Multimedia Tools and Applications
https://doi.org/10.1007/s11042-023-14676-8

Lexeme connexion measure of cohesive lexical ambiguity revealing factor: a robust approach for word sense disambiguation of Bengali text

Debapratim Das Dawn[1] · **Abhinandan Khan**[1,2] · **Soharab Hossain Shaikh**[3] · **Rajat Kumar Pal**[1]

Received: 17 November 2021 / Revised: 30 October 2022 / Accepted: 3 February 2023

Abstract

Word sense disambiguation (WSD) is the process of finding out the appropriate meaning of a polysemous word based on any given context. The Bengali language inherently comprises a large number of polysemous words. Recently, researchers in the domain of linguistics have been attracted to the problem of WSD in Bengali text due to its numerous interesting applications, viz. machine translation, opinion polarity identification, question-answering systems, etc. In this paper, lexeme connexion measure of cohesive lexical ambiguity revealing factor has been proposed that takes a decision on the disambiguation of senses of a Bengali polysemous word. All the polysemous words have been treated as *target* words, and a context window of three different sizes, viz. five, seven, and ten are considered based on these target words. This paper has generated lexeme harmony measure for quantifying heuristically of syntactic belongings of a collection of lexemes in Bengali text. The proposed methodology has been extracted a feature vector by considering the *cohesive lexical ambiguity revealing factor* or CLARF, depending on *frame lexeme harmony* (FLH), *sense lexeme harmony* (SLH), *polysemy singularity coherence* (PSC), *polysemy distribution factor* (PDF), and *relative polysemy singularity coherence* (RPSC) factor of a lexeme. This Bengali WSD technique has been applied max-rule of integrated *lexeme connexion measure* (LCM) of each lexeme of both the testing and training cases score for sense recognition. The proposed algorithm has succeeded in eliminating the drawback of the Bengali WSD approaches, as it can focus on both the lexical and semantic relationships between words. The performance of this algorithm has been evaluated on a dataset that consists of 100 polysemous words of three/four senses. Various evaluation metrics have been used to analyse the results obtained by the proposed algorithm. The obtained results indicate the robustness of the proposed algorithm.

Keywords Word sense disambiguation · WSD of resource scaring languages · WSD of Indian languages · Polysemous word · Sense identification

✉ Debapratim Das Dawn
debapratimdd@gmail.com

Extended author information available on the last page of the article.

1 Introduction

Computer science researchers working in the field of linguistics have started active research in *word sense disambiguation* or WSD since the late 1940s [2]. WSD is an AI-complete problem of selecting the correct sense of a word from the set of predetermined possibilities [35]. A WSD system performs two vital tasks, viz.

– searching for polysemous words for a given context, and
– assigning the obtained polysemous words with the most appropriate sense in the relevant context.

Consequently, the performance of any WSD algorithm is evaluated using various datasets consisting of multiple polysemous words. The frequency count of these polysemous words and some keywords play a vital role in creating a new database in WSD. The theory of "Law of Meaning" (proposed in 1949 [68]) states that there is a relationship between the rate of occurrence and the number of senses of a word. According to this theory, the polysemy property of a word varies with the number of its occurrences in any given context. A high-frequency word has more number of senses associated with it compared to a low-frequency word. This alludes to semantic vividness between frequency of a word and its number of senses.

WSD has now become open to discussion at the lexical level in the domain of *natural language processing* (NLP). A number of algorithms have been proposed in this domain covering various notable languages, like English [35], Chinese [9], Japanese [34], Spanish [58], Dutch [22], Italian [19], etc. Researchers have generally used two variations of the generic WSD problem, such as:

– *Target-word WSD*: restricting to a *target* word, occurring once per sentence or paragraph.
– *All-word WSD*: applicable for all one-class words.

Besides English, the recognition accuracy of the other above-mentioned languages is also quite good. Along with English and other European languages, the WSD has also been performed in various well-known Indian languages, such as, Hindi [57], Marathi [70], Malayalam [62], Telugu [46], Nepali [17], Manipuri [59], Punjabi [51], Kannada [48], Tamil [3], etc. However, investigations on WSD in various Indian languages has not achieved much success due to several reasons. One of the prominent reason is morphological complexity that depends on the number of inflected forms of a word present in the context. The processing of inflected forms of a word is so difficult that there is no standard database for Indian languages, especially the Bengali language. This lack of resources is further enhanced due to the absence of standard disambiguation techniques, in contrast with other frequently-spoken languages. Nonetheless, few research works in the domain of Bengali WSD has been published [13, 47].

The Bengali language has evolved a vast body of literature with an extensive collection of homonyms and polysemous words. Bengali is the fifth most-spoken language in the world, and the mother tongue of more than 280 million people [47]. Sentence formation in this language comprises monosemous and polysemous words, guided by Bengali grammatical rules. Over the last decade, few WSD techniques have been proposed for disambiguating the senses of polysemous Bengali words [47]. A generic Bengali WSD system, as shown in Fig. 1, consists of six major steps, which are as follows:

(a) *Pre-processing*: Identification of polysemous words in a given context.

(b) *Normalisation*: Equalisation of font sizes and removal of punctuations, brackets, extra spaces, slashes, etc.
(c) *Lemmatisation*: Determines the root word from the inflected forms by cutting off the end or beginning portion of a word. This step also performs by tagging the part of speech for each word and applying normalisation rules according to a given part of speech [25].
(d) *Feature Set Creation*: Algorithmic operations for extraction and collection of feature set.
(e) *Classification*: Find the different senses.
(f) *Recognition*: Break the ambiguity.

This work is motivated by the importance of Bengali language over other most spoken languages. It is the official language of 280 million people, and is the fifth most spoken language in the world [15]. A little research has been done till date and several problems are yet to be solved in the context pf WSD. Due to the high degree of morphological variation of prefixes, inflectional suffixes and derivational suffixes, polysemy resolution is very difficult in Bengali texts. There are pragmatic applications of a Bengali WSD system includes the following areas:

(a) *Machine Translation*: It is one of the finest applications of any resource-scaring language like Bengali. The translation of a Bengali polysemous word into any language varies according to its senses. The senses of a polysemous words are rendered according to the training data [13].
(b) *Opinion Polarity Identification*: It identifies the opinion of phrases, i.e., positive (i.e., like) or negative (i.e., dislike), from textual data. It includes various features, like chunk-level information, functional words, stemming clusters, negative words, etc. The organisation of the decision boundary for a polysemous word with multiple senses is a very challenging task in identifying opinion polarity [12].
(c) *Bengali Question Answering (QA) System*: A QA system automatically gives an answer to the question in a brief manner like a human being. It consists of enigma inspection, sentence extraction, and answer extraction. Named Entity Identification (NEI) for an ambiguous word plays a pivotal role in giving the exact answer [5].
(d) *Bengali Parts of Speech (POS) Tagging*: It deals with the labelling of each word according to the proper Bengali grammatical rules. It is helpful for the pre-processing

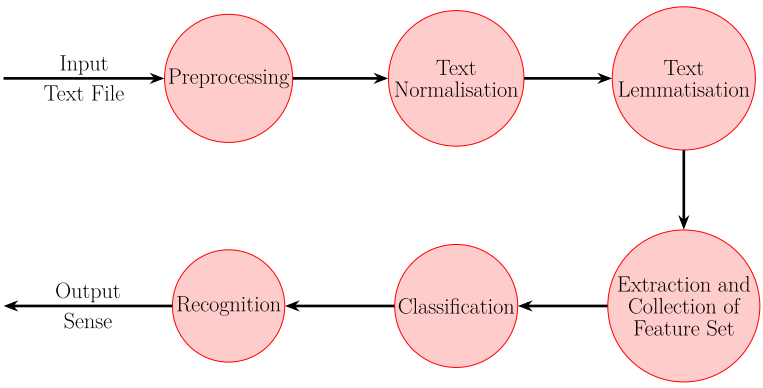

Fig. 1 Flow diagram of a generic Bengali word sense disambiguation system

step of language processing. The POS tagging for ambiguous words is turning into a challenging issue [18].

(e) *Emotion-Level Tagging for Bengali Blogs*: It assists in assigning emotion tags to Bengali blog data. The confusion matrix indicates the presence of ambiguous words. Introductory sentences of blogs, question words, special punctuation symbols, and negative words are taken into account on this domain [10].

(f) *Subjectivity Detection*: It classifies the text as either objective or subjective. It extracts the sensitive content of the text using labelled corpora and emotion lexicon. The ambiguous words, unambiguous words, and entries are used as resource [11].

Finally, to summarise, we have described the fourteen-fold contribution and novelty of the proposed work as follows:

1. This work presents a robust approach for sense disambiguation. All the polysemous words have been treated as *target* words, and a context window of three different sizes, viz. five, seven, and ten are considered based on these target words.
2. This paper introduces six feature extraction tools, such as:
   - *frame lexeme harmony* (FLH),
   - *sense lexeme harmony* (SLH),
   - *polysemy singularity coherence* (PSC),
   - *polysemy distribution factor* (PDF),
   - *relative polysemy singularity coherence* (RPSC), and
   - *cohesive lexical ambiguity revealing factor* (CLARF).
3. FLH has been generated to measure the syntactic belongings of a collection of lexemes heuristically.
4. SLH has been generated based on the number of senses of a polysemous word.
5. PSC has been set up to compute the rareness or uniqueness of a lexeme.
6. PDF has been generated to estimate the prominence of a lexeme in a text document collection.
7. RPSC has been produced for capturing all-inclusive term discrimination value.
8. CLARF has been created for combining both local and global feature types.
9. Sense lexeme in
   - training sets for $|S_j| = 3$,
   - training sets for $|S_j| = 4$, and
   - testing sets

   has been created.
10. Individual lexeme connexion measure of each lexeme has been estimated for both testing and training cases.
11. Integrated lexeme connexion measure has been calculated with respect to each sense of a polysemous word.
12. The proper sense of the testing data has been recognised by applying max-rule of integrated lexeme connexion measure score.
13. Experimentation has been done on the basis of precision, recall, F-measure, accuracy, and misclassification rate of one hundred polysemous words.
14. The robustness of the proposed method was proved using linear regression method.

The rest of the paper has been organised as follows. Section 2 presents a comprehensive review of the literature associated with the classification of Bengali WSD approaches.

Section 3 explains the proposed WSD technique in detail. Section 4 presents a dataset of 100 Bengali polysemous word. Section 5 validates the performance of the proposed algorithm with the dataset presented in Section 4. Section 6 presents an exhaustive analysis of the obtained results from various perspectives. Section 7 concludes this work, providing a glimpse of some future directions of the work.

2 Related works

The research on Bengali WSD techniques is a remarkable open problem at the lexical level of the language processing task. The number of productive research works on Bengali WSD techniques is very less compared to that of other oft-spoken languages. Presently, research in this domain is at the elementary level. Nonetheless, few researchers have proposed Bengali WSD techniques. We have briefly discussed some of this works in this section. The various approaches for WSD can be broadly divided into three categories, knowledge based, machine learning, and hybrid models.

The knowledge or dictionary based approach consists of relations between the words present in a database. The performance accuracy of this approach is admirable. This approach uses various knowledge sources, like wordbooks, thesauri, machine-readable dictionaries or MRDs, semantic networks, and lexical knowledge bases [36]. Various notable algorithms, like the Lesk algorithm, semantic similarity, selection preferences, and heuristic methods are used in this approach. The Lesk algorithm counts the frequency of each word separately and allocates exact sense of the polysemous word on the basis of the highest rate of repetition [26]. The semantic similarity measure is responsible for identifying the degree of semantic relationship between the two words [52]. The selection preferences count the occurrence of the word pairs in a syntactic relation of word types [53]. The heuristic methods use the most frequent sense; one sense per discourse and one sense per collocation for evaluating different linguistic properties of word sense [37]. Nevertheless, the performance of dictionary based approaches depend on dictionary definitions and suffer from overlap sparsity.

The machine learning or corpus based techniques consist of a set of statistical patterns for identifying the key aspect of the textual data [27]. This approach can further be classified into three broad categories, such as supervised, semi-supervised, and unsupervised. The supervised learning based model requires input data from manually created sense-annotated datasets for learning from the training data. It follows various algorithms, like decision tree, decision list, Naïve Bayes, neural networks, *support vector machine* (SVM), exemplary or instance based learning, etc. [28]. It has been observed from performance analysis that the output of this approach is better compared to unsupervised learning and knowledge based approaches.

Unsupervised learning based approaches use unannotated corpora for searching the word meaning. This approach follows various algorithms, like word clustering, context clustering, co-occurrence graph, spanning tree based approach, etc. The sense annotated dataset and sense inventory is not useful in this regard [50]. Nevertheless, it is very problematic to implement in the case of any resource-scaring language.

Semi-supervised or minimally-supervised learning based models deal with both annotated and unannotated data. This approach is thus an amalgamation of both supervised and unsupervised approaches. The classification strategy of this approach depends solely on input information [65]. Nonetheless, it suffers from the knowledge acquisition bottleneck.

The hybrid approach, on the other hand, builds up by incorporating the idea of knowledge and corpus based technique. This approach identifies semantic relation using corpus evidence [38]. The performance of this approach is also quite good. The categorisation of WSD approaches have been shown in Fig. 2. However, when we only talk about Bengali WSD method, only three types are available [6], viz.

2.1 Knowledge based Bengali word sense disambiguation techniques

Haque et al. [21] presented a notable work on Bengali WSD for dissolving the ambiguity of lexical semantics by following a knowledge/dictionary based approach, which has two stages, parsing and detection. In the parsing stage, the proposed methodology identified all the polysemous words inside the given context and identified a word as polysemous if it has multiple sense definitions according to the dictionary used. The proposed technique built a relationship pair between the polysemous words and its corresponding neighbourhood words for the generation of the parse tree. In the detection phase, the proposed algorithm achieved an overall accuracy of 82.4%. Haque et al. [21] implemented their approach only in the case of the noun, adjective, and verb based polysemous words. Thus, experimental results could not be analysed for the remaining parts of speech. Also, the proposed algorithm cannot capture the semantic feature of a sentence.

Pal et al. [45] disambiguated the senses of polysemous words using the Bengali dictionary. The authors followed the Lesk algorithm [26] for resolving the ambiguity. It achieved a very low precision rate (31%). Consequently, the authors extended the previous baseline process, for magnification of the overlapping area of senses of the polysemous words, by enlarging the window size for the collocating words only. Here, the authors used a Bengali lemmatisation tool for finding the root word and simplifying the text processing step, and this resulted in a 75% accuracy for this second stage. However, the authors did not consider polysemous collocating words, and thus, excluded such words from the lexical overlap.

2.2 Supervised learning based Bengali word sense disambiguation techniques

Pandit et al. [47] proposed a supervised learning based model for sense disambiguation in the test set. The proposed methodology used separate training and testing datasets, and followed the k-Nearest Neighbour or k-NN algorithm for the grouping of new text samples. k-NN can classify a new sample from its k nearest neighbours using Euclidean distance,

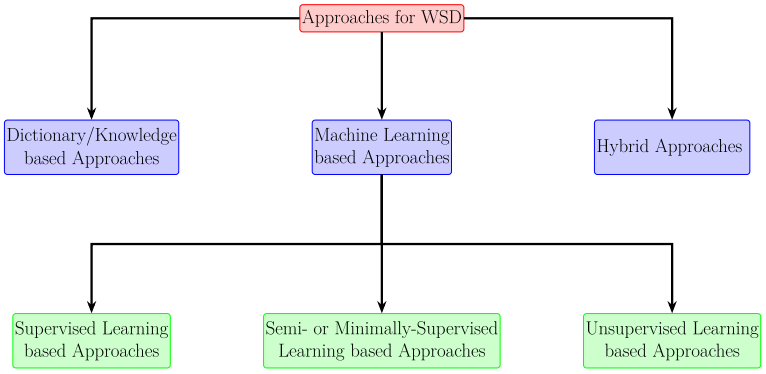

Fig. 2 Classification of word sense disambiguation approaches

Hamming distance, Manhattan distance, etc. [1]. However, the processing speed of k-NN is very slow for large databases, and as such, it is known as a *lazy learner*.

Pal et al. [41] proposed an automatic classification technique for Bengali annotated text. The proposed framework learnt a dataset from the raw text by annotating the text files and removing the stop words. The authors prepared two separate sets of training and testing data. The training dataset helped to classify the text from unknown samples. The authors used the Naïve Bayes probabilistic model [29] for learning from the annotated data. However, the authors only considered the noun words as polysemous words and did not use the other parts of speech.

Pal et al. [43] extended their work [41] for improving the obtained results. The authors performed text lemmatisation here before processing the text and collected the related corpus from a dataset for feature extraction. The proposed algorithm also used the Naïve Bayes classification model [29]. The Bengali text corpus developed in the *Technology Development for Indian Languages* (TDIL) project of the *Government of India* (GoI) has been used here as a dataset. This dataset has the following two types of input data: (*i*) regular input data without lemmatisation, and (*ii*) lemmatised data.

However, the functioning area of this algorithm is limited to a small database. Thus, a generalized algorithm is required for handling polysemous words of the various parts of speech. Additionally, Pal et al. [42] used Naïve Bayes classification model in 2019. The performance of this approach is better compare to decision tree, SVM, and artificial intelligence network. However, this approach is also suffered from information scarcity problem in WordNet. It is unable to detect sense-based contextual-similarity problem. In 2021, Biswas et al. performed supervised based Bengali WSD [6]. However, Bayes probability theorem does not work well if the number of senses in a polysemous word is increased.

2.3 Unsupervised learning based Bengali word sense disambiguation techniques

Das et al. [13] focused on Bengali to Hindi machine translation by implementing unsupervised learning based WSD technique for Bengali. The proposed algorithm used the correct lexical choice in Bengali to Hindi machine translation. The authors generated a co-occurrence graph after extracting the neighbouring context containing the target word. A co-occurrence graph consists of sets of vertices and edges. Each vertex represents a word in the corpus and an edge exists between the two vertices if the two words co-occur in a sentence or a context. The weight of an edge is equal to the number of contexts in which the two words co-occur. Edge-density of the graph is captured for identifying the clusters within the graph. Nonetheless, the proposed algorithm did not follow any general rule for word clustering. Hence, the result varies by a large magnitude for different datasets.

Pal et al. implemented sentence clustering using the maximum entropy method by considering the finest instances of a probability distribution of the present state of knowledge [40]. Human interference is required to attach a label to the relevant senses after sentence clustering. The sentence clustering is useful for categorising the sentences into a number of collections. This paper implemented *principal component analysis* (PCA) across the characteristic vector and context enlargement of sentences using Bengali WordNet (developed at the *Indian Statistical Institute* (ISI), Kolkata). However, PCA has some biasses concerning the variable feature set. In case of scale-down or scale-up of context from sentence-level to document-level, or vice versa, PCA generated high-variance outcomes. Subsequently, the intra-class relationship among the feature set were not developed properly and the accuracy was thus hampered.

Sengupta et al. performed *word sense induction* (WSI) without including a sense inventory [56]. The proposed technique followed distributional semantics using a context clustering approach. Each existence of a target word in a textual corpus was characterised as a context vector. The dataset consisted of fifteen polysemous words of two senses with nine sentences in each category. The authors used K-means clustering and achieved a better outcome. However, they tested their proposed algorithm on a very small-scale dataset, consisting of the average number of two senses.

Subsequently, the researchers working in this domain have attempted to develop various Bengali WSD technique. Most of the researchers have followed the supervised learning based approach. The supervised learning based models are very useful for any resource scaring and morphologically complex language, especially, like Bengali. Due to its strategical perspective of human-guidance, this model is most suitable for Bengali WSD. Furthermore, the performance of knowledge based techniques is also good compared to unsupervised learning based approaches.

In addition, most of the reported authors have tested their algorithms in normal or plain datasets, which comprise most of the useful features. However, the processing task may increase if one or more of the following complicated issues are taken care of:

- various grammatical structures present in a sentence,
- incorrect syntactic structure,
- less semantic information,
- large sentential form with insignificant contextual words,
- short sentences without enough information,
- spelling mistakes,
- adding extra non-functional words, etc.

2.4 Constraints in Bengali WSD

We have highlighted the following limitations of the present work, such as:

- The following three forms of morpheme cause a high degree of morphological variation of the words in the text:
  (a) prefix,
  (b) derivational suffix, and
  (c) inflectional suffix.
- Root words are difficult to identify due to the absence of proper word stemming or lemmatisation algorithms.
- The following two styles in the text make it difficult to process সাধু ভাষা (*chaste language*), such as:
  (a) তৎসম ($Ta\underline{t}sama$), and
  (b) তৎভব ($Ta\underline{t}bhaba$)
- It is very difficult to identify whether a word is named entity or not, because there is no concept of uppercase lowercase in Bengali.
- It is very difficult to compute the শব্দের ব্যুৎপত্তি (*etymology of word*) and সমাস (*compound word*) from the text, because most of the words in Bengali are made from Sanskrit.
- A significant change in ক্রিয়াবাচক বিশেষ্য (*adverbial noun*) and ক্রিয়াবাচক বিশেষণ বিশেষণ (*adverbial adjective*) makes it difficult to process Bengali text computationally.

3 Proposed methodology

Word sense disambiguation is an attractive but challenging proposition in the field of natural language processing, especially for any resource scaring language like Bengali. The Bengali corpora contain semantic relations (e.g., hypernymy, hyponymy, holonymy, etc.), lexical relations (e.g., synonymy, antonymy, etc.), and linked structure features. The linguistic complexity of Bengali is very high [63]. The Bengali language contains a huge number of inflected words. Obtaining the root word from an inflected word is very difficult in this language [44]. The linguistic structure of Bengali is very similar to Sanskrit. The basic pattern of this language is subject + object + verb (SOV). The auxiliary verb set is not present here, unlike English [23]. Infinite verbs act as auxiliary verbs in Bengali; e.g., -এ থাকবে $(\text{-}\bar{E}\ thākabē)$ in ক'রে থাকবে $(Ka'rē\ thākabē)$ (*He/she might have done it*) is an auxiliary.

Moreover, unlike in English, prepositions are placed after the noun or pronoun in Bengali, making them *post-positions*. There is no concept of capital letters and small letters in Bengali. Therefore, Bengali text does not provide any capitalisation information to indicate the beginning of a sentence, like English and most spoken European languages [16]. It is very difficult to understand whether a word is a naming word or a polysemous word or something else altogether, and to create a word boundary for the target word.

Let us consider the example of রবি (*Rabi*: *Rabi*), which happens to be the name of a renowned person. However, it can also be treated as a polysemous word because it has multiple meanings, like সূর্য $(Sūrya$: $Sun)$, সপ্তাহের প্রথম দিন $(Saptāhēra\ prathama\ dina$: $The\ first\ day\ of\ week)$, গম, যব,প্রভৃতি বসন্তকালীন শস্য $(Gama,\ yaba,\ prabhrti\ basantakālīna\ śasya$: $spring\ crops\ like\ wheat,\ barley,\ etc.)$ রবীন্দ্রনাথ ঠাকুরের সাহিত্যরীতি বা চিন্তাধারার অনুসরণ $(Rabīndranātha\ ṭhākurēra\ sāhityarīti\ bā\ cintādhārāra\ anusaraṇa$: *following Rabindranath Tagore's literary style*), etc. Consequently, *named entity disambiguation* (NED) may exist in this context in Bengali. Thus, it becomes a really challenging task to make a decision on the length of the frame boundary of the corresponding target word. A frame consists of a target word and its neighbouring words. It defines the context window for each target word. The target word is a member of a predefined set of polysemous words. The proposed methodology follows a supervised learning based approach, and is based on the CLARF score and max-rule of integrated LCM score.

The proposed methodology consists of five elements, (*i*) text pre-processing, (*ii*) collection of feature set, (*iii*) extraction of features using the CLARF score of each lexeme, (*iv*) computing integrated lexeme connexion measure from testing set to training set, and (*v*) recognition of testing data using max-rule of LCM score. The entire disambiguation process can be divided into two phases, namely, the *training* phase and the *testing* phase. The training phase comprises three steps, which have been pictorially represented in Fig. 3. The

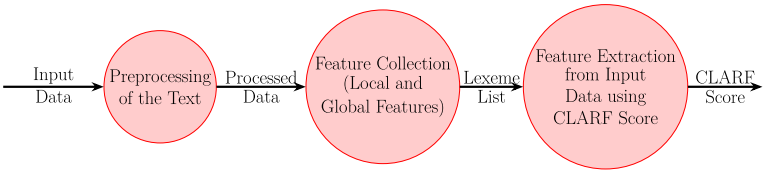

Fig. 3 Flow diagram of the training phase

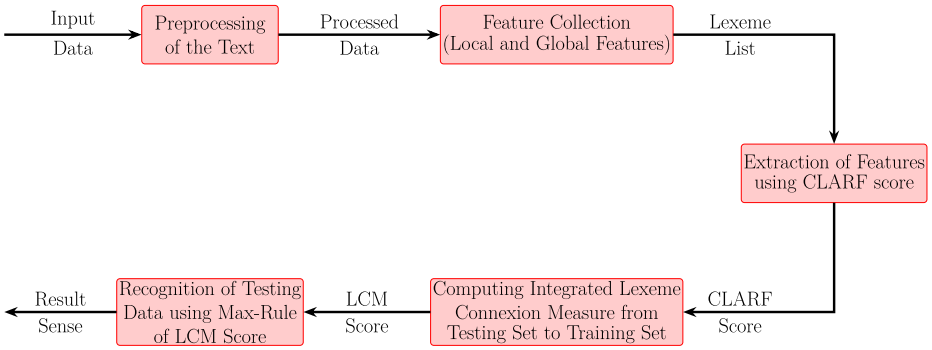

Fig. 4 Flow diagram of the testing phase

main objective of the training phase is to generate the CLARF score of each lexeme from the input data.

In the supervised learning based techniques, the testing phase is responsible for sense recognition. In this paper, we have implemented the entire testing procedure in five distinct steps, as has been shown in Fig. 4. The objective of this phase is to identify the correct sense of a polysemous word using max-rule of integrated LCM score with the help of the CLARF score associated with each lexeme.

3.1 Text pre-processing

Cognition is an underlying part of any WSD algorithm. In general, the Bengali literature contains two types of word: monosemous and polysemous. A monosemous word indicates a single sense/meaning, i.e., $|S(W)| = 1$; where W and S represent the word and the senses associated with it, respectively. It is easy to process a monosemous word. On the other hand, a polysemous word has more than one sense, i.e., $|S(W)| > 1$. The processing of polysemous words in the Bengali text is a challenging task. In this work, we have only considered words with $|S(W)| = 3$ and $|S(W)| = 4$.

The available databases are generally in an unstructured or semi-structured format, which makes feature collection very difficult as word-to-word relationships cannot be established. Thus, a structured format of the database is desirable as it includes all the words in a simple lemma form. The conversion of an unstructured or semi-structured database into a structured one comprises the following steps:

(a) **Conversion of heterogeneous font to homogeneous font**: It is a necessary step before processing the text. The homogenous font indicates the syntactic structure of the textual data.
(b) **Chopping of unwanted symbols**: This step eliminates the extra spaces, uneven brackets, broken lines, slashes, percentage signs, etc.
(c) **Deletion of terminal markers**: The set of terminal markers in Bengali literature is very similar to English literature. This step is responsible to remove the various types of terminal markers, such as |, !, ?, . . . , etc.
(d) **Removal of punctuation marks**: This step deletes punctuation marks, such as parenthesis, comma, tilde, single and double quotes, etc.

3.2 Feature selection

The objective of the feature selection step is to identify the key attributes from the input data, which is a key component for separating dissimilar items from ambiguous elements. The feature set provides this key attribute. In our proposed methodology, two distinct feature sets have been collected for encapsulating the contextual information, and they are as follows:

(a) **Local Features**: Local features deal with the local context of a particular word. The key attribute for this feature type is to collect from the single text file only, where the particular word is present. It includes all data of each sense of a polysemous word. Here, a local feature set has been generated based on a context window for each polysemous word, which contains the target word and its corresponding surrounding words. The target word is a member of the polysemous word set. The size of the frame is crucial for a fast execution of the algorithm. In this work, the proposed algorithms have extracted key attributes from context windows of sizes five, seven, and ten.

The frequency count of the target words has been taken into consideration for the local feature set. We have computed the frequency of each word for all sizes of context window, separately. The word frequency count has a direct relationship with the number of senses of a polysemous word present in the context. This phenomenon has been stated by Zipf's Law of Meaning Theory [68], which assumes that a more frequent word has more number of senses compared to a less frequent word.

(b) **Global Features**: Global features or non-local features depends on more than one text document. It includes semantic, syntactic cues, domain indicator, the argument based relationships between the target word and other words, outside the local context. Consequently, the key attribute for this feature type has been collected from all data corresponding to all senses of a particular polysemous word.

In our proposed framework these two types of features have been collected separately for the training and testing datasets. A description of the feature selection is listed below:

- **Feature Type: Local** Emphasis on retrieve keyword density with regard to single document file.
- **Feature Type: Global** Emphasis on softening the impact of terms that present too often in the documents. Thus, rare term always gets a higher priority comparison to common term.
- **Feature Type: Global** Emphasis on the keywords of given sense of a particular polysemous word.
- **Feature Type: Local + Global** Emphasis on the lexical level feature collection rudimentary metric to derive most expressive terms in a document.

3.3 Extracting features from input data using the cohesive lexical ambiguity revealing factor

In any machine learning based algorithm, feature extraction plays a pivotal role in making an extensive learning set, which validates the testing set for further processing. Such an algorithm encapsulates the feature vector from the feature set. This feature vector includes words and the corresponding word frequency counts, and identifies the top fifty, seventy, and hundred most frequent words from the context for window sizes, ws, of five, seven,

and ten, respectively. In (1), the size of the feature vector fv has been calculated using a variable number of paragraphs or documents used for training set p and window size ws.

$$|fv| = p \times [2 \times ws] \tag{1}$$

Here, in this work, we have considered that each paragraph or document contains different contextual information to describe a specific polysemy property (i.e., sense) of a polysemous word. Each paragraph contains descriptions, facts, data, and opinions to describe only one idea (or topic) per sense. This is the basis of consideration for p. On the other hand, ws refers to enclosing a smaller number of words in the disambiguation process. It refers to the minimum amount of data that is required to collect the critical lexical and semantic information from the context. From the experimental study, the minimum efficient value of ws has been found to be five. The size of the feature vector fv has been defined in (2).

$$|fv| = \begin{cases} 50, & \text{when } ws = \pm 5 \\ 70, & \text{when } ws = \pm 7 \\ 100, & \text{when } ws = \pm 10 \end{cases} \tag{2}$$

Five paragraphs (p's) have been selected from the database of each sense of a particular word for the training and testing set separately, i.e., $p = 5$. The context window expands to both left and right side form the target word [24]. Subsequently, the algorithm has computed the grandness of a lexeme by calculating the CLARF score.

In information retrieval, CLARF is a probability based statistical tool, which provides a weighting factor and sets up a unifying perspective about decision-making. It is easily computable and expressible to the key components of a document. It computes the value for every single word in a document to the percentage of documents that contain the words in question.

3.3.1 Frame lexeme harmony (FLH)

An array is considered for each polysemous word. The array contains polysemous word and surrounding or neighbouring words. Each element of the array is called lexeme and the array is called a frame. A frame F with different frame sizes fs has been created. The size of the frame $|fs|$ is specified in (3).

$$|fs| = 1 + 2 \times |ws| \tag{3}$$

In this work, we have assumed three window sizes, and accordingly the frame sizes are as following:

$$|fs| = \begin{cases} 11, & \text{when } ws = \pm 5 \\ 15, & \text{when } ws = \pm 7 \\ 21, & \text{when } ws = \pm 10 \end{cases} \tag{4}$$

In morphology analysis, lexeme provides the literal meaning of each sense of a polysemous word. The laws of reflection speak about one lexeme to its outlines and the derivational rules relate one lexeme to one more lexeme [7]. A lexeme embodies the utmost elementary building array in a language. Each lexeme provides internal meaning of each sense of a polysemous word. In this paper, we have considered both single word and multi-word as a lexeme.

In the case of feature mining, we have considered the lexeme-rate-of-repetition $\lambda(\ell)$ of each lexeme ℓ in a specific frame F. (5) provides the mathematical notation of the frame-based lexeme collection. This is a binary function, it returns either true or false.

$$\lambda(\ell) = \begin{cases} 1, & \text{if } \ell \in \mathrm{F} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

To collect the effects of each lexeme together, in this paper, we have summarised $\lambda(\ell)$ for a collection of all frames. Equation (6) provides a cognitive notation of the harmony, $\mathrm{H}(\ell)$, of each lexeme ℓ.

$$\mathrm{H}(\ell) = \sum^{n} \lambda(\ell), \tag{6}$$

where n is the number of frames in the ith polysemous word W of the jth sense S, i.e., $\mathrm{W}_i\mathrm{S}_j$.

Lexeme harmony apportions syntactic belongings of a collection of lexemes of same type. Equation (7) defines the lexeme harmony $\mathrm{LH}(\ell)$ of each ℓ as:

$$\mathrm{LH}(\ell) = [\mathrm{H}(\ell)]^2 + 2\mathrm{H}(\ell) \tag{7}$$

This is a continuous function for every real value of $\mathrm{H}(\ell)$. Equation (8) specifies the continuity of $\mathrm{LH}(\ell)$ at any point $\lambda(\ell)$.

$$\lim_{\mathrm{H}(\ell)\to\lambda(\ell)+0} \mathrm{LH}(\ell) = \lim_{\mathrm{H}(\ell)\to\lambda(\ell)-0} \mathrm{LH}(\ell) = \lim \lambda(\ell) \tag{8}$$

FLH indicates the grade of approximation of each ℓ. It is a heuristic measure of lexeme harmony. The working formula of this grade has been illustrated in (9).

$$\mathrm{FLH}(\ell) = \frac{|fs|}{|fv|} \times \mathrm{LH}(\ell) = \frac{1 + 2|ws|}{|fv|} \times \mathrm{LH}(\ell), \tag{9}$$

The corresponding algorithm of the generation of frame lexeme harmony is stated in Algorithm 1.

Input: training set (p), window sizes (ws), frame (F), and set of lexemes ℓ (L)
Output: frame lexeme harmony (FLH) and harmony (H) of each lexeme $\ell \in \mathrm{L}$

1: **while** $\mathrm{L} \neq \emptyset$ **do**
2: **for** each lexeme $\ell \in \mathrm{L}$ **do**
3: Calculate lexeme-rate-of-repetition $\lambda(\ell)$ using (5)
4: Calculate the harmony, $\mathrm{H}(\ell)$, of lexeme ℓ using (6)
5: Calculate the lexeme harmony, $\mathrm{LH}(\ell)$, of lexeme ℓ using (7) and (8)
6: Calculate the frame lexeme harmony, $\mathrm{FLH}(\ell)$, of lexeme ℓ using (1) through (5) and (9)
7: **end for**
8: **end while**
9: **Return** FLH and H

Algorithm 1 GENERATE FRAME LEXEME HARMONY.

3.3.2 Sense lexeme harmony (SLH)

Sense lexeme harmony (SLH) is the (FLH) across all senses of a specific polysemous word. (SLH) of each ℓ has been calculated using (10).

$$\mathrm{SLH}\,(\ell) = \sum^{|S_j|} \mathrm{FLH}\,(\ell) \tag{10}$$

In this experimental setup, we have considered the value $|S_j|$ is three or four, depending on number of senses of a specific polysemous word. The corresponding algorithm of the generation of sense lexeme harmony is stated in Algorithm 2.

Input: Sense of polysemous word j $\left(\mathrm{S}_j\right)$, frame lexeme harmony (FLH), and set of lexemes ℓ (L)
Output: Sense lexeme harmony (SLH)
1: **while** $\mathrm{L} \neq \emptyset$ **do**
2: **for** each lexeme $\ell \in \mathrm{L}$ **do**
3: **for** each sense $\left(\mathrm{S}_j\right)$ of a polysemous word j **do** ▷ where number of senses $|\mathrm{S}_j| = 3/4$
4: Calculate the sense lexeme harmony, SLH (ℓ), of lexeme ℓ according to (10)
5: **end for**
6: **end for**
7: **end while**
8: **Return** SLH

Algorithm 2 GENERATE SENSE LEXEME HARMONY.

3.3.3 Polysemy singularity coherence (PSC)

In (11), all the frames of each sense of a polysemous word is collected.

$$\mathrm{F}_{\mathrm{W}_i\mathrm{S}_j} = \bigcup_{i=1}^{n} \mathrm{F}_i \tag{11}$$

PSC computes the rareness or uniqueness of a lexeme across the entire document corpus, and enables us to identify any keyword from any text file. This score of a rare term is invariably high. The working formula of PSC of each ℓ has been defined below:

$$\mathrm{PSC}\,(\ell) = \begin{cases} \log_{|\mathrm{S}_j|} \frac{|\mathrm{S}_j|}{1}, & \text{if } \ell \in \left(\mathrm{F}_{\mathrm{W}_i\mathrm{S}_1}\right) \\ \log_{|\mathrm{S}_j|} \frac{|\mathrm{S}_j|}{2}, & \text{if } \ell \in \left(\mathrm{F}_{\mathrm{W}_i\mathrm{S}_1} \cap \mathrm{F}_{\mathrm{W}_i\mathrm{S}_2}\right) \\ \log_{|\mathrm{S}_j|} \frac{|\mathrm{S}_j|}{3}, & \text{if } \ell \in \left(\mathrm{F}_{\mathrm{W}_i\mathrm{S}_1} \cap \mathrm{F}_{\mathrm{W}_i\mathrm{S}_2} \cap \mathrm{F}_{\mathrm{W}_i\mathrm{S}_3}\right) \\ \log_{|\mathrm{S}_j|} \frac{|\mathrm{S}_j|}{4}, & \text{if } \ell \in \left(\mathrm{F}_{\mathrm{W}_i\mathrm{S}_1} \cap \mathrm{F}_{\mathrm{W}_i\mathrm{S}_2} \cap \mathrm{F}_{\mathrm{W}_i\mathrm{S}_3} \cap \mathrm{F}_{\mathrm{W}_i\mathrm{S}_4}\right) \end{cases} \tag{12}$$

The corresponding algorithm of the generation of polysemy singularity coherence is stated in Algorithm 3.

Input: frames (F), sense of word j $\left(\mathrm{S}_j\right)$, set of lexemes ℓ (L), number of frames (n), and set of lexemes ℓ (L)
Output: polysemy singularity coherence (PSC)
1: **while** L $\neq \emptyset$ **do**
2: **for** each lexeme $\ell \in$ L **do**
3: Calculate the polysemy singularity coherence, PSC (ℓ), of lexeme ℓ using (11) and (12)
4: **end for**
5: **end while**
6: **Return** PSC

Algorithm 3 GENERATE POLYSEMY SINGULARITY COHERENCE.

3.3.4 Polysemy distribution factor (PDF)

Polysemy distribution factor (PDF) is a statistical load used to estimate the prominence of a term in a text document collection. It specifies a falsetto value to the documents that include the rare terms with respect to all senses. It can be calculated by dividing the total number of occurrences of a ℓ in all senses to the total number of occurrences of a ℓ in the particular sense, according to:

$$\mathrm{PDF}\left(\ell\right) = \frac{\sum^{|\mathrm{S}_j|} \mathrm{H}\left(\ell\right)}{\mathrm{H}\left(\ell\right)} \tag{13}$$

Input: sense of word j $\left(\mathrm{S}_j\right)$, harmony (H), and set of lexemes ℓ (L)
Output: polysemy distribution factor (PDF)
1: **while** L $\neq \emptyset$ **do**
2: **for** each lexeme $\ell \in$ L **do**
3: **for** each sense $\left(\mathrm{S}_j\right)$ of a polysemous word j **do**
4: Calculate the polysemy distribution factor, PDF (ℓ), of lexeme ℓ using (13)
5: **end for**
6: **end for**
7: **end while**
8: **Return** PDF

Algorithm 4 GENERATE POLYSEMY DISTRIBUTION FACTOR.

The corresponding algorithm of the generation of polysemy distribution factor is stated in Algorithm 4.

3.3.5 Relative polysemy singularity coherence (RPSC)

Relative polysemy singularity coherence indicates the global feature type and is collected from multiple document files. The RPSC score helps in separating the key terms from the other terms by providing a higher score for the former. Nevertheless, the integration of unprocessed PDF with PSC may interrupt the all-inclusive term discrimination value,

because the PSC score is computed by diminishing through logarithm function. Hence, a slowly growing transfer function $f(x) = \frac{x}{1+x}$ is applied while calculating the RPSC score. The mathematical formulation of this score has been given in (14).

$$\text{RPSC}(\ell) = \frac{\text{PDF}(\ell)}{1 + \text{PDF}(\ell)} \times \text{PSC}(\ell) \tag{14}$$

The corresponding algorithm for the generation of RPSC has been stated in Algorithm 5.

Input: set of lexemes ℓ (L), polysemy singularity coherence (PSC), and polysemy distribution factor (PDF)
Output: relative polysemy singularity coherence (RPSC)
1: **while** $L \neq \emptyset$ **do**
2: **for** each lexeme $\ell \in L$ **do**
3: Calculate the relative polysemy singularity coherence, RPSC (ℓ), of lexeme ℓ using (14)
4: **end for**
5: **end while**
6: **Return** RPSC

Algorithm 5 GENERATE RELATIVE POLYSEMY SINGULARITY COHERENCE.

3.3.6 Cohesive lexical ambiguity revealing factor (CLARF)

Finally, the cohesive lexical ambiguity revealing factor or CLARF score has been computed in this work by multiplying the SLH and RPSC scores of a given lexeme, as shown in (16). It is a collection of both local and global features. This score is critical for the recognition phase.

$$\text{CLARF}(\ell) = \text{SLH}(\ell) \times \text{RPSC}(\ell) \tag{15}$$

$$= \sum^{|S_j|} \text{FLH}(\ell) \times \frac{\text{PDF}(\ell)}{1 + \text{PDF}(\ell)} \times \text{PSC}(\ell) \tag{16}$$

The corresponding algorithm for the generation of cohesive lexical ambiguity revealing factor or CLARF has been stated in Algorithm 6.

3.4 Sense lexeme preparation in testing and training sets

In this paper, the training and testing datasets for a polysemous word of a particular sense have been considered as disjoint sets. It has been applied 50% segregation of total data to determine training and test sets. In WSD, sense lexeme preparation plays an important role in identifying the correct concept of the polysemous word.

In the training phase, this paper has been performed sense lexeme preparation in the following two ways:

(a) Equation (17) denotes the mathematical notation of sense lexeme preparation, where number of senses is three. The corresponding algorithm of sense lexeme preparation in training sets for $|S_j| = 3$ has been mentioned in Algorithm 7.

Input: set of lexemes ℓ (L), sense lexeme harmony (SLH), and relative polysemy singularity coherence (RPSC)
Output: cohesive lexical ambiguity revealing factor (CLARF)
1: **while** L $\neq \emptyset$ **do**
2: **for** each lexeme $\ell \in$ L **do**
3: Calculate the cohesive lexical ambiguity revealing factor, CLARF (ℓ), of lexeme ℓ, using (15) and (16)
4: **end for**
5: **end while**
6: **Return** CLARF

Algorithm 6 Generate Cohesive Lexical Ambiguity Revealing Factor.

(b) Equation (18) represents the mathematical notation of sense lexeme preparation, where number of senses is four. The corresponding algorithm of sense lexeme preparation in training sets for $|S_j| = 4$ is has been mentioned in Algorithm 8.

$$\top_{|S_j|=3} = \{\ell \mid \ell \notin \left(F_{W_i S_1} \cap F_{W_i S_2} \cap F_{W_i S_3}\right)\} \tag{17}$$

$$\top_{|S_j|=4} = \{\ell \mid \ell \notin \left(F_{W_i S_1} \cap F_{W_i S_2} \cap F_{W_i S_3} \cap F_{W_i S_4}\right)\} \tag{18}$$

Input: all frames with words up to three senses $\left(F_{W_i S_j}\right)$ and set of lexemes ℓ (L).
Output: training set for three senses $\left(\top_{|S_j|=3}\right)$.
1: **while** L $\neq \emptyset$ **do**
2: **for** each lexeme $\ell \in$ L **do**
3: Generate the training set for three senses, $\top_{|S_j|=3}$, for each lexeme ℓ using (17)
4: **end for**
5: **end while**
6: **Return** $\top_{|S_j|=3}$.

Algorithm 7 Generate Training Set For Three Senses.

Input: all frames with words up to three senses $\left(F_{W_i S_j}\right)$ and set of lexemes ℓ (L).
Output: training set for four senses $\left(\top_{|S_j|=4}\right)$.
1: **while** L $\neq \emptyset$ **do**
2: **for** each lexeme $\ell \in$ L **do**
3: Generate the training set for three senses, $\top_{|S_j|=4}$, for each lexeme ℓ using (18)
4: **end for**
5: **end while**
6: **Return** $\top_{|S_j|=4}$.

Algorithm 8 Generate Training Set For Four Senses.

Sense lexeme filtration is essential to prepare an effective training module for the training session. Subsequently, in the testing phase, sense lexeme has been prepared without filtered any data. The detailed interpretation of this phase has been defined in (19). The corresponding algorithm of sense lexeme preparation in testing sets is mentioned in Algorithm 9.

$$\perp = \{\ell \mid \ell \notin \top_{|\mathrm{S}_j|=3} \text{ and } \ell \notin \top_{|\mathrm{S}_j|=4}\} \tag{19}$$

Input: all frames with words up to three senses $\left(\mathrm{F}_{\mathrm{W}_i\mathrm{S}_j}\right)$ and set of lexemes ℓ (L).
Output: testing set ($\perp$)
1: **while** L $\neq \emptyset$ **do**
2: **for** each lexeme $\ell \in$ L **do**
3: Generate testing set using (19)
4: **end for**
5: **end while**
6: **Return** $\perp$.

Algorithm 9 GENERATE TESTING SET.

3.5 Recognition of proper sense of the testing data

In sense recognition, this paper has been computed the lexeme connexion measure or LCM of each lexeme for both the testing and training cases. The mathematical expression of LCM $\omega\left(\perp(\ell), \top(\ell)_{\mathrm{W}_i\mathrm{S}_j}\right)_{\mathrm{W}_i\mathrm{S}_j}$ has been computed using (20) through (23). The corresponding algorithm of calculating individual lexeme connexion measure between testing sets to training sets has been mentioned in Algorithm 10.

$$\omega_1 = \omega\left(\perp(\ell), \top(\ell)_{\mathrm{W}_i\mathrm{S}_1}\right)_{\mathrm{W}_i\mathrm{S}_1} = \frac{\mathrm{CLARF}(\ell) \times \mathrm{CLARF}(\ell)_{\mathrm{W}_i\mathrm{S}_1}}{\sqrt{\mathrm{CLARF}(\ell)^2 \times \mathrm{CLARF}(\ell)^2_{\mathrm{W}_i\mathrm{S}_1}}} \tag{20}$$

$$\omega_2 = \omega\left(\perp(\ell), \top(\ell)_{\mathrm{W}_i\mathrm{S}_2}\right)_{\mathrm{W}_i\mathrm{S}_2} = \frac{\mathrm{CLARF}(\ell) \times \mathrm{CLARF}(\ell)_{\mathrm{W}_i\mathrm{S}_2}}{\sqrt{\mathrm{CLARF}(\ell)^2 \times \mathrm{CLARF}(\ell)^2_{\mathrm{W}_i\mathrm{S}_2}}} \tag{21}$$

$$\omega_3 = \omega\left(\perp(\ell), \top(\ell)_{\mathrm{W}_i\mathrm{S}_3}\right)_{\mathrm{W}_i\mathrm{S}_3} = \frac{\mathrm{CLARF}(\ell) \times \mathrm{CLARF}(\ell)_{\mathrm{W}_i\mathrm{S}_3}}{\sqrt{\mathrm{CLARF}(\ell)^2 \times \mathrm{CLARF}(\ell)^2_{\mathrm{W}_i\mathrm{S}_3}}} \tag{22}$$

$$\omega_4 = \omega\left(\perp(\ell), \top(\ell)_{\mathrm{W}_i\mathrm{S}_4}\right)_{\mathrm{W}_i\mathrm{S}_4} = \frac{\mathrm{CLARF}(\ell) \times \mathrm{CLARF}(\ell)_{\mathrm{W}_i\mathrm{S}_4}}{\sqrt{\mathrm{CLARF}(\ell)^2 \times \mathrm{CLARF}(\ell)^2_{\mathrm{W}_i\mathrm{S}_4}}} \tag{23}$$

After calculating the individual LCM, $\omega_j = \omega\left(\perp(\ell), \top(\ell)_{\mathrm{W}_i\mathrm{S}_j}\right)_{\mathrm{W}_i\mathrm{S}_j}$, the integrated LCM, $\Omega_j = \Omega\left(\perp, \top(\ell)_{\mathrm{W}_i\mathrm{S}_j}\right)_{\mathrm{W}_i\mathrm{S}_j}$ has been calculated using (24) through (27). Here, k_1, k_2, k_3, and k_4 indicate the total number of lexeme at Sense 1, Sense 2, Sense 3, and Sense 4, respectively. The corresponding algorithm of calculating integrated lexeme connexion

Input: sense (j), set of lexemes ℓ (L), cohesive lexical ambiguity revealing factor (CLARF), testing set ($\perp$), and training sets $\left(\top_{\mathrm{W}_i\mathrm{S}_j}\right)$
Output: individual lexeme connexion measure for all senses $j = 1$ to 4 $\left(\omega_j\right)$
1: **while** $\perp(\ell) \neq \emptyset$ **do**
2: Calculate the individual lexeme connexion measure for all senses $j = 1$ to 4, ω_j, using (20) through (23)
3: **end while**
4: **Return** $\omega_j \forall j$

Algorithm 10 CALCULATE INDIVIDUAL LEXEME CONNEXION MEASURE.

measure between testing sets to training sets has been mentioned in Algorithm 11.

$$\Omega_1 = \Omega\left(\perp, \top_{\mathrm{W}_i\mathrm{S}_1}\right)_{\mathrm{W}_i\mathrm{S}_1} = \sum_{\ell=1}^{k_1} \omega_1 = \sum_{\ell=1}^{k_1} \omega\left(\perp(\ell), \top(\ell)_{\mathrm{W}_i\mathrm{S}_1}\right)_{\mathrm{W}_i\mathrm{S}_1} \tag{24}$$

$$\Omega_2 = \Omega\left(\perp, \top_{\mathrm{W}_i\mathrm{S}_2}\right)_{\mathrm{W}_i\mathrm{S}_2} = \sum_{\ell=1}^{k_1} \omega_2 = \sum_{\ell=1}^{k_2} \omega\left(\perp(\ell), \top(\ell)_{\mathrm{W}_i\mathrm{S}_2}\right)_{\mathrm{W}_i\mathrm{S}_2} \tag{25}$$

$$\Omega_3 = \Omega\left(\perp, \top_{\mathrm{W}_i\mathrm{S}_3}\right)_{\mathrm{W}_i\mathrm{S}_3} = \sum_{\ell=1}^{k_1} \omega_3 = \sum_{\ell=1}^{k_3} \omega\left(\perp(\ell), \top(\ell)_{\mathrm{W}_i\mathrm{S}_3}\right)_{\mathrm{W}_i\mathrm{S}_3} \tag{26}$$

$$\Omega_4 = \Omega\left(\perp, \top_{\mathrm{W}_i\mathrm{S}_4}\right)_{\mathrm{W}_i\mathrm{S}_4} = \sum_{\ell=1}^{k_1} \omega_4 = \sum_{\ell=1}^{k_4} \omega\left(\perp(\ell), \top(\ell)_{\mathrm{W}_i\mathrm{S}_4}\right)_{\mathrm{W}_i\mathrm{S}_4} \tag{27}$$

Input: individual lexeme connexion measure for all senses $j = 1$ to 4 $\left(\omega_j\right)$
Output: integrated lexeme connexion measure for all senses $j = 1$ to 4 $\left(\Omega_j\right)$
1: **for** $j = 1$ to 4 **do**
2: $k_j = \left|\omega_j\right|$ ▷ Calculate the number of lexemes for each sense
3: **end for**
4: Calculate the integrated lexeme connexion measure for all senses $j = 1$ to 4, Ω_j, using k_j according to (24) through (27)
5: **Return** $\Omega_j \forall j$

Algorithm 11 CALCULATE INTEGRATED LEXEME CONNEXION MEASURE.

Subsequently, max-rule of integrated LCM score is applied for sense recognition. (28) specifies the mathematical notation of the identification of sense of $\perp$. It is shown four different cases in four senses. The first three cases are applied where the number of sense is three. This part of the proposed algorithm plays a crucial role in sense recognition.

$$\perp = \begin{cases} \textbf{Sense 1}, \text{ if } \max\left[\Omega_1, \Omega_2, \Omega_3, \Omega_4\right] = \Omega_1 \\ \textbf{Sense 2}, \text{ if } \max\left[\Omega_1, \Omega_2, \Omega_3, \Omega_4\right] = \Omega_2 \\ \textbf{Sense 3}, \text{ if } \max\left[\Omega_1, \Omega_2, \Omega_3, \Omega_4\right] = \Omega_3 \\ \textbf{Sense 4}, \text{ if } \max\left[\Omega_1, \Omega_2, \Omega_3, \Omega_4\right] = \Omega_4 \end{cases} \tag{28}$$

The corresponding algorithm of the recognition of proper sense of the testing data has been mentioned in Algorithm 12. The main algorithm has been stated in Algorithm 13.

Input: integrated lexeme connexion measure for all senses $j = 1$ to 4 $\left(\Omega_j\right)$
Output: Sense
1: Determine the appropriate sense using (28)
2: **Return** the sense obtained

Algorithm 12 PERFORM SENSE RECOGNITION.

Input: set of words (W), training set (p), window size (ws), frames (F), number of frames (n), and set of senses (S)
Output: the appropriate sense
1: Generate frame lexeme harmony using Algorithm 1
2: Generate sense lexeme harmony using Algorithm 2
3: Generate polysemy singularity coherence using Algorithm 3
4: Generate polysemy distribution factor using Algorithm 4
5: Generate relative polysemy singularity coherence using Algorithm 5
6: Generate cohesive lexical ambiguity revealing factor using Algorithm 6
7: Generate training set for three senses using Algorithm 7
8: Generate training set for four senses using Algorithm 8
9: Generate testing set using Algorithm 9
10: Calculate individual lexeme connexion measure using Algorithm 10
11: Calculate integrated lexeme connexion measure using Algorithm 11
12: Perform sense recognition using Algorithm 12
13: **Return** the sense found

Algorithm 13 WORD SENSE DISAMBIGUATION.

4 Datasets used

Over the last decade, researchers in the domain of NLP have made a meaningful move from unsupervised learning based approaches to supervised learning based ones in WSD, especially in the case of morphologically complex languages [38]. The supervised learning based WSD models provide a machine learning based categorisation tool from manually sense-annotated datasets. The potential of supervised learning based approaches depends on the dimension and standardisation of the training dataset, which is not available in Bengali. As a result, a dataset of one hundred polysemous words of $|S(W)| = 3$ and $|S(W)| = 4$ was presented in [14].

- **Set of Bengali Words with $|S(W)| = 3$**: This part of the database includes a collection of Bengali polysemous words that have three distinct senses. This part of the data contains fifty-five words out of a hundred polysemous words present in the whole database. The ambiguity of a polysemous word has been characterised

by a textual paragraph. Each sense of a specific polysemous word has five associated paragraphs in the training and testing datasets, and the data has been collected from Bengali news corpus, books, magazine, online news portal, social media, etc. Thus, in the training module, fifteen paragraphs have been included for three senses. In the testing module, five paragraphs have been used for validation. A list of polysemous words of this portion are the following: আগ ($\bar{A}ga$), উঠা ($U\dot{t}h\bar{a}$), তালা ($T\bar{a}l\bar{a}$), ছত্র ($Chatra$), বর্তমান ($Bartam\bar{a}na$), হল ($Hala$), পাতা ($P\bar{a}t\bar{a}$), শব্দ ($\acute{S}abda$), সময় ($Samaẏa$), নাম ($N\bar{a}ma$), ঘন্টা ($Ghan\dot{t}\bar{a}$) and so on.

- **Set of Bengali Words with $|S(W)| = 4$**: This part of the database contains Bengali polysemous words which have four senses. There are forty-five such polysemous words, out of hundred words in the database. The sense description of each polysemous word has been mentioned in textual form. In this case also, the textual data has been collected from the Bengali news corpus, online data, social media, literature, mass media, blog, etc. The training and testing datasets are two disjoint sets, containing five paragraphs each for a sense of a specific polysemous word. Thus, in the learning module, twenty paragraphs have been used for four senses. In the testing module, five paragraphs have been used for validation. This part of the database tests the algorithmic potential of the proposed method. The nearest English translation as well as a transliteration for each Bengali word has also been mentioned. A list of polysemous words of this portion are the following: জল ($Jala$), হাত ($H\bar{a}ta$), মাথা ($M\bar{a}th\bar{a}$), বুক ($Buka$), মন ($Mana$), কড়া ($Kaṛ\bar{a}$), পা ($P\bar{a}$), মুখ ($Mukha$), ফল ($Phala$), দিন ($Dina$) and so on.

The overall database for Bengali word জল ($Jala$: $Water$) of Sense 1 is পানীয় বিশেষ ($P\bar{a}n\bar{\imath}\ a\ bi\acute{s}\bar{e}\d{s}a$: $Type\ of\ drink$) has been presented following ways:

(i) Specimen for $\top_{W_1 S_1}$

(a) **Snapshot 1**: পানি বা জল হলো একটি যৌগ পদার্থ, যার রাসায়নিক সংকেত হল H_2O। এক অণু জল দু'টি হাইড্রোজেন পরমাণু এবং একটি অক্সিজেন পরমাণুর সমযোজী বন্ধনে গঠিত। সাধারণত পৃথিবীতে জল তরল অবস্থায় থাকলেও এটি কঠিন (বরফ) এবং বায়বীয় অবস্থাতেও (জলীয় বাষ্প) পাওয়া যায়।

Transliteration: Pāni bā jala halō ēkaṭi yauga padārtha, yāra rāsā anika saṅkēta hala H_2O. Ēka aṇu jala du'ṭi hā'iḍrōjēna paramāṇu ēba ēkaṭi aksijēna paramāṇura samayōjī bandhanē gaṭhita. Sādhāraṇata pr thibītē jala tarala abasthā a thākalē'ō ēṭi kaṭhina (barapha) ēba bā abī a abasthātē'ō (jalī a bāṣpa) pā'ō ā yā a.

Translation: Water is a compound, whose chemical formula is H_2O. One molecule of water consists of a covalent bond between two hydrogen atoms and an oxygen atom. Although, water is generally present in the liquid state on Earth, it can also be found in the solid (ice) and gaseous (vapour) states.

(b) **Snapshot 2**: সারাদিনে কতবার জল পান করেন? সকলেই বলেন বেশি করে জল খেতে, আর আপনিও মনের আনন্দে সুযোগ পেলেই জল খান। কিন্তু জানেন কি দিনের যে কোনও সময়ে বেশি জল খাওয়ার অভ্যাস মোটেই ভাল নয়?

Transliteration: Sārādinē katabāra jala pāna karēna? Sakalē'i balēna bēśi karē jala khētē, āra āpani'ō manēra ānandē suyōga pēlē'i jala khāna. Kintu jānēna ki dinēra yē kōna'ō sama ē bēśi jala khā'ō āra abhyāsa mōṭē'i bhāla na a?

Translation: How many times a day do you drink water? Everyone tells you to drink more water, and you consume it happily whenever you get a chance. However, do you know that the habit of drinking too much water at any time of the day is not good at all?

(c) **Snapshot 3**: সুস্থ থাকতে, ওজন কমাতে পর্যাপ্ত জল খাওয়া প্রয়োজন। চিকিৎসক থেকে ডায়েটিশিয়ান, সকলেই এই কথা বলে থাকেন। তবে ঠিকঠাক নিয়ম মেনে জল না খেলে গুরুতর শারীরিক সমস্যা হতে পারে। জেনে নিন সুস্থ থাকতে কোন কোন সময়ে জল খাওয়া এড়িয়ে চলবেন।

Transliteration: Sustha thākatē, ōjana kamātē paryāpta jala khā'ō ā pra ōjana. Cikitsaka thēkē ḍā ēṭiśi āna, sakalē'i ē'i kathā balē thākēna. Tabē ṭhikaṭhāka ni ama mēnē jala nā khēlē gurutara śārīrika samasyā hatē pārē. Jēnē nina sustha thākatē kōna kōna sama ē jala khā'ō ā ēṛi ē calabēna.

Translation: To remain healthy, lose weight, it is necessary to drink enough water. From doctors to dietitians, everyone says this. However, not drinking water according to proper rules can lead to serious physical problems. Find out when to avoid drinking water to stay healthy.

(ii) Specimen for $\perp_{W_1 S_1}$

(a) **Snapshot 1**: পৃথিবীর উপরকার জল ক্রমে মাটির ভিতরে কত নীচে পর্যন্ত চুঁইয়া পড়ে এবং কেন বা একস্থানে আসিয়া বাধা পায় এখনও তাহার ভালোরূপ তথ্য নির্ণয় হয় নাই। কিন্তু ইহা দেখা গিয়াছে যে, ভূতলের নিম্নস্তরে কিছুদূর নামিয়া আসিলেই একটি সম্পূর্ণ জলসিক্তস্থানে আসিয়া পৌঁছানো যায়, সেখানে মৃত্তিকার প্রত্যেক ছিদ্র বায়ুর পরিবর্তে কেবলমাত্র জলে পরিপূর্ণ। যেমন সমুদ্রের জল সর্বত্রই সমতল তেমনি মাটির নীচের জলেরও একটা সমতলতা আছে। কোনো বিশেষ প্রদেশের ভূগর্ভস্থ জলের তলোচ্চতা কী, তাহা সে দেশের কূপের জলতল দেখিলেই বুঝা যাইতে পারে।

Transliteration: Pr thibīra uparakāra jala kramē māṭira bhitarē kata nīcē paryanta cum i ā paṛē ēba kēna bā ēkasthānē āsi ā bādhā pā a ēkhana'ō tāhāra bhālōrūpa tathya nirṇa a ha a nā'i. Kintu ihā dēkhā gi āchē yē, bhūtalēra nimnastarē kichudūra nāmi ā āsilē'i ēkaṭi sampūrṇa jalasiktasthānē āsi ā paum chānō yā a, sēkhānē mr ttikāra pratyēka chidra bā ura paribartē kēbalamātra jalē paripūrṇa. Yēmana samudrēra jala sarbatra'i samatala tēmani māṭira nīcēra jalēra'ō ēkaṭā samatalatā āchē. Kōnō biśēṣa pradēśēra bhūgarbhastha jalēra talōccatā kī, tāhā sē dēśēra kūpēra jalatala dēkhilē'i bujhā yā'itē pārē.

Translation: Up to what depth water present on the earth's surface gradually percolates through the soil, and why does it stop somewhere is still not well-known. However, it has been observed that one can reach a completely water-soaked place by descending a little beneath the surface, where every pore in the soil is filled with water instead of air. Just as ocean water is at the same level everywhere, groundwater also has a level. On can ascertain the groundwater level in any particular province by observing the water table of the wells in that area.

(b) **Snapshot 2**: গৃহে জল সরবরাহ সংযোগের জন্য নিম্নলিখিত পদ্ধতি অনুসরণ করতে হবে। জল সরবরাহ বিভাগের স্থানীয় অফিসে সহকারী বাস্তুকারের সঙ্গে যোগাযোগ। সেখান থেকে নথিভুক্ত প্লাম্বারদের তালিকা সংগ্রহ। একজন প্লাম্বার নির্বাচন এবং সেই প্লাম্বার কর্তৃক সম্ভাব্য খরচের পরিমাণ নির্ধারণ। উপভোক্তা কর্তৃক নির্দিষ্ট আবেদনপত্রে অনুমোদনের জন্য সহকারী বাস্তুকারের কাছে আবেদন। অনুমোদনের পর টাকা জমা দিলে সংযোগ পাওয়া যাবে।

Transliteration: Gṛhē jala sarabarāha sanyōgēra jan'ya nimnalikhita pad'dhati anusaraṇa karatē habē. Jala sarabarāha bibhāgēra sthānīẏa aphisē sahakārī bāstukārēra saṅgē yōgāyōga. Sēkhāna thēkē nathibhukta plāmbāradēra tālikā saṅgraha. Ēkajana plāmbāra nirbācana ēba sē'i plāmbāra kartṛka sambhābya kharacēra parimāṇa nirdhāraṇa. Upabhōktā kartṛka nirdiṣṭa ābēdanapatrē anumōdanēra jan'ya sahakārī bāstukārēra kāchē ābēdana. Anumōdanēra para ṭākā jamā dilē sanyōga pā'ō ā yābē.

Translation: The following procedure needs to be followed for obtaining water supply link at home. Contacting the Assistant Inspector at the local office of the Water Supply Department. Collection of a list of registered plumbers from there. Selection of a plumber and determination of the amount of possible expenditure by that plumber. Application by the consumer to the Assistant Inspector for approval using the specific application form. After approval, link can be obtained on depositing money.

(c) **Snapshot 3**: শরীর ভাল রাখতে পর্যাপ্ত জল খাওয়া প্রয়োজন সকলেই জানেন। কিন্তু দিনে ঠিক কতটা জল পান করা উচিত এটা বলা কার্যত বেশ কঠিন। সাধারণ ভাবে আমাদের মত গরমের দেশে দিনে ৮ থেকে ১২ গ্লাস জল খাওয়া উচিত। কিন্তু অনেকে হয় খুব কম জল খান, কেউ কেউ আবার খুব বেশি জল খান। বেশি জল খেলে শরীরের কি কি সমস্যা দেখা দিতে পারে, সেটাই দেখার।

Transliteration: Śarīra bhāla rākhatē paryāpta jala khā'ō ā pra ōjana sakalē'i jānēna. Kintu dinē ṭhika kataṭā jala pāna karā ucita ēṭā balā kāryata bēśa kaṭhina. Sādhāraṇa bhābē āmādēra mata garamēra dēśē dinē 8 thēkē 12 glāsa jala khā'ō ā ucita. Kintu anēkē ha a khuba kama jala khāna, kē'u kē'u ābāra khuba bēśi jala khāna. Bēśi jala khēlē śarīrēra ki ki samasyā dēkhā ditē pārē, sēṭā'i dēkhāra.

Translation: Everyone knows that it is necessary to drink enough water to keep the body healthy. However, it is quite difficult to say exactly how much water to drink in a day. In general, in a hot country like ours, it is advisable to drink 8 to 12 glasses of water a day. However, many people drink very little water, while some people drink too much water. It remains to be seen what health problems can occur due to drinking too much water.

5 Experimental results

The performance of any WSD technique is evaluated based on four parameters, *precision*, *recall*, F-measure or F-score, and *accuracy*. The number of senses correctly disambiguated

may differ from the actual number of senses associated with any target word. The algorithm has been validated on a database of a hundred polysemous words. Each word has either three or four senses. Each sense has been included in two disjoint sets; five paragraphs for training and five paragraphs for the testing dataset. The experimental framework consists of Python with the *natural language tool kit* (NLTK).

5.1 Performance metrics

The above-mentioned performance metrics have been defined in the following sections.

5.1.1 Precision

Precision can be defined as the number of senses correctly classified among all senses classified. In other words, it is an estimation of the number of true sense disambiguations out of all sense disambiguations. It brings forth the best possible value for the selection of a distinct correct positive case, and sheds light on a baseline parameter. Precision has been calculated in this work according to (29). This value indicates a refinement in the measurement and is also known as the *positive predictive value*.

$$p = \frac{tp}{tp + fp}, \tag{29}$$

where tp (true positive) indicates the number of senses of the target word that have been correctly disambiguated and fp (false positive) denotes the number of senses of the target word that have been incorrectly disambiguated.

However, precision alone is not an approvingly relevant evaluation criterion for WSD techniques. It only focuses on the true classifications and enhances the negative cases as unwanted items. A high value of precision cannot express much about the efficiency of any working principle.

5.1.2 Recall

Recall or *true positive rate* (TPR) can be described as the fraction of instances that have been categorised properly, and accounts for the non-classification instances, unlike precision. In our work, recall is thus the ratio of the correctly predicted senses to all the number of senses present in the assessment corpus. Recall has been calculated in this work as follows:

$$r = \frac{tp}{tp + fn}, \tag{30}$$

where fn (false negative) indicates the number of senses of a target word that has not been disambiguated. Recall indicates the *sensitivity* of any algorithm.

Subsequently, permutations of p and r is generally used as the foremost means of performance assessment of any algorithm. These two parameters are utilised to compare two state-of-the-art methodologies. A relative higher value of p and r alludes to the better performance of an algorithm compared to another.

Both p and r are responsible for the number of correct disambiguations, in case of three or more senses of a polysemous target word [8], and ignore the nature of misclassification. Consequently, the combination of p and r is very useful in our case, because the

performance of the proposed algorithm requires evaluation of a dataset that consists of three and four senses. A harmonic mean of p and r has been used to calculate the F-measure score for a better comparison of achieved performance.

5.1.3 F-measure or F-score

The F-measure (F_1) strives for steadiness between p and r. It is very useful in the case of asymmetrical sense distribution, i.e., when a significant quantity of actual negatives are present in the confusion matrix. The F-measure has been estimated in this work according to (31).

$$F_1 = 2 \cdot \frac{p \cdot r}{p + r} \tag{31}$$

The F-measure can be maximum, i.e., $F_1 = 1$, only if both precision and recall are maximum, i.e., $p = 1$ & $r = 1$. Thus, F_1 can have a high value only when both p and r are high. Thus, it measures the effectiveness of an algorithm. In essence, the effectiveness of an algorithm is directly proportional to the values of p, r, and F_1.

In this work, the performance of the proposed methodology has been evaluated on the basis of three window sizes, i.e., $ws = \{\pm 5, \pm 7, \pm 10\}$. The confusion matrices has been created for $|S(W)| = 3$ and $|S(W)| = 4$. The values of $\langle p, r, F_1 \rangle$ have been calculated for $\mathrm{W}_i\mathrm{S}_j$, where i indicates word index and j the sense index. These values have been presented in Tables 1, 2 and 3.

F_1 is the harmonic mean of p and r and thus considers both fp-s and fn-s. It is good for measurement when the difference between the number of fp-s and fn-s is very high, i.e., there is unequal sense distribution. It is influenced by majority voting and assumes p and r distributions to be alike. However, in the case of a fair distribution, it can not be as easily evaluated as accuracy. If the number of fp-s and fn-s are relatively close, accuracy is the best measurement.

5.1.4 Accuracy

Accuracy signifies the number of proper sense disambiguations among the entire number of sense instances of a particular target word, and is a good measurement when a dataset is unbiased. In this work, accuracy has been calculated according to (32).

$$acc = \frac{tn + tp}{tn + fp + tp + fn}, \tag{32}$$

where tn (true negative) denote non-class senses that have been correctly classified as non-class, i.e., the predicted sense is false and the actual sense is also false.

The overall accuracy for each polysemous word has been shown in Figs. 5, 6 and 7 for the different window sizes. The graphs in these figures depict the overall accuracy of the hundred polysemous words used in this work.

In analysing the experimental results, this paper has been compared the performance of this algorithm with all other supervised-based Bengali WSD algorithms. Figure 8 refers to accuracy comparison between Biswas et al. [6], Pal et al. [42], and our method based on the five-most common polysemous words, such as:
জল $(Jala)$, মুখ $(Mukha)$, মাথা $(Mātha\bar{a})$, বেল $(Bēla)$, and পাতা and
জল $(Jala)$, মুখ $(Mukha)$, মাথা $(Mātha\bar{a})$, বেল $(Bēla)$, and পাতা $(Pāta\bar{a})$. Due to the robust capability of sense anomaly detection, the accuracy of our method is better compared to Biswas et al. [6], and Pal et al. [42]. Our method emphasises the primary metric

Table 1 $\langle p, r, F_1 \rangle$ values of polysemous words, where W_i indicates word index and S_j indicates sense index

Idx.	$ws = \pm 5$				$ws = \pm 7$				$ws = \pm 10$			
	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4
W1	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$
W2	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 0, 0, 0 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–
W3	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 0, 0, 0 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 0, 0, 0 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 0, 0, 0 \rangle$	–
W4	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$
W5	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 0, 0, 0 \rangle$
W6	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–
W7	$\langle 1, 1, 1 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 0, 0, 0 \rangle$
W8	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–
W9	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–
W10	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–
W11	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 0, 0, 0 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	–
W12	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 0, 0, 0 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 0, 0, 0 \rangle$	–
W13	$\langle 1, 1, 1 \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 1, 1, 1 \rangle$
W14	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–
W15	$\langle 0, 0, 0 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	–	$\langle 0, 0, 0 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–
W16	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 0, 0, 0 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–
W17	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 0, 0, 0 \rangle$
W18	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$
W19	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$
W20	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$
W21	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 0, 0, 0 \rangle$	–	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 0, 0, 0 \rangle$	–

Table 1 (continued)

Idx.	$ws = \pm 5$				$ws = \pm 7$				$ws = \pm 10$			
	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4
W22	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	–	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	–	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	–
W23	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$
W24	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$
W25	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,\frac{1}{2},\frac{2}{3}\rangle$	$\langle 0,0,0\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 0,0,0\rangle$	$\langle 1,\frac{1}{2},\frac{2}{3}\rangle$
W26	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	–	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	–	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	–
W27	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	–	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	–	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	–
W28	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	–	$\langle 0,0,0\rangle$	$\langle 1,1,1\rangle$	$\langle 1,\frac{1}{2},\frac{2}{3}\rangle$	–	$\langle 0,0,0\rangle$	$\langle 1,1,1\rangle$	$\langle 1,\frac{1}{2},\frac{2}{3}\rangle$	–
W29	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 0,0,0\rangle$	$\langle 1,1,1\rangle$	$\langle 1,\frac{1}{2},\frac{2}{3}\rangle$	$\langle 0,0,0\rangle$	$\langle 0,0,0\rangle$	$\langle 1,1,1\rangle$	$\langle 1,\frac{1}{2},\frac{2}{3}\rangle$	$\langle 1,1,1\rangle$
W30	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	–	$\langle 0,0,0\rangle$	$\langle 1,1,1\rangle$	$\langle 1,\frac{1}{2},\frac{2}{3}\rangle$	–	$\langle 1,\frac{1}{2},\frac{2}{3}\rangle$	$\langle 0,0,0\rangle$	$\langle 1,1,1\rangle$	–
W31	$\langle 1,1,1\rangle$	$\langle 1,\frac{1}{2},\frac{2}{3}\rangle$	$\langle 0,0,0\rangle$	–	$\langle 1,1,1\rangle$	$\langle 1,\frac{1}{2},\frac{2}{3}\rangle$	$\langle 0,0,0\rangle$	–	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	–
W32	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	–	$\langle 1,\frac{1}{2},\frac{2}{3}\rangle$	$\langle 0,0,0\rangle$	$\langle 0,0,0\rangle$	–	$\langle 1,\frac{1}{2},\frac{2}{3}\rangle$	$\langle 1,1,1\rangle$	$\langle 0,0,0\rangle$	–
W33	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$
W34	$\langle 0,0,0\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,\frac{1}{2},\frac{2}{3}\rangle$	$\langle 0,0,0\rangle$	$\langle 1,1,1\rangle$	$\langle 1,\frac{1}{2},\frac{2}{3}\rangle$	$\langle 1,1,1\rangle$	$\langle 0,0,0\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,\frac{1}{2},\frac{2}{3}\rangle$
W35	$\langle 0,0,0\rangle$	$\langle 1,\frac{1}{2},\frac{2}{3}\rangle$	$\langle 1,1,1\rangle$	–	$\langle 0,0,0\rangle$	$\langle 1,\frac{1}{2},\frac{2}{3}\rangle$	$\langle 1,1,1\rangle$	–	$\langle 0,0,0\rangle$	$\langle 1,\frac{1}{3},\frac{1}{2}\rangle$	$\langle 0,0,0\rangle$	–
W36	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,\frac{1}{2},\frac{2}{3}\rangle$	$\langle 0,0,0\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,\frac{1}{2},\frac{2}{3}\rangle$	$\langle 0,0,0\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$
W37	$\langle 0,0,0\rangle$	$\langle 1,1,1\rangle$	$\langle 1,\frac{1}{2},\frac{2}{3}\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 0,0,0\rangle$	$\langle 1,\frac{1}{2},\frac{2}{3}\rangle$	$\langle 1,1,1\rangle$	$\langle 0,0,0\rangle$
W38	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	–	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	–	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	–
W39	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,\frac{1}{2},\frac{2}{3}\rangle$	$\langle 0,0,0\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$
W40	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	$\langle 1,1,1\rangle$	–	$\langle 0,0,0\rangle$	$\langle 0,0,0\rangle$	$\langle 1,1,1\rangle$	–	$\langle 0,0,0\rangle$	$\langle 1,\frac{1}{2},\frac{2}{3}\rangle$	$\langle 1,1,1\rangle$	–

Table 2 $\langle p, r, F_1 \rangle$ values of polysemous words, where W indicates word index and S indicates sense index

Idx.	$ws = \pm 5$				$ws = \pm 7$				$ws = \pm 10$			
	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4
W41	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–
W42	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 0, 0, 0 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	–
W43	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–
W44	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$
W45	$\langle 1, 1, 1 \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	–
W46	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–
W47	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–
W48	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 0, 0, 0 \rangle$	–	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 0, 0, 0 \rangle$	–	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 0, 0, 0 \rangle$	–
W49	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 0, 0, 0 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–
W50	$\langle 1, 1, 1 \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	–
W51	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$
W52	$\langle 0, 0, 0 \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	–	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 0, 0, 0 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	–
W53	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 0, 0, 0 \rangle$
W54	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 0, 0, 0 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 0, 0, 0 \rangle$	–
W55	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–
W56	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$
W57	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 0, 0, 0 \rangle$	–	$\langle 0, 0, 0 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 0, 0, 0 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 1, 1, 1 \rangle$	–

Table 2 (continued)

Idx.	$ws = \pm 5$				$ws = \pm 7$				$ws = \pm 10$			
	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4
W58	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$
W59	$\langle 1, \frac{1}{2}, \frac{2}{3}\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 0, 0, 0\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3}\rangle$	$\langle 1, 1, 1\rangle$	$\langle 0, 0, 0\rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3}\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 0, 0, 0\rangle$
W60	$\langle 1, \frac{1}{2}, \frac{2}{3}\rangle$	$\langle 0, 0, 0\rangle$	$\langle 1, 1, 1\rangle$	–	$\langle 1, \frac{1}{2}, \frac{2}{3}\rangle$	$\langle 0, 0, 0\rangle$	$\langle 1, 1, 1\rangle$	–	$\langle 1, \frac{1}{2}, \frac{2}{3}\rangle$	$\langle 0, 0, 0\rangle$	$\langle 1, 1, 1\rangle$	–
W61	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 0, 0, 0\rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3}\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3}\rangle$	$\langle 0, 0, 0\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3}\rangle$	$\langle 0, 0, 0\rangle$
W62	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3}\rangle$	$\langle 0, 0, 0\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3}\rangle$	$\langle 0, 0, 0\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3}\rangle$	$\langle 0, 0, 0\rangle$
W63	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	–	$\langle 1, \frac{1}{2}, \frac{2}{3}\rangle$	$\langle 1, 1, 1\rangle$	$\langle 0, 0, 0\rangle$	–	$\langle 1, \frac{1}{2}, \frac{2}{3}\rangle$	$\langle 1, 1, 1\rangle$	$\langle 0, 0, 0\rangle$	–
W64	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 0, 0, 0\rangle$	$\langle 0, 0, 0\rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3}\rangle$
W65	$\langle 0, 0, 0\rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3}\rangle$	$\langle 1, 1, 1\rangle$	–	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	–	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	–
W66	$\langle 1, 1, 1\rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3}\rangle$	$\langle 0, 0, 0\rangle$	–	$\langle 1, 1, 1\rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3}\rangle$	$\langle 0, 0, 0\rangle$	–	$\langle 1, 1, 1\rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3}\rangle$	$\langle 0, 0, 0\rangle$	–
W67	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$
W68	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3}\rangle$	$\langle 1, 1, 1\rangle$	$\langle 0, 0, 0\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$
W69	$\langle 0, 0, 0\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, \frac{1}{3}, \frac{1}{2}\rangle$	$\langle 0, 0, 0\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 0, 0, 0\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3}\rangle$
W70	$\langle 1, 1, 1\rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3}\rangle$	$\langle 1, 1, 1\rangle$	$\langle 0, 0, 0\rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3}\rangle$	$\langle 0, 0, 0\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 0, 0, 0\rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3}\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$
W71	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$
W72	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	–	$\langle 1, \frac{1}{2}, \frac{2}{3}\rangle$	$\langle 0, 0, 0\rangle$	$\langle 1, 1, 1\rangle$	–	$\langle 1, \frac{1}{2}, \frac{2}{3}\rangle$	$\langle 0, 0, 0\rangle$	$\langle 1, 1, 1\rangle$	–
W73	$\langle 1, 1, 1\rangle$	$\langle 0, 0, 0\rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3}\rangle$	$\langle 0, 0, 0\rangle$	$\langle 1, 1, 1\rangle$	$\langle 0, 0, 0\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3}\rangle$	$\langle 1, 1, 1\rangle$	$\langle 0, 0, 0\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3}\rangle$
W74	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	–	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	–	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	–
W75	$\langle 0, 0, 0\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3}\rangle$	$\langle 1, 1, 1\rangle$	$\langle 0, 0, 0\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3}\rangle$	$\langle 1, 1, 1\rangle$	$\langle 0, 0, 0\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3}\rangle$
W76	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	–	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	–	$\langle 1, \frac{1}{2}, \frac{2}{3}\rangle$	$\langle 1, 1, 1\rangle$	$\langle 0, 0, 0\rangle$	–
W77	$\langle 1, 1, 1\rangle$	$\langle 0, 0, 0\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3}\rangle$	$\langle 0, 0, 0\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3}\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 0, 0, 0\rangle$	$\langle 1, \frac{1}{3}, \frac{1}{2}\rangle$	$\langle 0, 0, 0\rangle$
W78	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	–	$\langle 1, \frac{1}{2}, \frac{2}{3}\rangle$	$\langle 0, 0, 0\rangle$	$\langle 1, 1, 1\rangle$	–	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	–
W79	$\langle 0, 0, 0\rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3}\rangle$	$\langle 1, 1, 1\rangle$	–	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	$\langle 1, 1, 1\rangle$	–	$\langle 0, 0, 0\rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3}\rangle$	$\langle 1, 1, 1\rangle$	–
W80	$\langle 1, \frac{1}{2}, \frac{2}{3}\rangle$	$\langle 0, 0, 0\rangle$	$\langle 0, 0, 0\rangle$	–	$\langle 1, 1, 1\rangle$	$\langle 0, 0, 0\rangle$	$\langle 0, 0, 0\rangle$	–	$\langle 1, 1, 1\rangle$	$\langle 0, 0, 0\rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3}\rangle$	–

Table 3 $\langle p, r, F_1 \rangle$ values of polysemous words, where W_i indicates word index and S_j indicates sense index

Idx.	$ws = \pm 5$				$ws = \pm 7$				$ws = \pm 10$			
	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4
W81	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, 1, 1 \rangle$
W82	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–
W83	$\langle 0, 0, 0 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$
W84	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$
W85	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$
W86	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, \frac{1}{3}, \frac{1}{2} \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, \frac{1}{3}, \frac{1}{2} \rangle$
W87	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–
W88	$\langle 1, 1, 1 \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	–
W89	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$
W90	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–
W91	$\langle 1, 1, 1 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 0, 0, 0 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 0, 0, 0 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–
W92	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, 1, 1 \rangle$
W93	$\langle 0, 0, 0 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, \frac{1}{3}, \frac{1}{2} \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, \frac{1}{3}, \frac{1}{2} \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, \frac{1}{3}, \frac{1}{2} \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 0, 0, 0 \rangle$
W94	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–
W95	$\langle 0, 0, 0 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 0, 0, 0 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	–
W96	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$
W97	$\langle 1, 1, 1 \rangle$	$\langle 1, \frac{1}{2}, \frac{2}{3} \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, \frac{1}{3}, \frac{1}{2} \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 1, \frac{1}{3}, \frac{1}{2} \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 0, 0, 0 \rangle$
W98	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–
W99	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–
W100	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	$\langle 1, 1, 1 \rangle$	–

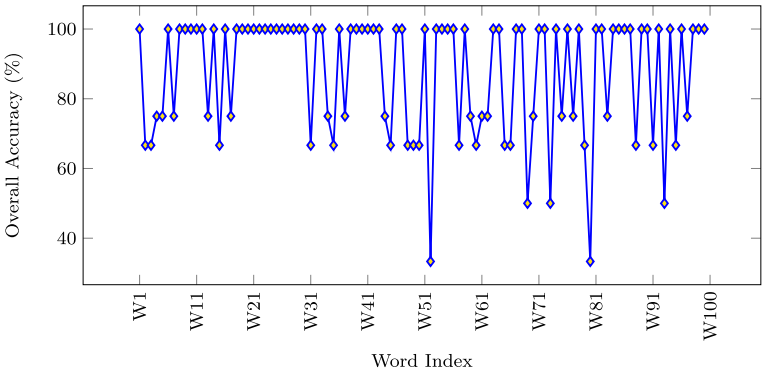

Fig. 5 Accuracy when the window size is ±5

of collecting lexical level features to obtain the most expressive terms in a document. It is achieved 100% accuracy in জল $(Jala)$, মাথা $(Mātha\bar{a})$, বেল $(Bēla)$, and পাতা $(Pāta\bar{a})$.

Conceivably, মাথা $(Mātha\bar{a})$ is the most common word in the list of polysemous words. The various senses of this polysemous word have mentioned in Haque et al. [21], and Pal et al. [41]. Figure 9 is illustrated sense-wise accuracy comparison between Haque et al. [21], Pal et al. [41], and our method. The sense-detection accuracy of our method is 100% in all senses. However, the Naïve Bayes algorithm suffers from less accurate and zero reliance problems in Pal et al. [41]. On the other hand, attainment rate of Haque et al. [21] depends on length of a sentence.

6 Results analysis

In this section, qualitative observations have been presented regarding the performance of the proposed algorithm. The obtained results have been analysed based on various perspectives in interrelatedness with computational linguistics. The effectiveness of the algorithm has been determined using a number of factors and exhaustive numerical breakdown.

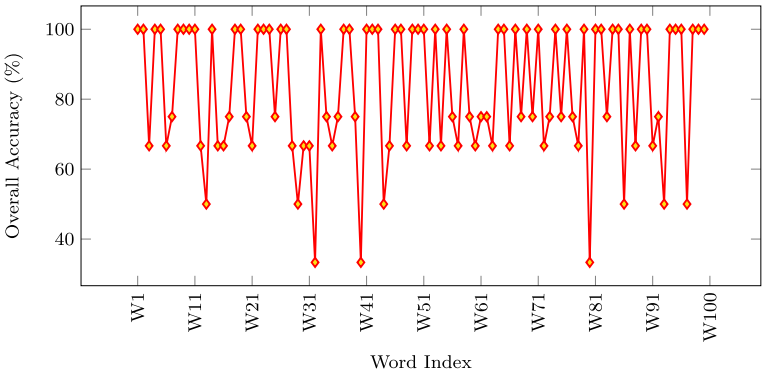

Fig. 6 Accuracy when the window size is ±7

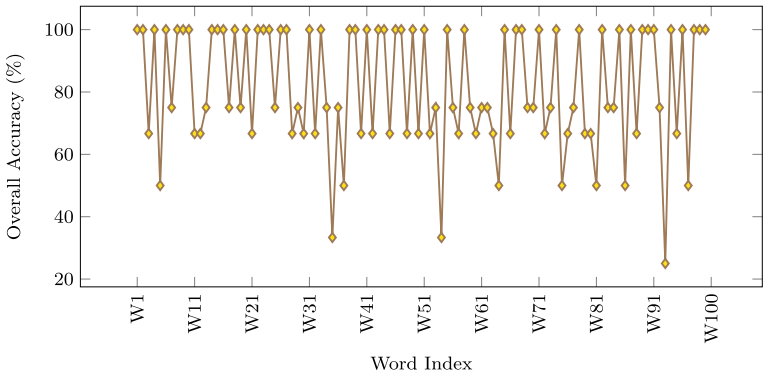

Fig. 7 Accuracy when the window size is ±10

6.1 Result analysis based on average accuracy values

In Section 5, the overall accuracy of each polysemous word have been presented using Figs. 6, 7, 8, 9 and 10. The average accuracy, on the other hand, provides an overall evaluation of the performance of the proposed algorithm, and has been calculated as follows:

$$avg\_acc = \frac{1}{N}\sum_{i=1}^{N} acc_i, \tag{33}$$

where i represents the word index and the total number of polysemous words present in the dataset is $N = 100$. The avg_acc of the proposed algorithm has been measured based on the following three parameters:

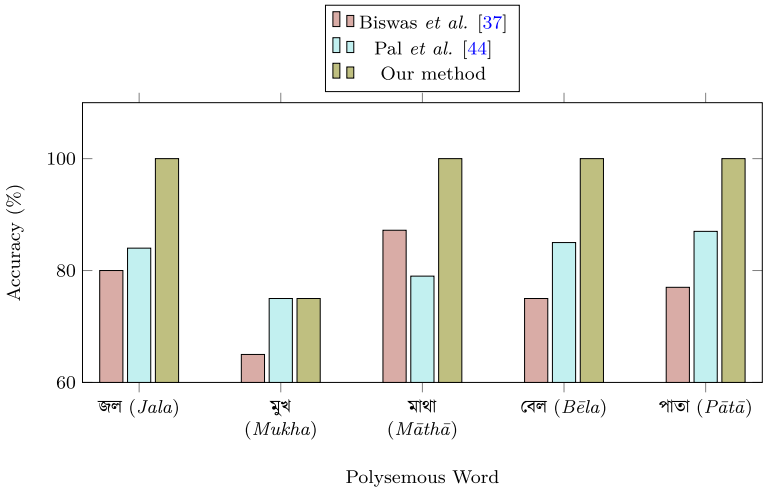

Fig. 8 Polysemous word-based accuracy comparison between Biswas et al. [6], Pal et al. [42], and our method

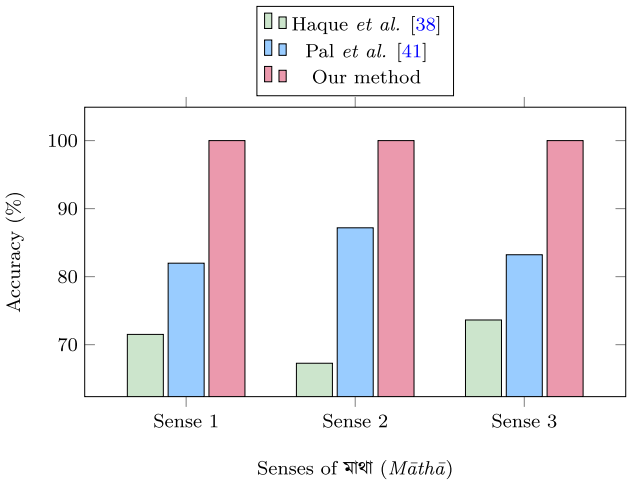

Fig. 9 Sense-wise accuracy comparison between Haque et al. [21], Pal et al. [41], and our method

- $ws = \pm 5$: In this case, we have considered five predecessor and five successor words neighbouring the target polysemous word. Thus, there are eleven words including the target word, making this the smallest-sized window in our experimentation.
- $ws = \pm 7$: In this case, the seven words before and after the target word has been considered. This is a medium-sized window, consisting fifteen words including the target word.
- $ws = \pm 10$: In this case, the number of words considered on both sides of a target word is ten. This is the largest-sized window with twenty-one words including the target word.

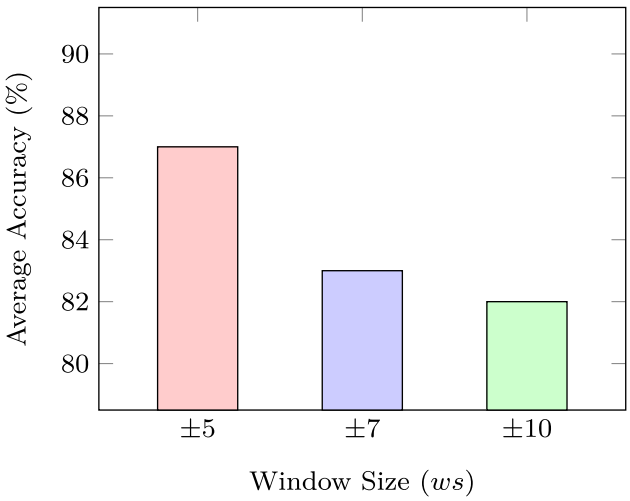

Fig. 10 Average accuracy avg_acc of each window sizes for all the polysemous words

The average accuracy of the proposed algorithm with respect to the moveable window sizes have been presented in Fig. 10. The maximum accuracy of 87% has been achieved for $ws = \pm 5$. In addition, the overall accuracy for $ws = \pm 7$ and $ws = \pm 10$ are also good. The average accuracy achieved by our proposed algorithm is 84%.

Subsequently, this paper has been compared avg_acc of our method with state-of-the-art techniques. Figure 11 is illustrated the avg_acc comparison between Haque et al. [21], Pandit et al. [47], Biswas et al. [6], Pal et al. [43], Pal et al. [42], Pal et al. [44], and our method. It has been demonstrated that our method has scored the best of all the state-of-art-the techniques. The avg_acc of our method is always high for $ws = \pm 5$, $ws = \pm 7$, and $ws = \pm 10$. Due to the SLH balance in all senses, the accuracy of our method is always at the top regardless of the size of the window. PSC is able to produce a good singularity coherence factor for all senses.

In contrast, Haque et al. [21] is not considered syntactic features of a sentence and the performance of this algorithm is the lowest. Pandit et al. [47] suffers from spatial problems and does not work on high-dimensional datasets with a number of senses greater than three. In Biswas et al. [6], Bayes probability theorem does not work well if the number of senses in a polysemous word is increased. Due to the absence of a proper text legitimization algorithm for Bengali, Pal et al. [43] requires an advanced version of stemming for verbs. On the other hand, Pal et al. [42] achieved better results using Naïve Bayes classification strategy. Naïve Bayes enables very fast and conditional independence acquisition. Also, Pal et al. [44] attained up to the mark accuracy level using bootstrapping comprising lemmatization attribute. Contextual expansion of sentences is good for PCA.

6.2 Result analysis based on misclassification rate

WSD is a challenging problem in a morphologically complex language like Bengali. The *misclassification rate/cost* or error rate or error of class probability, M is dependant on the number of entities that are classified by the method in a different sense. The objective of any algorithm is to minimise M, which is the combination of fp and fn, i.e., negative senses being predicted as positive. M has been calculated in this work according to the (34).

$$M = 1 - a \tag{34}$$

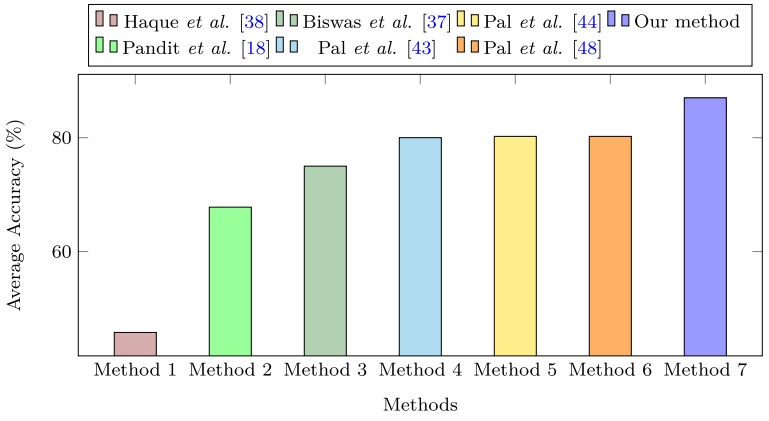

Fig. 11 Average accuracy avg_acc comparison between Haque et al. [21], Pandit et al. [47], Biswas et al. [6], Pal et al. [43], Pal et al. [42], Pal et al. [44], and our method

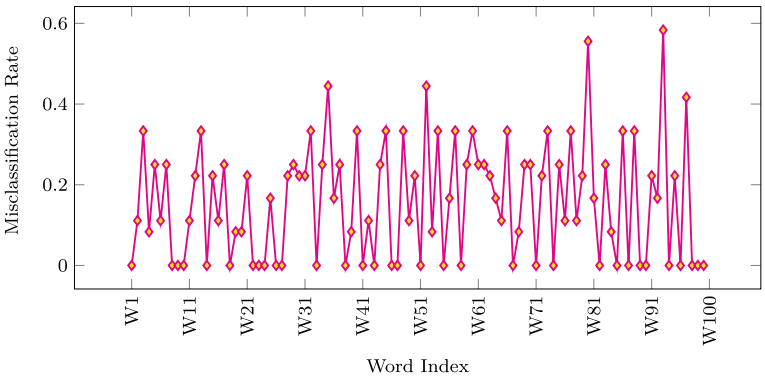

Fig. 12 Misclassification rate of all polysemous words

The misclassification rate of this algorithm has been plotted in Fig. 12. Polysemy is a prevalent characteristic of a language, and here, we have found that high-frequency words have low polysemous properties compared to less frequent words. This property is known as *polysemy advantage*. The types of polysemy of the target words has been classified according to Table 4. The polysemy property of a polysemous word is directly related to the misclassification rate.

The proposed algorithm has facilitated the classification of a target word according to its polysemy property, which in turn, is directly interlinked with misclassification rate. In our experimental study, it has been found that a target word, which has the polysemy advantage property, has less misclassification rate. High polysemy words have high misclassification rate compared to low polysemous words.

6.3 Result analysis based on the positive and negative compilation sets

The compilation of outcomes is critical for the determination of the predictive power of any algorithm. Consequently, the post-processing phase is crucial for any WSD algorithm. Due to the deficiency of lexical resources in Bengali, the output compilation becomes very beneficial to evaluate the performance of any proposed framework. Knowledge mining is thus a potential major hurdle in solving a WSD problem.

A polysemy exists in a word or phrase with dissimilar items but correlated senses. The assessment of polysemy is an ambiguous concept, related to the affinity of words. As a result, in this work, the output compilation has been summarised into a positive (+) and negative (-) result set. A positive result set represents the fact that the proposed methodology has disambiguated a polysemous word correctly. On the other hand, a negative result set

Table 4 Types of polysemy with respect to the misclassification rate

Types of polysemy	Misclassification rate
High polysemy	0.41 – 0.60
Mid Polysemy	0.21 – 0.40
Low Polysemy	0.00 – 0.20

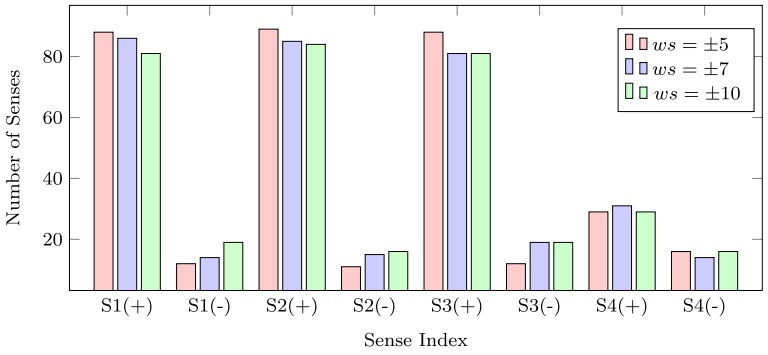

Fig. 13 Compilation of positive (+) and negative (-) result set

denotes that the proposed methodology has been unable to disambiguate a polysemous word correctly.

The analysis of the obtained results based on positives and negatives has been done by considering all the senses of a polysemous word with respect to the various window sizes. Figure 13 presents this positive and negative result set compilation with respect to four senses and three window sizes.

6.4 Result analysis based on the distribution of accuracy

The structural representation of the results at various levels of compilation has been presented in Fig. 14. The performance of our proposed algorithm is remarkable at the top level. On the other hand, the smaller entries indicate the robustness of the proposed algorithm at the lowest level. Our proposed method achieves a high-level recognition accuracy for $ws = \pm 5$. It can also be observed from Fig. 14 that the performance of the proposed algorithm degrades gradually with increase in the window size ws.

6.5 Result analysis based on linear regression

Linear regression is a basic concept in statistics and machine learning. Regression is generally performed to analyse the prediction capability of any proposed algorithm. In this

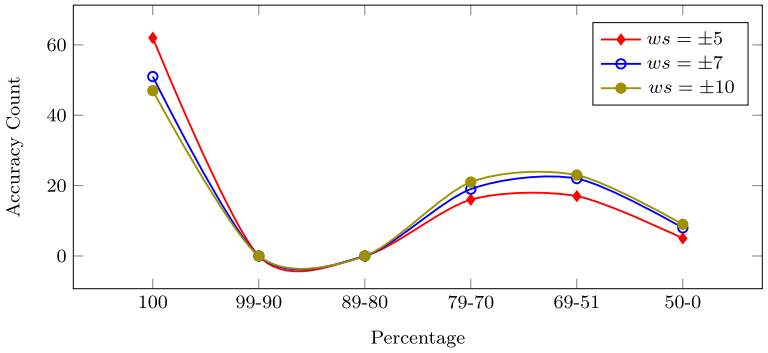

Fig. 14 Distribution of accuracy count over various window sizes

paper, simple linear regression has been used, which consists of one dependent and one independent variable, calculated using (35).

$$y = \beta_0 \cdot x + \beta_1, \tag{35}$$

where y and x indicate count and accuracy, respectively. Optimum values have been selected for β_0 and β_1 to minimise the error. Figure 15 represents linear regression with respect to accuracy and count.

The three separate linear regression equations correspond to the three window sizes. Additionally, correlation coefficient has been calculated to evaluate the strength of the relationship between the accuracy and count. The range of this variable always lies between -1 and $+1$, where -1 indicates strong negative, $+1$ indicates strong positive, and 0 indicates no or weak relationship. The correlation coefficient has been calculated as follows:

1. Linear Regression for $ws = \pm 5$, where correlation coefficient r_5 is 0.9081.
2. Linear Regression for $ws = \pm 7$, where correlation coefficient r_7 is 0.9424.
3. Linear Regression for $ws = \pm 10$, where correlation coefficient r_{10} is 0.9614.

From this post-experimental analysis, it has been observed that the values of $\{r_5, r_7, r_{10}\}$ lie in the range [0.9, 1.0]. The obtained results show a very high positive correlation [33], and prove the robust nature of the proposed algorithm.

6.6 Result analysis based on parts of speech (POS)

In corpus linguistics, word-category disambiguation is also known as parts of speech or POS based sense tagging, and is very useful in the case of an applied linguistic engineering framework. Hence, in this paper, the obtained experimental results have been analysed based on grammatical classification. This analysis has been graphically presented using Fig. 16., which shows the average accuracy (for $ws = \{\pm 5, \pm 7, \pm 10\}$) achieved by the proposed algorithm for the six parts of speech. It can be clearly observed from Fig. 16 that the proposed algorithm can make accurate predictions in the cases of adjective, verb, noun, and interjection based polysemous words. It is worth mentioning here that our proposed algorithm has succeeded in achieving an accuracy of 100% in the case of interjection based polysemous words. The average accuracy of these four parts of speech is 89.28%.

On the other hand, the performance of our proposed algorithm is moderate in the case of adverb based polysemous words, i.e., an average accuracy of 61.9%. The function of an adverb is to describe a verb, adjective, or another adverb in various forms, such as time, place, manner, frequency, quantity, affirmation, negation, etc. It can be positioned after a

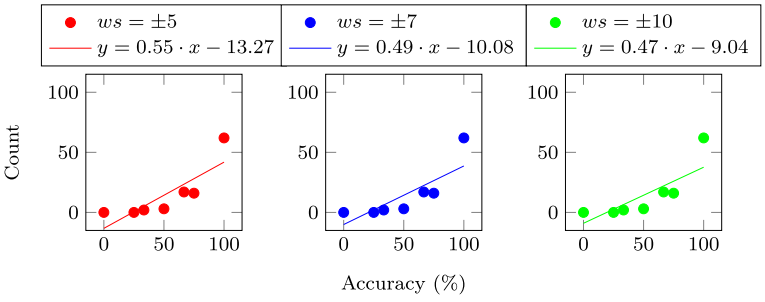

Fig. 15 Generation of liner regression of each window size

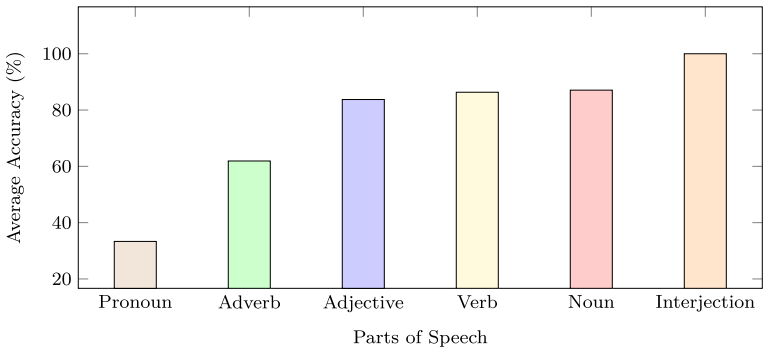

Fig. 16 Average accuracy of all window sizes according to the part of speech

verb, or at the starting of a sentence, or before a verb, adjective or another adverb. This variable positioning may be the cause behind the moderate performance of the proposed algorithm in this case.

Finally, the performance of the proposed algorithm is very poor in the case of pronoun based polysemous words, i.e., 33.33%. Due to the non-availability of root lexicon and suffix lexicon in Bengali, it is very challenging to disambiguate senses in the case of pronoun-based polysemous words.

6.7 Result analysis based on miscellaneous facts

In this section, some miscellaneous facts regrading the result set have been presented, which are very interesting. They have been enumerated as follows:

(a) The proposed algorithm has also been used named entity disambiguation; e.g., in the following case, 100% accuracy has been achieved. For example polysemous word: নাম $(N\bar{a}ma)$ [42].
(b) The proposed has been successful in achieving the maximum possible accuracy of 100% in various polysemous words that consist of inter-domain information. For example polysemous word: জল $(Jala)$ [40].
(c) The proposed algorithm has also succeeded in achieving a 100% accuracy in various POS-based tagging. For example polysemous word: দিন $(Dina)$ [39].

6.8 Result analysis based on arabic language

Figure 17 depicts average accuracy avg_acc comparison between Zouaghi et al. [69], Merhbene et al. [31], Hadni et al. [20], Merhbene et al. [32], Menai et al. [30], and our method. Zouaghi et al. [69] performed the WSD task by generating equivalence connections between the senses of a polysemous word using the Lesk algorithm. However, this algorithm is very erogenous for the multiple sense definitions of the polysemous word, when context is limited. Merhbene et al. [31] applied *term frequency inverse document frequency* (TFIDF)-based string matching algorithm using linguistics cohesiveness grade. However, the time complexity of this string matching algorithm is very high and it is unable to extract the original words from the context-generated words, which is also a tedious task. Hadni et al. [20] encapsulated *Naïve Bayes* (NB) with SVM-based technique, especially for Arabic WSD.

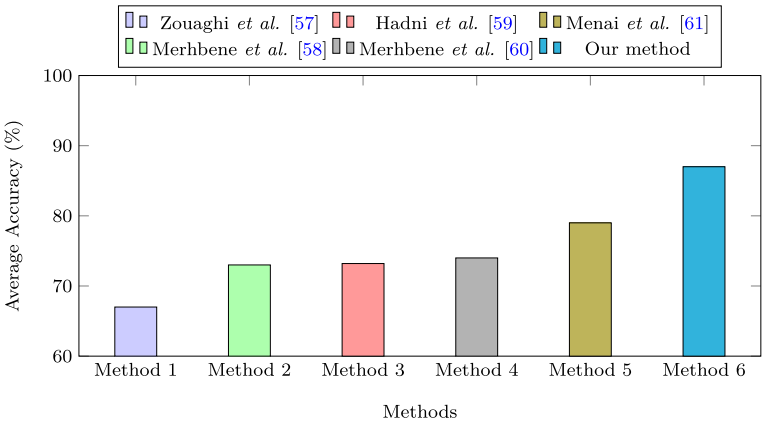

Fig. 17 Average accuracy avg_acc comparison between Zouaghi et al. [69], Merhbene et al. [31], Hadni et al. [20], Merhbene et al. [32], Menai et al. [30], and our method

This algorithm [20] selects the Chi-Square measure for the feature set and indicates the closest realization for the unambiguous laxes. However, the performance of this method drops significantly when it is tested on large datasets when the number of senses of a polysemous word is greater than three. Merhbene et al. [32] employed weighted oriented graphs to match the syntactic form of a given word. This algorithm [32] exploits sense clustering to represent sentences where the meanings of a particular sense are grouped together in similar clusters. However, the ranking method is very difficult in terms of majority voting for the most syntactically similar terms. Menai et al. [30] developed evolutionary approach to generate fitness function for testing data. However, word sense affinity of fitness function is not worked well in case of Lesk and modified Lesk algorithm. Thus, the performance of this approach is not up to the mark.

6.9 Result analysis based on hindi language

Figure 18 illustrates average accuracy avg_acc comparison between Sinha et al. [61], Singh et al. [60], Vishwakarma et al. [66], Bala et al. [4], Yadav et al. [67], Tayal et al. [64], and our method. Sinha et al. [61] captured semantic relations of correlated words in the context. This algorithm [61] is based on percentage similarity between sense overlaps and is unable to capture morphological inconsistencies. It does not work well for highly overlapping words. It has the lowest overall accuracy. In the same way as Sinha et al. [61], Singh et al. [60] applied direct overlapping method. This algorithm is based on the Leacock-Chodorow quota of semantic similarity between senses. Nevertheless, this algorithm is only applicable to nouns and the performance of other POS is not desirable. Vishwakarma et al. [66] developed *depth first search* (DFS)-based construction technique of sentence-to-sentence assumption. However, it is not suitable for the shortest distance of a graph node. Bala et al. [4] considered information overlap of multiple senses of a polysemous word. However, this method does not work properly due to selection constraints. Yadav et al. [67] applied connected *if-then* rules of mining to find minimum threshold. However, this algorithm does not fit when number of senses of a polysemous word is drastically increased. Tayal et al. [64] produced semantic bag of hyperspace analogue of dimension reduction matrix. However, fuzzy measure does not work well when fuzziness ambiguity is increased.

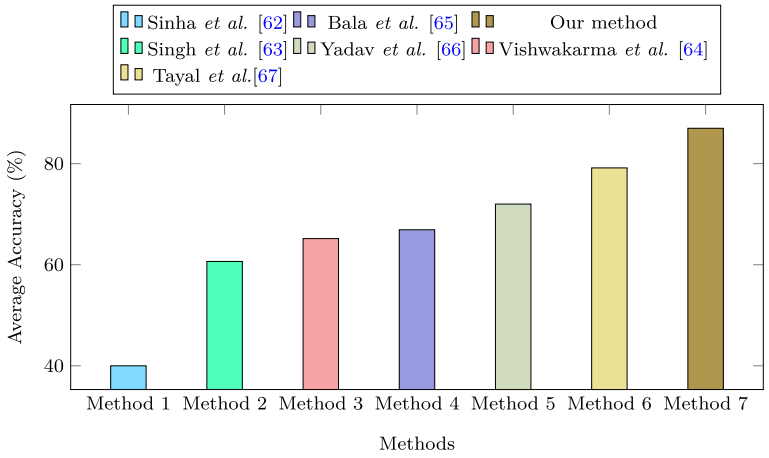

Fig. 18 Average accuracy avg_acc comparison between Sinha et al. [61], Singh et al. [60], Vishwakarma et al. [66], Bala et al. [4], Yadav et al. [67], Tayal et al. [64], and our method

6.10 Result analysis based on very low-resource languages

Figure 19 depicts average accuracy avg_acc comparison between Punjabi, Nepali [54], Assamese [55], Manipuri, Kannada [49], and our method. Rana et al. worked on overlap-based approach on Punjabi WSD for enumerating overlap between neighbouring terms. However, this algorithm, like Pal et al. [45] and Zouaghi et al. [69], suffers overlap sparsity problem. Due to the inability to clearly identify the context basket and the sense basket, its retrieval accuracy is the lowest level. Like Vishwakarma et al. [66], Roy et al. [54] performed on Nepali WSD using distance computation of graph semantic approach. This algorithm [54] does not work well on all POS except nouns. It suffers sparse overlap problem, like Rana et al., Pal et al. [45], and Zouaghi et al. [69]. Sarmah et al. [55] worked on decision tree based Assamese WSD using gain ration and splitting parameter of polysemous word. Nevertheless, this algorithm, like Hadni et al. [20] does not capable to capture

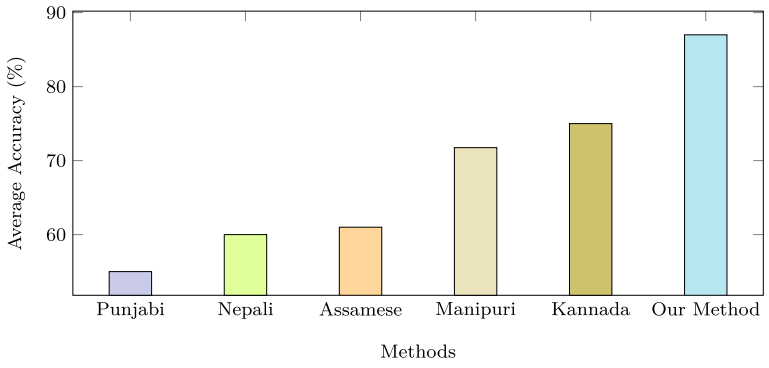

Fig. 19 Average accuracy avg_acc comparison between Punjabi, Nepali [54], Assamese [55], Manipuri, Kannada [49], and our method

semantic level disambiguation. Singh et al., performed Manipuri WSD using decision tree, like Roy et al. [54], Rana et al., Pal et al. [45], and Zouaghi et al. [69]. This algorithm suffers from typical positional and framework based feature problems. Parameswarappa et al. [49] performed Kannada WSD using decision list of uni-, bi-, and trigrams of single sense with respect to each collocation. However, this algorithm is not suitable for compound words.

7 Conclusion

In this paper, a robust technique has been proposed for sense disambiguation of Bengali polysemous words. It has been shown here that the lexeme connexion measure of cohesive lexical ambiguity revealing factor can be applied to a WSD system for eliminating the major shortcomings of Bengali WSD approaches. Validation has been performed by testing the proposed algorithm against a new dataset containing hundred Bengali polysemous words. Each such polysemous word consists of either three or four senses.

Due to the inherent morphological complexity of Bengali, the overall performance evaluation is directly influenced by the size of the context window. A context window has been created on the basis of the target word. The size of the context window depends on a number of parameters, such as the number of senses, amount of training and testing data, etc. Half of the total data has been used for training, while the other half for testing. The proposed algorithm has been evaluated on context windows of three different sizes, viz. five, seven, and ten. The context windows of size five, seven, and ten, respectively contain eleven, fifteen, and twenty-one words including the target word. The performance evaluation has been done based on the following metrics: precision, recall, F-measure, and accuracy, for three different sizes of the context window.

It has been observed from the obtained results that a relatively short-sized context window is more effective because it can present contextual information more precisely. The robustness of the proposed algorithm has been proved at various levels of our study, such as the result analysis based on the average accuracy values, misclassification rate, compilation of positive set and negative set, accuracy distribution, linear regression, and part-of-speech tagging.

WSD poses an interesting problem for linguistic researchers worldwide. WSD has been performed on various most-spoken languages. The success rate of developed algorithms depends on various parameters, like morphological complexity of the language, variations between dictionary definitions, inter-evaluator discrepancies, distinct sub-meanings of senses, etc. The performance of proposed WSD techniques also depends on the dissertation structure of separation and coherence relationships in the middle of words. In this work, an endeavour has been undertaken to overcome these barriers in most of the cases. Due to the robustness of the proposed algorithm, the lexeme connexion measure of cohesive lexical ambiguity revealing factor can also be applied to cross-lingual WSD, and multilingual WSD.

For future opportunities, linguistic tools and resources need to be developed in the case of Bengali. More comprehension is needed to find lexical and semantic similarities between the two sentences. A strong mathematical formula is needed to find the heart of the subject matter in a more conceptual way.

Data Availability The datasets generated and/or analysed during the current study are available in the "Kaggle" repository, https://www.kaggle.com/dsv/3985193 with DOI: 10.34740/KAGGLE/DSV/3985193.

Declarations

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

References

1. Agirre E, De Lacalle OL (2007) Ubc-alm: combining k-nn with svd for wsd. In: Proceedings of the fourth international workshop on semantic evaluations (SemEval-2007), pp 342–345
2. Agirre E, Edmonds P (2007) Word sense disambiguation: algorithms and applications, vol 33. Springer science & business media
3. Anand Kumar M, Rajendran S, Soman KP (2014) Tamil word sense disambiguation using support vector machines with rich features. Int J Appl Eng Res 9(20):7609–20
4. Bala P (2013) Knowledge based approach for word sense disambiguation using hindi wordnet. Int J Eng Sci 2(4):36–41
5. Banerjee S, Naskar SK, Bandyopadhyay S (2014) Bfqa: a bengali factoid question answering system. In: International conference on text, speech, and dialogue. Springer, pp 217–224
6. Biswas M, Sharif O, Hoque MM (2021) An empirical framework for bangla word sense disambiguation using statistical approach. In: International conference on machine learning and big data analytics. Springer, pp 22–33
7. Bonami O, Boyé G, Dal G, Giraudo H, Namer F (2018) The lexeme in descriptive and theoretical morphology. Language science press
8. Cohn T (2003) Performance metrics for word sense disambiguation. In: Proceedings of the australasian language technology workshop, vol 2003, pp 86–93
9. Dang HT, Chia C-Y, Palmer M, Chiou F-D (2002) Simple features for chinese word sense disambiguation. In: Proceedings of the 19th international conference on computational linguistics. Association for computational linguistics, vol 1, pp 1–7
10. Das D, Bandyopadhyay S (2009) Word to sentence level emotion tagging for bengali blogs. In: Proceedings of the ACL-IJCNLP 2009 conference short papers, pp 149–152
11. Das A, Bandyopadhyay S (2009) Subjectivity detection in english and bengali: a crf-based approach. Proceeding of ICON
12. Das A, Bandyopadhyay S (2010) Opinion-polarity identification in bengali. In: International conference on computer processing of oriental languages, pp 169–182
13. Das A, Sarkar S (2013) Word sense disambiguation in bengali applied to bengali-hindi machine translation. In: Proc of international conference on natural language processing (ICON), vol 10, pp 20–28
14. Das Dawn D, Khan A, Shaikh SH, Pal RK (2022) A dataset for evaluating Bengali word sense disambiguation techniques. J Ambient Intell Humanized Comput 1–30
15. Dawn DD, Shaikh SH, Pal RK (2020) A comprehensive review of bengali word sense disambiguation. Artif Intell Rev 53(6):4183–4213
16. Dey A (2020) Attention based lstm cnn framework for sentiment extraction from bengali texts. In: 2020 11th International conference on electrical and computer engineering (ICECE). IEEE, pp 226–229
17. Dhungana UR, Shakya S (2014) Word sense disambiguation in nepali language. In: 2014 fourth international conference on digital information and communication technology and its applications (DICTAP). IEEE, pp 46–50
18. Ekbal A, Haque R, Bandyopadhyay S (2007) Bengali part of speech tagging using conditional random field. In: Proceedings of seventh international symposium on natural language processing (SNLP2007), pp 131–136
19. Florian R, Wicentowski R (2002) Unsupervised Italian word sense disambiguation using wordnets and unlabeled corpora. In: Proceedings of the ACL-02 workshop on Word sense disambiguation: recent successes and future directions, pp 67–73
20. Hadni M, Ouatik SEA, Lachkar A (2016) Word sense disambiguation for arabic text categorization. Int Arab J Inf Technol 13(1A):215–222
21. Haque A, Haque MM (2016) Bangla word sense disambiguation system using dictionary based approach. ICAICT, Bangladesh
22. Hoste V, Daelemans W, Hendrickx I, Bosch AVD (2002) Dutch word sense disambiguation: optimizing the localness of context. In: Proceedings of the ACL-02 workshop on word sense disambiguation: recent successes and future directions. Association for computational linguistics, vol 8, pp 61–66
23. Islam M, Islam M, Mohammad Masum AK, Abujar S, Hossain SA et al (2021) Abstraction based bengali text summarization using bi-directional attentive recurrent neural networks. In: Emerging technologies in data mining and information security. Springer, pp 317–327

24. Joachims T (1996) A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. Technical report, Carnegie-Mellon Univ Pittsburgh PA dept of computer science
25. Korenius T, Laurikkala J, Järvelin K, Juhola M (2004) Stemming and lemmatization in the clustering of finnish text documents. In: Proceedings of the thirteenth ACM international conference on information and knowledge management, pp 625–633
26. Lesk M (1986) Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In: Proceedings of the 5th annual international conference on Systems documentation, pp 24–26
27. Liu H, Johnson SB, Friedman C (2002) Automatic resolution of ambiguous terms based on machine learning and conceptual relations in the umls. J Am Med Inform Assoc 9(6):621–636
28. Màrquez L, Escudero G, Martínez D, Rigau G (2007) Supervised corpus-based methods for wsd. In: Word sense disambiguation. Springer, pp 167–216
29. McCallum A, Nigam K et al (1998) A comparison of event models for naive bayes text classification. In: AAAI-98 workshop on learning for text categorization. Citeseer, number 1, pp 41–48
30. Menai MEB (2014) Word sense disambiguation using an evolutionary approach. Informatica, vol 38(3)
31. Merhbene L, Zouaghi A, Zrigui M (2010) Ambiguous arabic words disambiguation. In: 2010 11th ACIS international conference on software engineering, artificial intelligence, networking and parallel/distributed computing. IEEE, pp 157–164
32. Merhbene L, Zouaghi A, Zrigui M (2013) A semi-supervised method for arabic word sense disambiguation using a weighted directed graph. In: Proceedings of the sixth international joint conference on natural language processing, pp 1027–1031
33. Mukaka MM (2012) Statistics corner: a guide to appropriate use of correlation coefficient in medical research malawi medical journal
34. Murata M, Utiyama M, Uchimoto K, Ma Q, Isahara H (2001) Japanese word sense disambiguation using the simple bayes and support vector machine methods. In: Proceedings of SENSEVAL-2 second international workshop on evaluating word sense disambiguation systems, pp 135–138
35. Navigli R (2009) Word sense disambiguation: a survey. ACM Comput Surveys (CSUR) 41(2):1–69
36. Navigli R, Velardi P (2005) Structural semantic interconnections: a knowledge-based approach to word sense disambiguation. IEEE Trans Pattern Anal Mach Intell 27(7):1075–1086
37. Ng HT, Lee HB (1996) Integrating multiple knowledge sources to disambiguate word sense: an exemplar-based approach. In: Proceedings of the 34th annual meeting on association for computational linguistics. Association for computational linguistics, pp 40–47
38. Pal AR, Kundu A, Singh A, Shekhar R, Sinha K (2015) A hybrid approach to word sense disambiguation combining supervised and unsupervised learning. arXiv:1611.01083
39. Pal AR, Saha D (2016) Word sense disambiguation in bengali: an auto-updated learning set increases the accuracy of the result. In: Information systems design and intelligent applications. Springer, pp 423–430
40. Pal AR, Saha D (2019) Word sense disambiguation in bengali language using unsupervised methodology with modifications. Sādhanā 44(7):168
41. Pal AR, Saha D, Dash NS (2015) Automatic classification of bengali sentences based on sense definitions present in bengali wordnet. arXiv:1508.01349
42. Pal AR, Saha D, Dash NS, Naskar SK, Pal A (2019) A novel approach to word sense disambiguation in bengali language using supervised methodology. Sādhanā 44(8):1–12
43. Pal AR, Saha D, Naskar S, Dash NS (2015) Word sense disambiguation in bengali: a lemmatized system increases the accuracy of the result. In: 2015 IEEE 2nd international conference on recent trends in information systems (ReTIS). IEEE, pp 342–346
44. Pal AR, Saha D, Naskar SK, Dash NS (2021) In search of a suitable method for disambiguation of word senses in bengali. Int J Speech Technol 24(2):439–454
45. Pal AR, Saha D, Pal A (2017) A knowledge based methodology for word sense disambiguation for low resource language. Adv Computat Sci Technol 10(2):267–283
46. Palanati DP, Kolikipogu R (2013) Decision list algorithm for word sense disambiguation for telegu natural language processing. Int J Electron Commun Comput Eng 4(6):176–180
47. Pandit R, Naskar SK (2015) A memory based approach to word sense disambiguation in bengali using k-nn method. In: 2015 IEEE 2nd international conference on recent trends in information systems (reTIS). IEEE, pp 383–386
48. Parameswarappa S, Narayana VN (2011) Kannada word sense disambiguation using association rules. In: International conference on computing and communication systems. Springer, pp 47–56
49. Parameswarappa S, Narayana VN, Yarowsky D (2013) Kannada word sense disambiguation using decision list. Int J Emerging Trends Technol Comput Sci (IJETTCS) 2(3):272–278
50. Pedersen T (2007) Unsupervised corpus-based methods for wsd. In: Word sense disambiguation. Springer, pp 133–166

51. Rana P, Kumar P (2015) Word sense disambiguation for punjabi language using overlap based approach. In: Advances in intelligent informatics. Springer, pp 607–619
52. Resnik P (1995) Using information content to evaluate semantic similarity in a taxonomy. arXiv:cmp-lg/9511007
53. Ritter A, Etzioni O et al (2010) A latent dirichlet allocation method for selectional preferences. In: Proceedings of the 48th annual meeting of the association for computational linguistics. Association for computational linguistics, pp 424–434
54. Roy A, Sarkar S, Purkayastha BS (2014) Knowledge based approaches to nepali word sense disambiguation. Int J Natural Lang Comput (IJNLC) 3(3):51–63
55. Sarmah J, Sarma SK (2016) Decision tree based supervised word sense disambiguation for assamese. Int J Comput Appl 141(1):42–48
56. Sengupta S, Pandit R, Mitra P, Naskar SK, Sardar MM (2019) Word sense induction in bengali using parallel corpora and distributional semantics. J Intell Fuzzy Syst 36(5):4821–4832
57. Sharma DK et al (2015) A comparative analysis of hindi word sense disambiguation and its approaches. In: International conference on computing, communication & automation. IEEE, pp 314–321
58. Sidorov G, Gelbukh A (2001) Word sense disambiguation in a spanish explanatory dictionary. In: Proceedings of TALN, pp 398–402
59. Singh RL, Ghosh K, Nongmeikapam K, Bandyopadhyay S (2014) A decision tree based word sense disambiguation system in manipuri language. Adv Comput 5(4):17
60. Singh S, Singh VK, Siddiqui TJ (2013) Hindi word sense disambiguation using semantic relatedness measure. In: International workshop on multi-disciplinary trends in artificial intelligence. Springer, pp 247–256
61. Sinha M, Kumar M, Pande P, Kashyap L, Bhattacharyya P (2004) Hindi word sense disambiguation. In: International symposium on machine translation, natural language processing and translation support systems, Delhi, India
62. Sruthi Sankar KP, Reghu Raj PC, Jayan V (2016) Unsupervised approach to word sense disambiguation in malayalam. Proced Technol 24:1507–1513
63. Sultana M, Chakraborty P, Choudhury T (2022) Bengali abstractive news summarization using seq2seq learning with attention. In: Cyber intelligence and information retrieval. Springer, pp 279–289
64. Tayal DK, Ahuja L, Chhabra S (2015) Word sense disambiguation in hindi language using hyperspace analogue to language and fuzzy c-means clustering. In: Proceedings of the 12th international conference on natural language processing, pp 49–58
65. Turian J, Ratinov L, Bengio Y (2010) Word representations: a simple and general method for semi-supervised learning. In: Proceedings of the 48th annual meeting of the association for computational linguistics. Association for computational linguistics, pp 384–394
66. Vishwakarma SK, Vishwakarma CK (2012) A graph based approach to word sense disambiguation for hindi language. Int J Sci Res Eng Technol (IJSRET) Vol 1:313–318
67. Yadav P, Vishwakarma S (2013) Mining association rules based approach to word sense disambiguation for hindi language. Int J Emerging Technol Adv Eng 3(5):470–473
68. Zipf GK (1949) Human behavior and the principle of least effort. Adison-Wesley Press
69. Zouaghi A, Merhbene L, Zrigui M (2011) Word sense disambiguation for arabic language using the variants of the lesk algorithm. WORLDCOMP 11:561–567
70. Zungre NB, Dhopavkar GM (2016) Sense disambiguation for marathi language words using decision graph method. In: 2016 World conference on futuristic trends in research and innovation for social welfare (startup conclave). IEEE, pp 1–6

Affiliations

Debapratim Das Dawn[1] · Abhinandan Khan[1,2] · Soharab Hossain Shaikh[3] · Rajat Kumar Pal[1]

Abhinandan Khan
khan.abhinandan@gmail.com

Rajat Kumar Pal
pal.rajatk@gmail.com

1 Department of Computer Science and Engineering, University of Calcutta, Acharya Prafulla Chandra Roy Shiksha Prangan, JD-2, Sector-III, Saltlake, Kolkata, 700106, India

2 Product Development and Diversification, ARP Engineering, 147 Nilgunj Road, Kolkata, 700056, India

3 Department of Computer Science and Engineering, BML Munjal University, National Highway 8, 67KM Milestone, Gurugram, Haryana, 122413, India