]. This parsing phase extracts the original words by removing postfixes and prefixes, omit the unnecessary words from the context of the sense files of ambiguous words and determines the respective POS tags.

6.4 Feature extraction

The success of supervised learning classifiers is based on the quality of features extracted from the sense tagged training data set. This step determines the discriminating six lexical features (unigrams, bigrams with WS (2, 3, 4), collocations and co-occurrence) and five syntactic features (combinations of POS tags) extracted for 150 individual Punjabi noun words. The feature files containing string text values are generated in this phase. The distributional semantic features (word embedding) have been generated from the untagged Punjabi corpus. Two shallow NN models

of *word2vec* approach have been used to generate word vectors. These vectors have been used independently for each ambiguous Punjabi noun word for the supervised WSD classification task.

6.5 Classification task

The six classifiers namely DL, DT, NB, K-NN, RF, and SVM have been individually trained on different lexical, syntactic and word embedding features of 150 ambiguous Punjabi noun words. The manually crafted string files of lexical and syntactic features have been converted to binary vector files that indicates the 1 for the presence of word and 0 for its absence. The six classifiers from WEKA (Waikato Environment for Knowledge Analysis) toolkit have been trained on the default configuration based on the empirically obtained values because it tend to perform well [36, 37]. A 10-fold cross validation have applied to reduce the variability of training dataset.

The two deep NN techniques Multi-Layer Perceptron (MLP) and Long Short Term Memory (LSTM) have also been experimented on 150 ambiguous Punjabi noun words [38]. The MLP NN used in this experiment is a feedforward neural network with five neural layers. The LSTM technique is a gated recurrent neural networks which can also propagate data backwards. Both the techniques have been prepared by stacking of linear layers from Python *keras* API [38]. The set of parameters configured for model training are shown in table 11.

Table 11. Parameter configuration of MLP and LSTM techniques.

Parameters	MLP	LSTM
<i>Model initialization</i>		
Hidden layers	3	3
Hidden units	100, 200, 300	Gated memory units 128, 256, 128
Layer Types	Dense Embedded	LSTM Embedded Dense Dropout Flatten
Vector dimension	100/200/300	100/200/300
Weight initialization		Uniform
Activation function	ReLU, Softmax	Sigmoid, tanh, softmax
<i>Model Compilation</i>		
Optimizer	Adam (adaptive moment estimation)	
Loss Function	Categorical cross entropy	
Metrics	Accuracy	
<i>Model Training</i>		
Epoch		25
Batch size		16

The results of the proposed models using the configured parameters are discussed in the next section.

7. Results and discussion

The proposed independent lexical WSD system of Punjabi language has been evaluated using accuracy metric. The accuracy is defined as the correctly estimated instances C of i senses of target word t over the total number of instances n and is given in equation (5).

$$Acc_n^t = \frac{C_i\{i \in \{1, 2, \dots, n\}\}}{n} \quad (5)$$

Moreover, to estimate the robust accuracy of the individual classifier, cross-validation method has been applied. Cross-validation divides the training data set into 10 equal subsets. Repeatedly one subset is used as a test data set while the remaining data sets have been used for training purpose. The mean of 10 different accuracies determined by each classifier forms the final accuracy of each classifier. The average accuracies of disambiguation of 150 Punjabi language ambiguous noun words have been determined for each individual classifier.

The most frequent sense of an ambiguous Punjabi noun word has been considered as the baseline classifier for this

WSD. The average accuracy of the baseline line classifier for 150 ambiguous words is 52.6%.

7.1 Performance of supervised ML techniques based on lexical features

Figure 4 depicts the average accuracies of seven different classifiers (Decision List, Decision Tree, Naive Bayes, K-NN, Random Forest, SVM, MLP) when executed individually on lexical features unigram, bigram with window size 2, 3 and 4, collocations and co-occurrence.

It has been observed that Decision List classifier has produced an accuracy of 80.72% for the bigram feature with window size 3 as shown in figure 4(a). The accuracies of Decision Tree classifier as shown in figure 4(b) are almost same approximately (76%) irrespective of the type of feature except for co-occurrence features set where its accuracy is 73.25%. Naive Bayes classifier has performed exceedingly well (81.1%) with unigram feature (independent features) and collocation feature (80.58%) as indicated in figure 4(c). The average accuracy of classifier K-NN is 79.5%, 82.7% and 78.83% for the bigram features with window size 2, 3 and 4 respectively as shown in figure 4(d). Random Forest classifier has outperformed the other five supervised techniques when evaluated on the bigram features with window size 2, 3 and 4. Figure 4(e) shows the

Figure 4. Accuracies of classifiers using lexical features.

maximum average accuracy of 83.5% attained by Random Forest classifier when the bigram feature with window size 3 is used. Figure 4(f) shows that collocation features are the most (82.43%) suitable while the unigram features are second most (81.4%) suitable for SVM classifier. Figure 4 (g) depicts the predictive performance of MLP NN classifier based on six lexical features. The MLP classifier has outperformed all other classifiers when the collocation feature has been used.

It has been observed that the performance of the classifiers is significantly higher than the average baseline classifier accuracy 52.6%. From the evaluation of classifiers on different lexical features sets, it has been observed that the bigram with window size 2 and 3 are individually most significant parameters to determine the right sense of the ambiguous words of Punjabi language. It has been demonstrated that the increase of the size of the windows for bigrams gradually decreases the overall accuracies of all the classifiers. The performance of the six classifiers slightly drops when the co-occurrence feature set is used. It has been analysed that this is due to the limited sense tagged data set of individual target words. It has been evident from figure 4(g) that the MLP performed exceptionally well on the handcrafted lexical features.

7.2 Performance of supervised ML techniques based on syntactic features

This section determines the accuracies of different classifiers using syntactic features. From figure 5(a), it has been

concluded that the P_0P_{+1} POS combination is most suitable for Decision List classifier with (56.77%) accuracy. Figure 5(b) shows that decision tree classifier has yielded highest accuracy 54.83% and 54.19% for combinations $P_{-1}P_{+1}$ and $P_{-2}P_{-1}P_0P_{+1}P_{+2}$. Naive Bayes (55.48%) is the best performing classifier for the two POS combinations $P_{-1}P_0$ and $P_{-1}P_0P_{+1}$ as shown in figure 5(c). K-NN as shown in figure 5(d) has outperformed other five classifiers with the POS combinations $P_{-1}P_0$ and $P_{-2}P_{-1}P_0P_{+1}P_{+2}$. Figure 5(e) shows the average accuracies attained by Random Forest classifier. SVM classifier performs best on the POS combination $P_{-1}P_0$ as shown in figure 5(f). It has been observed that the $P_{-1}P_0$ and P_0P_{+1} POS combinations, i.e., a word prior to and a word after the target word are the most indicative combinations for identification of sense of an ambiguous Punjabi language word. It has been concluded that inadequate sense tagged data set and small syntactic combinations can be the reason for poor results of the classifiers using syntactic feature sets.

7.3 Performance of ML classifiers using word embedding

The predictive performance of the ML classifiers using word embedding features is discussed in this section. Two NN based *word2vec* CBOW and SG models have been applied on same unlabelled data set of Punjabi language to generate 100, 200 and 300 dimensional word vectors. The predictive performance of two deep NN based classifiers MLP and LSTM using word vectors is given in table 12.

Figure 5. Accuracies of classifiers using syntactic features.

Table 12. Average accuracies obtained by two deep learning supervised classifiers using word embedding features.

	CBOW			SG		
Dim*	100	200	300	100	200	300
MLP	78.89	79.99	76.46	83.15	80.55	82.1
LSTM	79.52	79.85	80.45	83.45	82.5	84.15

*Dimensions

Table 13. Average accuracies attained by six supervised classifiers by combining word embedding features with Concatenation (Con.) and Average (Avg.) strategies.

	word2vec SG					
Dim*	100	100	200	200	300	300
Com ~	Con.	Avg.	Con.	Avg.	Con.	Avg.
DL	74	75.68	77.85	78.62	79.87	80.76
DT	78	77.84	81.3	80.15	79.56	80.45
NB	76.59	77.6	78	79.8	77.84	79.6
KNN	79.56	80.1	80.4	81.15	80.15	81.46
RF	77.89	78.35	78.5	79.45	83.65	80.51
SVM	78.56	79.86	77.65	79.56	82.1	83.75

*Dimensions, ~Combinations

Table 14. Average classification accuracies obtained by two NN classifiers and pre-trained 300 dimensional *FastText* word embedding.

Classifiers	Accuracy
MLP	82.5
LSTM	83.45

The highlighted cells in table 12 shows the best average accuracies achieved by deep NN classifiers.

It has been observed that the word embedding features produced by SG model of *word2vec* are more suitable for small data sets of Punjabi language. The LSTM classifier has yielded the best performance (84.15%) using 300 dimensional SG *word2vec* word embedding. The performance of traditional ML classifiers has been estimated by using embedding features. The word vectors have been combined with bag of word features using concatenation and average combination strategies. The average predictive performance of the individual classifier using six different word embedding features on 150 Punjabi noun words is given in table 13. The highlighted cells in table 13 shows the best average accuracies yielded by ML classifiers. It has been observed that the performance of the WSD system based on traditional ML classifiers has enhanced with the application of the word embedding techniques.

Table 13 shows that the performance of classifiers using average combination strategy is higher than the

Table 15. Statistical significance t-test between LSTM and SVM using word vector features.

	LSTM	SVM
Mean	84.149	83.750
Variance	21.192	17.434
Observations	152	152
Pooled variance	19.313	
Degree of freedom	302	
t Stat	0.792	
P(T≤t) two-tail	0.428

Figure 6. Average accuracies of Random Forest and LSTM *w.r.t* #senses of ambiguous words. **Figure 7.** Accuracies of RF and LSTM using *w.r.t* #instances of ambiguous words having coarse sense 6.

concatenation combination strategy. The word vector with 300 dimensions is better than the other two dimensions for this classification task.

The Punjabi WSD system has been further evaluated on the pre-trained 300-dimensional embedding. The average accuracies of Punjabi WSD systems for individual 150 ambiguous words using two deep learning techniques MLP and LSTM are shown in table 14. It has been observed that the average accuracy of LSTM classifier using pre-trained *FastText* word embedding is almost similar to its accuracy yielded by it using *word2vec* SG word embedding.

The significance test of the LSTM and SVM models have been performed by using SG 300 dimensional word vector

Table 16. Characteristics of different Punjabi WSD systems.

WSD System	Sense tagged Corpus	Number of ambiguous words	WSD method	Techniques applied	Evaluation metric
Rana and Kumar [24]	WordNet	30	Knowledge based	Overlap Lesk	Accuracy
Singh and Singh [25]	1549 instances	15	Knowledge based	Modified Overlap Lesk	Precision
Proposed System	22153 instances	150	Supervised	Supervised ML and deep learning classifiers	Accuracy

features. The results of this significance test has been reported in table 15. From this test, it has been observed that there is no significant difference between the results achieved by SVM and LSTM models using SG word vectors.

From the analysis of the results, it has further been observed that lexical features (collocations, bigrams) and word embedding features are more significant than the syntactic features.

7.4 Error analysis of the proposed system

The major reasons for errors in the proposed system are due to a large number of senses of ambiguous words and the small size of sense tagged data set. The sense data set is skewed towards the most frequent sense of the ambiguous word. The results of supervised machine learning techniques reveal that the accuracies of the proposed systems are higher for the ambiguous words with lesser number of senses. While the performance of the system degrades when the target words have higher number of senses. In order to analyze the impact of large number of senses on the accuracies of classifiers figure 6 has been depicted. Figure 6 shows accuracies of two best performing classifiers Random Forest using Bigram WS 3 feature and NN based LSTM model using *word2vec* SG word embedding for 150 ambiguous words. The x-axis of figure 6 shows the senses of Punjabi ambiguous words ranging from 2 to 8. The left y-axis depicts the number of words belonging to different senses corresponding to the x-axis. It can be observed from figure 6 that the average accuracies of both the classifiers degrade as the senses of the ambiguous words increase.

The effect of data set size on the performance of the WSD system using Random Forest and LSTM has been depicted in figure 7. The 15 most ambiguous words having coarse sense 6 are represented on the x-axis and their respective number of instances are represented on the y-axis of figure 7. It has been observed that the increase in the size of data set of target ambiguous words significantly improves the accuracies of both the classifier as shown in figure 7.

It can be observed that the accuracies of different classifiers used in this paper decline for the words having small

data set size. The large number of sense of the ambiguous words also have negative impact on the performance of different classifiers. The lower accuracies of classifiers on feature sets like co-occurrence, unigram shows that the single context word around the ambiguous word is not enough for its disambiguation. Accuracies of syntactic features of individual ambiguous words do not divulge much about the sense of the ambiguous words. The overlapping features of the words among different senses are the major cause of accuracy degradation of almost all the classifiers.

7.5 Comparison of WSD system of Punjabi language

The comparison of proposed supervised Punjabi WSD with the two knowledge based Punjabi WSD systems have been shown in table 16. To the best of our knowledge, no supervised Punjabi WSD system has been reported. There exist no common data sets for evaluation of Punjabi WSD systems. The parameters that are considered for comparison are sense tagged corpus size, number of ambiguous words, techniques used and evaluation metrics.

It has been observed from table 16 that the proposed system uses a substantially large sense tagged corpus. Both the knowledge-based systems have applied simple overlap techniques on the limited number of ambiguous words.

While the proposed system has been evaluated on 150 ambiguous words using supervised machine learning and deep learning techniques.

8. Conclusions and future scope

In this paper, the WSD system for Punjabi language has been proposed and analyzed. This proposed system has been tested on manually sense tagged corpus of 150 most ambiguous Punjabi language noun words, using six machine learning classifiers. Two deep learning MLP and LSTM techniques have also been applied to the same sense tagged data set. The lexical and syntactical features have been used individually in this experiment. The word embedding features have been extracted from the Punjabi

unlabeled text and applied on six machine learning and two deep machine learning classifiers.

The pre-trained word vectors have shown an enhanced predictive performance of the supervised classifiers. The LSTM deep learning classifier has outperformed the other classifiers while using the word embedding features.

In the future, scalability of the proposed system can be improved by adding more ambiguous words of the Punjabi language. This system is one target word WSD and can be extended to all-word WSD system. The performance of deep learning techniques like Bi-directional LSTM (BLSTM), recurrent neural network and convolutional network can be evaluated on Punjabi WSD. The ensemble of classifiers and feature sets can be used to further improve the performance of this WSD system. In the future, this proposed WSD system for Punjabi can be integrated with various Punjabi NLP applications to improve their respective overall performances.

Acknowledgements

This Publication is an outcome of the R&D work undertaken in the project under the Visvesvaraya PhD Scheme of Ministry of Electronics & Information Technology, Government of India, being implemented by Digital India Corporation (formerly Media Lab Asia).

References

- [1] Navigli R 2009 Word sense disambiguation: A survey. *ACM Comput. Surv.* 41(2): 1–69
- [2] Bhala R V and Abirami S 2014 Trends in word sense disambiguation. *Artif. Intell. Rev.* 42(2): 159–171
- [3] Agirre E, de Lacalle OL, Soroa A 2014 Random walks for knowledge-based word sense disambiguation. *Comput. Linguist.* 40(1): 57–84
- [4] Agirre E and Edmonds P eds. 2007 *Word sense disambiguation: Algorithms and applications*, vol. 33, Springer Science & Business Media, New York
- [5] Zhong Z and Ng H T 2010 It makes sense: a wide-coverage word sense disambiguation system for free text. In: *Proceedings of the 48th ACL*, Uppsala, Sweden, pp. 78–83
- [6] Bengio Y, Ducharme R, Vincent P and Janvin C 2003 A neural probabilistic language model. *J. Mach. Learn. Res.* 3:1137–1155
- [7] Iacobacci I, Pilehvar M T and Navigli R 2016 Embeddings for word sense disambiguation: An evaluation study. In: *Proceedings of the 54th annual meeting of the ACL*, vol 1, Berlin, Germany, pp. 897–907
- [8] Narang A, Sharma R and Kumar P 2013 Development of Punjabi WordNet. *CSI Trans ICT* 1(4): 349–354
- [9] Mikolov T, Le Q V and Sutskever I 2013 Exploiting similarities among languages for machine translation. arXiv preprint [arXiv:1309.4168](https://arxiv.org/abs/1309.4168)
- [10] Grave E, Bojanowski P, Gupta P, Joulin A and Mikolov T 2018 Learning word vectors for 157 languages. arXiv preprint [arXiv:1802.06893](https://arxiv.org/abs/1802.06893)
- [11] Fellbaum C 1998 *WordNet*. Blackwell Publishing Ltd.
- [12] Narouei M, Ahmadi M and Sami A 2015 SePaS: Word sense disambiguation by sequential patterns in sentences. *Nat. Lang. Eng.* 21(2): 251–269
- [13] Mohammad S and Pedersen T 2004 Combining lexical and syntactic features for supervised word sense disambiguation. In: *CoNLL*, Boston, USA, pp. 25–32
- [14] Manning C D 2016 Computational linguistics and deep learning. *Computational Linguistics*, pp. 1–7
- [15] Yarowsky D and Florian R 2002 Evaluating sense disambiguation across diverse parameter spaces. *Nat. Lang. Eng.* 8(4): 293–310
- [16] Ronan Collobert and Jason Weston 2008 A unified architecture for natural language processing: Deep neural networks with multitask learning. In: *Proceedings of the 25th ICML*, Helsinki, Finland, pp. 160–167
- [17] Trask A, Michalak P and Liu J 2015 Sense2vec-A fast and accurate method for word sense disambiguation in neural word embeddings. arXiv preprint [arXiv:1511.06388](https://arxiv.org/abs/1511.06388), pp. 1–9
- [18] Kageback M and Salomonsson H 2016 Word Sense Disambiguation using a Bidirectional LSTM. CoRR [arXiv:1606.03568](https://arxiv.org/abs/1606.03568), pp. 1–6
- [19] Mikolov T, Grave E, Bojanowski P, Puhersch C and Joulin A 2017 Advances in pre-training distributed word representations. arXiv preprint [arXiv:1712.09405](https://arxiv.org/abs/1712.09405)
- [20] Popov A 2018 Neural network models for word sense disambiguation: an overview. *Cybern. Inf. Technol.* 18(1): 139–151
- [21] Gupta S, Namavari A and Smith T O 2016 Word sense disambiguation using skip-gram and LSTM models. *stanford.edu*, pp. 1–9
- [22] Yuan D, Richardson J, Doherty R, Evans C and Altendorf E 2016 Semi-supervised word sense disambiguation with neural models. arXiv preprint [arXiv:1603.07012](https://arxiv.org/abs/1603.07012), pp. 1–12
- [23] Raganato A, Bovi C D and Navigli R 2017 Neural sequence learning models for word sense disambiguation. In: *Proceedings of the 2017 conference on empirical methods in natural language processing*, Copenhagen, Denmark, pp. 1156–1167
- [24] Rana P and Kumar P 2015 Word sense disambiguation for punjabi language using overlap based approach. In: *Advances in Intelligent Informatics*, pp. 607–619. Springer International Publishing
- [25] Singh J and Singh I 2015 Word sense disambiguation: enhanced Lesk approach in Punjabi language. *Int. J. Comput. Appl. (0975-8887)* 129(6): 23–27
- [26] Josan G S and Lehal G S 2008 Size of N for word sense disambiguation using N-gram model for Punjabi language. *Int. J. Transl.* 20(1): 47–56
- [27] Saini T S and Lehal G S 2011 Word Disambiguation in Sharmukhi to Gurmukhi transliteration. *Asian Language Resources collocated with IJCNLP*, Chiang Mai, Thailand, pp. 79–87
- [28] Kachru B, Kachru Y and Sridhar S 2008 *Language in South Asia*. Cambridge University Press. <https://books.google.co.in/books?id=O2n4sFGDEMYC>
- [29] Kumar P and Sharma R K 2012 Punjabi to UNL enconversion system. *Springer Sadhana Acad. Proc. Eng. Sci.* 37(2): 299–318
- [30] Chan Y S, Ng H T and Chiang D 2007 Word sense disambiguation improves statistical machine translation. In:

- Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic, pp. 33–40
- [31] Bhattacharya P 2010 IndoWordNet. In: *Proceeding of the Lexical Resources Engineering Conference*, Malta, pp. 1–8
 - [32] Palmer M, Dang H T and Fellbaum C 2007 Making fine-grained and coarse-grained sense distinctions, both manually and automatically. *Nat. Lang. Eng.* 13(2): 137–163
 - [33] Rong X 2014 word2vec parameter learning explained. arXiv preprint [arXiv:1411.2738](https://arxiv.org/abs/1411.2738)
 - [34] Rehurek R and Sojka P 2010 Software framework for topic modelling with large corpora. In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pp. 45–50
 - [35] Kumar P 2012 UNL Based Machine Translation System for Punjabi Language. Ph.D. dissertation, CSED, TU, Patiala, Punjab, India
 - [36] Witten I H and Frank E 2005 *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, USA
 - [37] Amancio D R, Comin C H, Casanova D, Travieso G, Bruno O M, Rodrigues F A and da Fontoura Costa L 2014 A systematic comparison of supervised classifiers. *PloS one*. 9(4): e94137
 - [38] Chollet F 2015 keras software library. GitHub. <https://github.com/fchollet/keras>