Artificial Intelligence Review
https://doi.org/10.1007/s10462-019-09790-9

A comprehensive review of Bengali word sense disambiguation

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

Abstract
The entities of communication have an enormous impact on interaction. Textual data is an important attribute of communication. Textual analysis of this data is carried out by the linguistic researchers in various perspectives. It helps to understand the people's perception by analyzing the contextual data into its various senses. The sense of a polysemous word is varied according to its context. Hence, the process of identifying the proper meaning of a polysemous word with respect to the context is known as word sense disambiguation (WSD). For the extraction of actual meaning, WSD is an essential technique in Natural Language Processing. Over the last two decades, a lot of algorithms have been proposed to solve this linguistic ambiguity problem in various languages. In addition, a number of review papers have been published in various most spoken languages. Even so, it is elevating that there is a discontinuity in the literature when it comes to the techniques of Bengali WSD. This paper confers an extensive survey work regarding approaches of Bengali WSD. It also presents a survey work of the existing dataset of Bengali WSD.

Keywords Word sense disambiguation · Bengali WSD · WSD survey

1 Introduction

The living entity utilizes language in diverse forms for communicating their thoughts. The information is extracted from the linguistic data. Human can derive the exact meaning of a word from the interpretation of the text as well as contextual information. A WSD problem is

✉ Debapratim Das Dawn
debapratimdd@gmail.com

Soharab Hossain Shaikh
soharab.hossain@bmu.edu.in

Rajat Kumar Pal
pal.rajatk@gmail.com

[1] Department of Computer Application, B.P. Poddar Institute of Management and Technology, Kolkata, India

[2] Department of Computer Science and Engineering, BML Munjal University, Gurgaon, India

[3] Deparment of Computer Science and Engineering, Calcutta University, Kolkata, India

Published online: 05 December 2019

the most troublesome for identifying the most relevant meaning of a word in a more context-specific way. The practical importance of a WSD system is to regain out the accurate sense of a polysemous word automatically by using machines. The automatic recognition of senses of words by processing textual data is called word sense disambiguation.

The research on WSD has been coming forth over a few decades ago. In fact, WSD has great historical momentousness over a brace of years ago. In the 1940s, WSD had gained importance during automatic machine-based translation (Weaver 1949). The hardness of the WSD problem was the leading stumbling block still 1960 (Bar-Hillel 2003). By integrating the ideas of Artificial Intelligence (AI), researchers were trying to solve it in a more intelligent way in the 1970s (Wilks 1975). Entirely the same, resource unavailability was the remain limitation for gathering appropriate knowledge to disambiguate the word with proper sense. Hence, after making the availability of huge amount knowledge resources; WSD research got momentum in the module of knowledge exaction for processing the natural languages (Wilks and Carter 1990). At the beginning of the twentieth century, it got importance periodically after incorporating the ideas of statistical knowledge (Ide and Véronis 1998).

The complexity level of a WSD problem is reckoned with the AI-complete problem. WSD is implemented on a single document, or multiple documents depending on contextual toughness. The language ambiguity of Bengali literature presents in three forms, such as: (a) ambiguity at word level, (b) ambiguity at sentence level, and (c) ambiguity at meaning level. The WSD problem deals with the ambiguity at meaning level. The meaning of a word is varied according to its senses. The sense of a word, which is also dependent on the textual data, is influenced by its contextual information. The subjectivity of textual data is determined by the linguistic and grammatical rules of a particular language. The linguistic and grammatical rules of a language are described by the number of senses present of a word. If the number of sense of a word is one, the word is monosemous; otherwise, the word is polysemous.

In general, the algorithms of WSD are tested on the dataset of polysemous words rather than monosemous words. A monosemous word indicates only a single sense; however, a polysemous word indicates multiple senses. The symbolic interpretation of a monosemous word and polysemous word are mentioned in Eqs. 1, and 2 respectively. Here, w indicates the specific word and i varies in the number of senses. The formulation of recognition is described in Eq. 3.

$$Monosemous : |(Senses_i(w)| = 1 \tag{1}$$

$$Polysemous : |(Senses_i(w)| > 1 \tag{2}$$

$$Recognition : |(Senses_i(w)|; where\, i \in 1, 2, 3, 4, \ldots, n. \tag{3}$$

The approaches of WSD are varied from a number of polysemous words present in the context. Broadly, the metamorphosis of WSD algorithms is classified in target word and all word-based WSD.

(a) Target Word WSD The target word WSD approach is used to disambiguate the sense of a polysemous word from a restricted set of words. This set is called lexical sample. It is a word specific context-based WSD. It is generally followed by various supervised-based algorithms, such as: decision list, decision tree, Naïve Bayes, neural networks, and so on. It is required to predefine the polysemous words at training level, where training and testing set are designed separately (Carpuat and wu 2005).

(b) All Word WSD The all word WSD approach is disambiguating the sense of all polysemous words, present in the context. This approach is followed as an open-class problem. It is a generalized form to disambiguate sense of a polysemous word from textual data. The knowledge-based or unsupervised-based algorithms are used here, such as: context cluster-

ing, word clustering, co-occurrence graph, sectional preferences, and so on. Training set is not required here, but an external knowledge source is required to check whether a word is polysemous or not (Lee and Ng 2002).

Referable to the hardness, WSD is an interesting problem for most of the Natural Language Processing (NLP) research communities worldwide. An extensive level of various research papers have been already published in most of the notable languages in the world. Over the last four decades (Moro et al. 2014), the utmost research works regarding WSD have been reported for solving the ambiguity problem in English. In addition to, WSD is implemented in various widespread languages like: Italian (Lupu et al. 2005), French (Segond 2000), German (Escudero et al. 2000), Chinese (Yunfang 2009) and so along. The highest tier of recognition accuracy has been carried through in these languages. In the example of Asian languages, a lot of approaches have been presented regarding WSD, like Nepali (Dhungana and Shakya 2014), Arabic (Zouaghi et al. 2012; Elayeb 2019), Myanmar (Aung et al. 2011) and so on. Along with, WSD research has been done in various Indian languages such as, Assamese (Sarmah and Sarma 2016b), Kannada (Parameswarappa and Narayana 2013), Malayalam (Haroon 2010), Hindi (Mishra et al. 2009), etc.

Over the last two decades, a lot of notable review works have been done in most common languages in Europe, Asia and India. However, there is a gap in the literature when it comes to reviewing the work of Bengali WSD methodologies. Actually, the Bengali language belongs to the family of Indo-Aryan languages. It is an official language of Bangladesh and some states of India, such as West Bengal, Tripura and Assam. The modern Bengali literature has been developing since the twentieth centuries. It is generally followed by subject–object–verb (S–O–V) word order, written alignment from left to right and there is no capital letter at that place.

Over the last five years, a lot of research works have been published in WSD of Bengali literature. The Bengali language includes a huge set of homonyms and polysemous words. A generic WSD system consists of six steps, such as,

Step 1 Text preprocessing It is the beginning step of any WSD algorithm. It consists of a collection of Bengali word in contextual forms. Basically, Bengali words are of two types, monosemous and polysemous. The monosemous word has singleton meaning, whether the polysemous word has multiple significance. The main motto of this module is to distinguish a set of polysemous words on the basis of contextual information.

Step 2 Text normalization The conversion of unstructured format to the structured format of raw textual data is the objective of the text normalization module. The several types of text normalization step are described here:

Step 2.1 Modification of dissimilar fonts to similar cases.

Step 2.2 Removing the set or unnecessary symbols such as *,@,-, …,>, <,braces, and so on.

Step 2.3 Chopping of the terminal markers especially for the Bengali.

Step 2.4 Deletion of all the punctuation marks present in the text.

Step 3 Feature selection Features are selected from the textual data to find the distinctive attribute for separating key items. The set of characteristics is collected from a single document or multiple documents. The local features are chosen from a single text document, and global features are selected from multiple documents.

Step 4 Feature extraction The key characteristics are collected from the textual data by extracting useful information. This phase is very crucial for sense identification. The types of feature extraction depend upon the morphological complexity of the language.

Step 5 Sense recognition Ambiguity removal is a crucial function of sense recognition. It identifies the exact sense of a word by processing the extracting features of the setting. It eliminates the sense, which does not confer any meaning in the sentence.

Step 6 Performance evaluation This phase is validating the robustness of the algorithm. If the sense is identified properly, it is called a positive result; otherwise it is called a negative result. The average grade is calculated by collecting the result of the entire dataset.

In addition, the Bengali WSD has many linguistic applications in real life. A list of applications is mentioned here, such as,

(1) Machine Translation (*MT*) It is pre-programmed translation as defined in the standard lexicon. It is the most direct enthralling application of WSD. The exact word is preferred from the bilingual corpora by solving the ambiguity problem of polysemous words. Another objective of MT is to spread out the Bengali language all over the world by translating in a suitable form (Bhala and Abirami 2014).

As an example,

Polysemous word: জল (*Water*)

Sense: অত্যধিক পরিশ্রম (*Exhausted/Ruined*)

Sentence: এই পদকের পিছনে রয়েছে রক্ত জল করা পরিশ্রমের কাহিনী ।

Translation (without WSD): *Behind this medal is the story of blood-watering labour.*

Translation (with WSD): *There is a lot of hard work behind this medal.*

The translation without WSD does not confer any meaning. However, the translation with WSD is described the exact meaning of the sentence.

(2) Part of speech (*POS*)*-Tagging* The POS determination of Bengali word depends upon the contextual information. POS of a word is varied according to it's sensory information. Thus, POS-tagging after determination of sense always gives better results. From the previous example, জল (*Water*) is noun. However, জল (*Water*) in this sentence এই পদকের পিছনে রয়েছে রক্ত জল করা পরিশ্রমের কাহিনী। (*There is a lot of hard work behind this medal.*) is adverb, when sense is অত্যধিক পরিশ্রম (*Exhausted/Ruined*).

(3) Subjectivity detection Subjectivity detection is considered as a text classification problem for categorizing the text along with the basis of subject or object. It is really important for recognizing an individual's opinion and sentiment. The polysemous words are playing a crucial role to determine subjectivity. As for the example:

Polysemous word: জল (*Water*)

Sense 1: এক প্রকার পানীয় বিশেষ (*One type of drink*)

Sentence 1: এক অণু জল দু'টি হাইড্রোজেন পরমাণু এবং একটি অক্সিজেন পরমাণুর সমযোজী বন্ধনে গঠিত। (*One molecule of water is formed by a covalent bond between two hydrogen atoms and an oxygen atom.*)

Subjectivity 1: Chemistry

Sense 2: অত্যধিক পরিশ্রম (*Exhausted/Ruined*)

Sentence 2: এই পদকের পিছনে রয়েছে রক্ত জল করা পরিশ্রমের কাহিনী। (*There is a lot of hard work behind this medal.*)

Subjectivity 2: Sociology

(4) Emotion Tagging from Blogs Emotion analysis (such as happiness, sorrow, fear, surprise and so on) on text is a very challenging task due to the presence of polysemous words in the context. The polysemous words are playing a critical role in the determination of emotion. Senses of the polysemous word are the keywords of this diligence. As for the example:

Polysemous word: বোমা (*Bomb*)

Sense 1: বিস্ফোরক পদার্থ (*Explosive material*)

Sentence 1: প্রথমেই যে অস্ত্রের নাম সবার মাথায় আসবে সেটা হলো পারমাণবিক বোমা। (*The first weapon that comes to the mind is an atomic bomb.*)

Emotion 1: Negative
Sense 2: পাম্প (*Pump*)
Sentence 2: বোমা চালিত ১০টি গভীর নলকুপের পানিতে চাষ হচ্ছে ধানসহ বিভিন্ন ফসল। (*Water from ten deep tube-wells run by pumps is used to cultivate various crops.*)
Emotion 2: Positive

5) Opinion-Polarity Identification: In web information extraction, identification of opinion polarity for mining customer reviews is the main objective of opinion-polarity identification. It is a sentiment recognition from the text. The ambiguous words are very important in this scenario for telling, "What other people imagine"? As for the example,
Polysemous word: টিকা (*Vaccination*)
Sense 1: তিলক (*A holy mark on the forehead*)
Sentence 1: জ্যোতিষ শাস্ত্র অনুযায়ী মোক্ষ লাভের জন্য বৃদ্ধাঙ্গুষ্ঠি বা বুড়ো আঙুল দিয়ে টিকা কাটলে তা মঙ্গলজনক হয়। (*According to astrology, it is auspicious to draw the mark of sandal paste on the forehead using the thumb.*)
Opinion polarity 1: Appreciate
Sense 2: রোগের প্রতিষেধক (*Antidote*)
Sentence 2: শিশুর সুস্থতার জন্য টিকা দেয়া খুবই গুরুত্বপূর্ণ। (*Vaccination is very important for the health of a child.*)
Opinion polarity 2: Symptom

2 Related works

The WSD problem is one of the challenging tasks for NLP researchers. The complexity level of a WSD algorithm depends on the coarseness of the sensible direction. WSD is a combination of machine learning, computational linguistics and language processing. Due to the vast application areas, WSD is famous in various languages worldwide. Over the past decade, a great dearth of survey papers of sense disambiguation has been published in several languages along with English. Some set of important review papers is referred to in this section for analysing the WSD in details.

In English, Navigli (2009) has been published a review paper of WSD in 2009. An outline of WSD has been made here for formalizing all tasks. The elements of WSD have been separated into four modules, such as sense selection, knowledge source, context representation and a suitable classification method. Supervise (such as, Decision Lists and Trees, Naive Bayes, Neural Networks, Support Vector Machine, Ensemble Method, etc.), semi-supervised (such as: Bootstrapping, Monosemous Relatives), unsupervised (such as, Context Clustering, Word Clustering, Co-occurrence Graphs, etc.) and knowledge-based approaches (such as, Selectional Preferences, Structural Approaches, etc.) have been discussed in details. Nevertheless, it was unable to focus on performance tuning for finding out the percentage of accuracy in the disambiguation task.

Consequently, Zhou and Han (2005) have categorized the WSD algorithms on the basis of computational complexity and performances. Since knowledge and contextual features are depicted here for explaining the WSD techniques. The sense knowledge of a WSD system has been classified into lexical knowledge and world knowledge. Lexical knowledge includes various features like sense frequency, glosses, concept tress, subject code, POS, etc. The world knowledge describes syntactic features, domain-specific knowledge, parallel corpora and so on. However, it has felt to do standardization of all the algorithms of WSD.

In addition, Tatar (2005) has been performed a survey regarding machine learning approaches of WSD. In machine learning approach, the polysemy property has been formulated by combining the two approaches, such as sensory discrimination and sense tagging. The context is defined in two contours, such as a bag of words (BOW) and relational information. The BOW model is more suitable to a noun, compared to other POS, especially the verb. Because the verb has obtained much disambiguate information of an object rather than a subject. Moreover, the WSD approaches have classified into three categories, such as supervised, bootstrapping and unsupervised. The bootstrapping approach is a sandwich technique between supervised and unsupervised. However, this survey only concentrated on WSD in the information retrieval domain and did not discuss other wings of WSD.

Borah et al. (2014) have meditated on sense knowledge, lexical features and semantic features for building conceptual models of WSD. This survey paper has focused on knowledge-based approach rather than supervised and unsupervised approach. The knowledge-based approaches have been classified into overlap based approach and selectional preferences. The conceptual density, Lesk's algorithm, and Walker's approach have been hashed out in the section overlap-based algorithms. Lesk's algorithm is very sensitive compared to the other two approaches. Because it consists of the reading of a nominee's sense of the polysemous word and every single sense of the contextual word. Moreover, this paper has been reported that Decision List has better mean precision value and baseline accuracy value compared to other supervised-based techniques. However, it has not compared the approaches of knowledge and an unsupervised approach. It has set side by side only contemporary supervised-based techniques.

In Japanese, Kalita and Barman (2015) have classified the disambiguation techniques in three phases, such as sense inventory, training information and test data. SemEval is very popular in this language for appraising semantic scanning system. Context, collocation, association and topic vector are generally used for punching. Context vector indicates co-occurrence of words in the same text files, while collocation vector has a relatively high precision value with recall scare. Association vector is applied to remove the dependent variables. Data sparseness of context vector and association vector are different. Topic vector is generally used for finding a topic of textual data by using Probabilistic Latent Semantic Indexing (PLSI). This paper has mentioned that supervised-based algorithms are really popular in the Japanese language. Web pages, newspaper, books and white paper, are very popular resources for creating the dataset. However, this paper has covered very little works in Japanese state-of-the-art WSD methodologies.

In Chinese, Yunfang (2009) has performed a review of works of contemporary WSD algorithms. This paper has been also included linguistic resources and semantic evaluation. Automatic lexicon collection and semantic annotation have been seen for explaining linguistic resources. SemEval has been surveyed for the semantic evaluation. However, this report has been reviewed only supervising-based techniques.

In Arabic, Alian et al. (2017) have reviewed all the concurrent techniques of sense disambiguation. Being Arabic is a Semitic language, this paper has explained the hardness of Arabic WSD. In general, an Arabic WSD consists of sense inventory, context representation and disambiguation process. Due to the absence of diacritics, the number of possible senses increases in a polysemous word. Arabic WordNet has also presented for the evaluation of knowledge-based Arabic WSD. However, this paper has been felt to describe the way-out for overcoming the challenges of Arabic WSD.

In Hindi, Sharma (2015) have classified unsupervised approaches into type-based and token-based. The clustering occurrence of an objective word is considered for type-based approach while congregating context of an ambivalent word is heeded for the token-based

approach. This paper has been also tested that the modified Lesk's algorithm is applicable also in Hindi WSD. The algorithmic steps of Lesk's algorithm are overlapped checker, score checker and select sense with the highest score. However, this report has been concentrated on knowledge-based approaches, especially Lesk's algorithm.

In Assamese, Sarmah and Sarma (2016a) have classified the overview of the disambiguation problem in three phases, such as sense repository, context representation, and identification. It has discussed several lexical doubtfulness and related application areas of Assamese WSD, especially: cross-lingual information retrieval and text classification. It has categorized the approaches of Assamese WSD in dictionary-based and machine learning. The pros and cons of contemporary systems have been remarked. However, it has not mentioned any loophole overcoming procedures of the associated systems.

In Manipuri, Shallu and Gupta (2013) have suggested that the supervised-based plan of attack is more suitable for performing WSD. Due to the contrasting in syntactic and semantic structures, the WSD systems of this language has generally followed supervised-based techniques. This paper has reported that a generic WSD of this language has followed five steps, such as preprocessing, feature section and extraction, training, testing and acknowledgment. The various cases of features are: target word, previous and next words with the position, next to next word, etc. The performance of the algorithm has been quantified by comparing the actual results with predicted sense. However, this report has not focused on other knowledge-based and unsupervised approaches.

In Malayalam, Srinivas and Rani (2016) have made a sketch of the state-of-the-art WSD techniques. It has been reported that the WSD systems of this language generally followed a knowledge-based approach, consisted of knowledge acquisition and sense selection on the groundwork of the score. It has followed Lesk's and Walker's algorithms. It has also been reported that noun based computation for determining a correct sense is applicable for measuring conceptual density. If a sentence consists of multiple nouns, it is computed conceptual density by calculating the summation of the depth of each noun. However, this paper has not described the non-noun based disambiguation technique.

In Bengali, Pal and Saha (2015) have performed a short survey regarding WSD methods. It has gone over only unsupervised graph-based approaches. In the co-occurrence graph, vertex indicates the word; edge indicates co-occurrence of the words and weight indicates a number of existing context edges. Besides Bengali, this paper has performed a review of WSD techniques in Indian languages, such as Manipuri, Nepali, Kannada, etc. It has been compared algorithms with performances of diverse techniques. However, this report has not scanned the entire methodologies of Bengali WSD.

The whole of the reported linguistic researchers remarked on various WSD systems in several languages and discussed a heap of future scopes. Table 1 presents a short summary of the reported articles. Subsequently, the review work on the field of Bengali has been progressed in very little. Hence, this paper delivers an extensive survey work on Bengali WSD.

3 Survey of literature on Bengali WSD approaches

The effective research work on surveying the Bengali WSD systems is not the same standard as research work on most spoken languages. Hence, this section has reviewed a comprehensive written report regarding state-of-the-art technologies. It is an open problem at the lexical level in computational linguistics. A variety of proposals currently available in Bengali for sense disambiguation. In the general run of affairs, the approaches of Bengali WSD system are classified into two categories such as knowledge-based and machine learning-based.

Table 1 A precise overview of all the reported papers

Paper	Year	Language	Comments
Zhou and Han (2005)	2005	English	Standardization of all the approaches is required for quantifying the performance under a single umbrella
Tatar (2005)	2005	English	Going over the contemporary techniques on the basis of information retrieving scenario
Navigli (2009)	2009	English	The performance tuning of WSD approaches is not carried out for evaluating the perception value high or low
Yunfang (2009)	2009	Chinese	Giving a special importance to semantic evaluation technique and reviewing SemEval with different editions
Shallu and Gupta (2013)	2013	Manipuri	Emphasizing on supervised-based plan of attacks due to the contrasting in syntactic and semantic structures of textual data
Borah et al. (2014)	2014	English	Concentrated, only on supervised-based algorithms of WSD and discussed about precision and baseline accuracy
Kalita and Barman (2015)	2015	Japanese	Reviewing a short on state-of-the-art techniques with emphasizing on supervised-based
Pal and Saha (2015)	2015	Bengali	Hardness of processing Bengali text is needed to do a further discussion of overcoming morphological complexity within the language
Sharma (2015)	2016	Hindi	Discussing knowledge-based WSD techniques with emphasising on WordNet
Sarmah and Sarma (2016a)	2016	Assamese	Taking out the shortcomings of state-of-the-art technologies for finding out the exact sense are required
Srinivas and Rani (2016)	2016	Malayalam	Putting importance of noun-based disambiguation technique in knowledge acquisition module
Alian et al. (2017)	2017	Arabic	Overall problems of processing of Arabic textual data for disambiguation are not directed properly

The dictionary or knowledge-based WSD is performed by building up the relations with the contextual words. It acquits itself by utilizing knowledge of various structured resources, like Thesauri, Machine Readable Dictionary (MRD), Ontology, WordNet and so on. The definition of the word meaning present in the knowledge sources is playing a lively role in disambiguation. The quality of knowledge and potency of evaluating technique is the key parameters of this type of admittance. The important of knowledge-based algorithms are Lesk's algorithm, heuristic methods, selectional preferences, semantic similarity and so on. The first algorithm for MRD-based WSD is Lesk's algorithm. Simplified Lesk and Corpus Lesk are the two variations of its (Banerjee and Pedersen 2002). The heuristic approach includes occurrences, discourse and collocation of senses (Moro et al. 2014). Selectional preferences (McCarthy 1997) estimate the coexistence of predicates and theoretical classification in morphology. Semantic similarity (Sinha and Mihalcea 2007) computes the semantic distances between semantically related words.

Table 2 An overall analysis of Bengali WSD approaches

Approach	Comments on applicability in the context of Bengali WSD
Knowledge-based	It is truly useful for all word WSD in Bengali dataset. However, the explicit knowledge resources are not available much more in the Bengali language. To overcome the overlap sparsity problem is really difficult for this morphological complex language.
Supervised	This access is better compared to the other two approaches. It is applicable for target word WSD. Sense-annotated text or raw corpora are used here. However, the main hurdle of this approach is lack of resource. Due to the scarceness of database, the research on Bengali WSD is not up to the mark compared to other most spoken languages.
Unsupervised	Bengali literature consists of a vast collection of inflected forms by adding prefix and suffix with the base word. There is no capital letter in Bengali text. Thus, it has always a low precision value compared to the other two approaches.

Over the last twenty years, machine learning-based approaches are very trendy in word sense disambiguation. The supervised, unsupervised and hybrid approach are the three common streamline of machine learning-based schema. The supervised and unsupervised both models are tunefully organized for Bengali WSD. The leading algorithms of supervised-based approaches are decision list, decision tree, Naive Bayes, Support Vector Machine (SVM) and so on (Màrquez et al. 2007). The unsupervised approaches include context clustering, word clustering (Popescu and Hristea 2011), co-occurrence graph, etc. (Pedersen 2007). However, there is a shortage of notable work done in hybrid approaches. Due to the semi-supervision in a high level morphological language, it is a challenging task to do research work on in this domain based on hybrid approach. The comments on applicability in the context of Bengali WSD has been mentioned in Table 2.

In this survey, we have also reviewed some of the articles that have used combination of the neural network model with English WSD. It has been included in this section to convey the sense that such works can also be adopted for Bengali WSD using similar combination of the neural network model.

3.1 Literature survey on knowledge-based Bengali WSD approaches

The knowledge-based Bengali WSD comprises of two algorithms, such as:

3.1.1 Haque's algorithm

Haque and Haque (2016) published research work on the dictionary-based approach of Bengali WSD by eliminating lexical semantic ambiguity. The major steps of this algorithm are parsing and detection. It has stuck with all word WSD with the helping of Bengali MRD knowledge source. The algorithmic procedures are mentioned below.

Step 1 Text Tokenization The beginning step of text processing has done by the tokenizer. It has called for input on a text document and generated tokens of respective words.

Step 2 Parse Tree Generation Parser generator has built up a parse tree after taking tokens as input by following the rules of context-sensitive grammar.

Step 3 Ambiguity Tracking Down Ambiguity identification has been performed by taking the parse tree as input for individual sentence. The detection technique has followed the following steps:

Step 3.1 Ambiguity detector has summoned up comprehensive knowledge from the Bengali dictionary and parsed tree.

Step 3.2 The ambiguous words have been detected if multiple definitions of a word exist in the dictionary.

Step 3.3 Matched up against the neighbour words to the dictionary definition for finding a proper sense of ambiguous words.

Step 4 Sense Recognition Sense recognizer has been answered with the help dictionary definition and corresponding word.

In the "Ambiguity Tracking Down" module, the algorithm has been collected lexical, syntactic and semantic information from the textual data.

3.1.2 Pal's algorithm 1

Pal et al. (2017) have extended traditional Lesk's algorithm into context expansion through synset analysis.

Step 1 Preprocessing Text normalization and deleting of non-functional words have been performed to regulate the text in the preprocessing module.

Step 2 Text Lemmatization Extraction of root word from the inflected word has been done in growing the lexical coverage of the data by using the Bengali lemmatizer tool. Manually modification has been performed at this level for identifying exact root word.

Step 3 Processing with Lesk's Algorithm The Lesk's algorithm is first MRD-based WSD technique. The steps of processing this algorithm are mentioned beneath:

Step 3.1 Selecting a short phrase from entire text carrying an ambiguous word.

Step 3.2 Resemblance has done between glosses of other words, in a specific phrase in a dictionary definition.

Step 3.3 Counting most frequent words in phrases for assigning an appropriate text in context.

Step 4 Augmentation of Baseline Strategy Due to the scarcity of information in *Step 3*, this method has been adopted in the following steps:

Step 4.1 Collocating words has been built up by the non-fixing window size.

Step 4.2 WordNet has identified synonymous words of collocating words.

Step 4.3 Overlap is computed between each sense declaration definition and string.

Step 5 Sense Recognition The maximum overlaps of ambiguous words have been boasted as the key role for finding appropriate sense.

The extension of the baseline method is the key point of this algorithm for producing results in high accuracy compare to simple Lesk approach. The Bengali lemmatizer tool is not performing well in high morphological complex words. Moreover, it is felt to handle of large number lexical overlap for a huge dataset.

3.2 Literature survey on supervise-based Bengali WSD approaches

3.2.1 Pandit's algorithm

Pandit and Naskar (2015) had performed Bengali WSD on supervised k-Nearest Neighbour (k-NN) technique. k-NN puts into sets the test dataset with the help of the training dataset. The cosine similarity measurement is very useful for the discriminative relations between testing set to the training set. The steps of this algorithm are mentioned the following:

Step 1 Removal of Stopping Words It has performed preprocessing step by just removing the stop words. The stop word set in English and Bengali is almost likely same.

Step 2 Part-of-Speech (POS) Labeling Tagging POS of each word within a sentence by using Stanford Log-linear Part-Of-Speech Tagger has been executed.

Step 3 Stemming Root Words Hold backing the root word from the inflected words has been performed. The Bengali language has a huge collection of inflected words. The processing of inflected word in a WSD algorithm is a very challenging task.

Step 4 Calculating Distance between New Instance with Stored Set It has computed distance by using overlapping metric between testing set to the training set. Context words containing the surround words of target word have been considered as a context vector for calculating distance metric. Hamming Distance, Manhattan Distance, Euclidean distance are also applicable here for measuring distance, apart from the overlap metric.

Step 5 Allocation of Weights Assigning weights with the ground of the closeness property of the overlap metric by using the k-NN method has been executed greatly. The workable formula is mentioned in Eq. 4, where d_{max} and d_{min} presents maximum and minimum distance for ith set with assigning weight w_i.

$$w_i = \frac{d_{max} - d_i}{d_{max} - d_{min}} \tag{4}$$

Step 6 Preference of k The value of k is very essential for calculating k-NN based method in WSD. The minimum measure of considerable quantity always gives better results in this scenario.

Step 7 Majority Voting Attributing the test dataset to the closet training set has been performed by using the majority voting scheme. The performance of this strategy depends upon the value of the k. The lower value of k indicates a tie between the closet dataset. The majority voting algorithm is useful for locating the larger part of a sequence of features using bounded space with a linear time period.

In contrary, this overture is very prolonged for large dataset. The k-NN is not suitable for quick learning. Finding the suitable value of k is very time consuming for a huge dataset.

3.2.2 Pal's algorithm 2

The probabilistic model-based classifier has been first used by Pal et al. (2015a) for Bengali WSD in 2015. Naive Bayes classifier has been employed for classification of the Bengali text in an automatic fashion. The computational steps are:

Step 1 Annotation of Textual Data Target word-based approach has been observed here. The sentence containing the target word has been keyed out from the whole text. The preprocessing tasks have been performed here by removing unwanted symbols, spaces, terminal markers and hence on.

Step 2 Erasing of Stopping Words Identification of the stopping words has been performed by utilizing a standard Bengali dictionary. Here, the conjunction words are treated as stop words. Because, repetition of conjunction with text, reduces the data containing the text.

Step 3 Building Learning Set The Naive Bayes model has been used for training the documents of a polysemous word with multiple senses. The working rule for finding the conditional probability of keyword occurrences w_k of giving class c_k is mentioned on Eq. 5, where n_k denotes summation of frequency count in each class.

$$P(w_k|c_k) = \frac{Frequncy\ Count\ of\ Each\ Word + 1}{n_k + |v|} \tag{5}$$

Step 4 Classification of Test Documents The unknown testing data have been classified using Eq. 6; where $P(c_k|W)$ is the conditional probability of a specific class and j varies in a number of sets. The maximum value of probability indicates the resulting consequence.

$$P(c_k|W) = P(c_k) \times \sum_{j=1}^{|v|} P(w_j|c_k) \tag{6}$$

The functioning of this algorithm is good. However, this algorithm only tested with noun dataset. The noun words only considered as the target word.

3.2.3 Pal's algorithm 3

In addition, Pal et al. (2015b) have used lemmatization-based system for magnification of precision value. The algorithmic steps for computing actual output are mentioned the following:

Step 1 Sentence Re-modification The sentence comprises of polysemous words have been split up from the whole text for selecting the appropriate context in ambiguity scenario.

Step 2 Text Lemmatization The goal of this algorithm is to get better accuracy by using lemmatization approach. Bengali literature contains a huge amount of infected objects. Thus, deriving the base word from the inflected words has been performed for doing text normalization before processing the data.

Step 3 Sentence Annotation In the preprocessing stage, sentence annotation has been formed to indicate the target word. The target words are the set of the predefined polysemous word.

Step 4 Construction of Naive Bayes Model The Naive Bayes statistical model has been used for disambiguating the senses of an equivocal word. The Bayes formula has been employed for computing a conditional probability of selected feature set in a specific class. The method of working has been using the formula 5 for construction of this module. The Laplace estimation model has been incorporated for avoiding the zero frequency problem in Bengali text.

Step 5 Disambiguate the Sense The operational procedure has been used working formula 6 for classification of proper testing data with training data. The highest probability value has been decided on the resulting outcome.

The overall performance of the process of lemmatization is quite acceptable compared to the previous algorithm. However, it has covered a very tiny database for measuring the performance of the algorithm. The database contains the only polysemous word in noun and adjective category.

3.3 Literature survey on unsupervised Bengali WSD approaches

In unsupervised strategy, the study on Bengali WSD has not great much importance. Due to the morphological complexity of the Bengali language, the unsupervised-based algorithm is not suitable. However, Das et al. have attempted to publish their work on unsupervised approach.

3.3.1 Das's algorithm

In unsupervised approach, Das and Sarkar (2013) have followed graph-based technique. Due to the prominent result, the graph-based approach has been received a lot of momentum compared to context clustering and word clustering. The operational procedures are mentioned below:

Step 1 Collecting Contextual Data The target word-based approach has been followed. Extracting of proper context containing the target word has been performed from the whole text file.

Step 2 Text Processing In the prepossessing step, target words have been removed from the contextual data for processing graph-based design. The noun words with greater than the decided outset value have been identified for further computation.

Step 3 Graph Building The co-occurrence graph has been sketched for eliciting sense-specific information corresponding to a target word. In the graph, each vertex has been designated word in the corpus set; an edge has been allotted for occurring words in the sentence, and weight of the edge has been assigned by counting the number of a context of two co-occurring words.

Step 4 Edge Deletion The removing of edges in the co-occurring graph has been conducted for knowing key edges in the graph. According to the word frequency count, the edge higher than the threshold value has been eligible for deletion. The threshold rate depends upon contextual information.

Step 5 Community Detection Operation A community suitability function has been utilized for computing sub-graph from the co-occurrence graph for a specified community. The working formula of the community fitness function is mentioned in Eq. 7, where k_{in}^{i} and k_{out}^{i} indicates internal and external degree of sub-graph i. Here, β is a tuning parameter, depending on the textual data.

$$F_i = \frac{k_{in}^{i}}{(k_{in}^{i} + k_{out}^{i})^{\beta}} \tag{7}$$

Still, the overall performance of this algorithm is not pretty convincing. The exactness value of some polysemous is very faint.

3.3.2 Pal's algorithm 4

In addition, Pal and Saha (2017) have proposed an unsupervised algorithm-based on similarity measure. In this algorithm, clustering has been performed using the weka-3-6-13 tool with the similarity measure of grouping sentences. It has used sentence level clustering with partitioning methods. It has done grouping by accumulating the sentences of likelihood features. Each group has labelled into a single sense. The algorithmic steps are mentioned below:

Step 1 Text Normalization It is the process of transforming the un-normalized text into normalized text through a set of normalization tasks. This paper has followed manual procedures to solve the un-normalization problem. It has removed a new line, hyphen, extra spaces, unwanted symbol, slash, tilde, and so on. A set of stop words, especially for Bengali, have been removed from the text. The font size of the text has been taken into consideration and maintained uniformity.

Step 2 Text Lemmatization The text lemmatization task is very important for any morphologically complex language. A morphologically complex language contains a huge amount

of inflected words. In this paper, text lemmatization has performed for all the inflected words and obtained root words.

Step 3 Feature Selection The term frequency (TF) of each word has been considered for the feature selection. The keywords have been selected from the list of most frequent words. The least frequent words have been discarded.

Step 4 Sentence Clustering In this paper, sentence clustering has been performed in two phases, (a) type-based, and (b) token-based. In the type-based method, a feature vector (i.e. TF score) of neighbour words of target word has been considered for the sentence clustering, where a number of clusters are two. In a token-based method, synsets of co-occurring words have been considered for the sentence clustering. Each cluster presents a particular sense of a polysemous word.

However, this algorithm is not suitable for the large database, where a number of senses of a polysemous word are greater than five. The context expansion using the synonymous word is not worked well.

3.4 Combination of the neural network model with WSD

Moreover, the multilingual WSD algorithms are applicable for any resource scaring language, like Bengali. The multilingual WSD algorithms are generally followed neural network-based approach. Apart from the above-reported methodologies of WSD, the combination of the neural network model with WSD has been also useful to solve the ambiguity problem (Cottrell and Small 1983). In English and some other most spoken languages, a lot of algorithms have been proposed to resolve the ambiguity problem on the basis of neural network. Generally, the neural WSD algorithms have been followed the supervised approach of WSD. Kågebäck and Salomonsson (2016), have proposed bidirectional long short term memory (BLSTM) with word embedding to achieve language independence feature in WSD. The architecture of this model is constructed with three-layer, such as softmax layer, hidden layer, and BLSTM. The BLSTM is an extended version of long short term memory (LSTM), where the state of each timestamp is going forward and backward direction only. The system has been provided with the position of target word to this model. This paper also introduced the concept drop word to increase the independence of individual words in the training corpora. It has achieved 73% accuracy in SensEval 3 (SE3) and 66.9% accuracy in SensEval 2 (SE2). The key role of this model is to consider the context window of a polysemous word to capture relative potion of surrounding words within the context window. However, this model is not useful for all word WSD. In addition, Raganato et al. (2017) have enhanced the previous BLSTM model into the encoder–decoder model and removed the vanishing gradient problem of recurrent neural network (RNN). It has used embedding layer, BLSTM layer, attention layer, and a softmax layer. This model has been evaluated all word WSD by considering POS-tagging and coarse-grained semantic labels. It has achieved 69.1% accuracy in SE3 and 72% accuracy in SE2. However, this model was tested on English dataset only, and it is not guaranteed language independent feature.

In order to address the cons of both supervised and knowledge-based approach of WSD (according to Table 12), Luo et al. (2018a) have proposed unified framework (i.e. GAS: a gloss augmented WSD neural network) by integrating the glosses and context of target word in order to facilitate both lexical knowledge and labeled data. The architecture of GAS consists of four modules, such as context [i.e. encodes sequence of surrounding words with BLSTM (Kågebäck and Salomonsson 2016)], gloss (i.e. encodes all the glosses of target word), memory (i.e. employs semantic relationship between the context and gloss embedding), and

scoring module (i.e. generates probability distribution of all possible senses). It has achieved 70.5% accuracy in SE3 and 72.2% accuracy in SE2. However, this approach is not considered the structural information for neural WSD. To overcome this limitation, Luo et al. (2018b) have applied the co-attention mechanism to capture both word and sentence level structural information. This model has consisted of three-layer, such as input embedding (i.e. encodes context and glosses), co-attention (i.e. generates context vector and gloss vector), and output layer (i.e computes the score of each sense of a polysemous word). The BLSTM has been used in case of sentence embedding in input embedding layer. It has achieved 70.3% accuracy in SE3 and 72.8% accuracy in SE2. This model is also applicable in all word WSD for Bengali.

4 Survey of literature on Bengali WSD database

On the presence of knowledge acquisition bottleneck, the evaluation of any methodology is a very hard task for the NLP researchers. Knowledge accession bottleneck is the most prominent obstacle to the WSD research of Asian languages, especially Bengali. It has been mentioned earlier that the a trivial amount of efficient dataset has been published by the Bengali WSD community over half of one decade. In general, knowledge resources are tabulated in structured resources, WordNet, semcore, and disorganized resources. The structured sources are systematized into Thesauri, Machine-Readable-Dictionary (MRD), and ontology. The unstructured resources are systematized into corpora and other miscellaneous resources (i.e. Frequency list, field labels, stops list, etc.). Raw corpora and semi-annotated corpora are the two sets of corpora.

The notable works of the Bengali WSD system have acted on the basis of MRD, WordNet and raw corpora-based dataset. Several state-of-the-art technologies have been tested on this dataset. The detail review work on Bengali WSD has been talked about here.

4.1 Literature survey on MRD-based Bengali dataset

MRD is a computer loaded dictionary available for the English, Dutch, Spanish, Hindi and so on. It is saved inside the hard disk or accessible from the network. The pros of the MRD over other dataset is that it has the adaptability to take any kind of change for accessing and storing information. It is handled as a database (Baldwin et al. 2008).

4.1.1 Dataset 1

In the dictionary-based approach, MRD has received a lot of momentum over the half of one decade in Bengali WSD. MRD has comprised lexicons of Bengali language and corresponding meaning. A lexicon is a word-stock, consists of a catalogue of lexemes. In Haque and Haque (2016) performed WSD task by taking the input of MRD. It has included all the possible senses of ambiguous words with corresponding meaning. The attributes in this MRD have been included in various parameters, such as word, type, feature, keyword and meaning. POS of each polysemous word has mentioned in the type subsection.

The parser has directly tied with the dictionary for taking input and further analysis. The parse tree has verified with the help dictionary definition. The ambiguity detector has clarified that a word is polysemous or not. A word is exposed as a polysemous word if it provides multiple meaning. However, this report has considered only a noun, adjective and verb as

Table 3 Gloss of Bengali MRD-based *Dataset 1* (Haque and Haque 2016)

Polysemous word	POS tagging	Selected feature	Key word	Sense
মাথা (*Head*)	Noun	Organ	গাঁ, গ্রাম (*Village*)	গ্রামের প্রধান বা মোড়ল (*Foreman*)
			ব্যথা (*Pain*)	শারীরিক অসুস্থতা (*Physical illness*)
			কাটা (*Cut*)	আত্মসম্মান হানি (*Loss of self-esteem*)
পাকা (*Ripe*)	Verb	Quantitative	আম (*Mango*)	পরিপক্ক হওয়া (*Be mature*)
			বাড়ি (*Home*)	ইষ্টক নির্মিত (*Made of bricks*)
			চাকরি (*Job*)	স্থায়ী নিযুক্তিকরণ (*Permanent job*)
ছাড়া (*Leave*)	Verb	Transitive	হাত (*Hand*)	আশানুরূপ না হওয়া (*Belie*)
			চাকরি (*Job*)	অন্তিম অবলম্বন (*The last resort*)
			দেওয়া (*Give*)	স্বাধীনতা প্রাপ্ত হওয়া (*Getting independence*)
মুখ (*Face*)	Noun	Organ	রাখল (*Keep*)	গৌরবান্বিত করা (*Glorify*)
			চুনকালি (*Lime*)	লজ্জাস্কর ঘটনা (*Shameful event*)
			ভার (*Load*)	বিষণ্ণ বদন (*Sad face*)

Table 4 A snapshot of *Dataset 5*

Word	POS	Sense 1	Sense 2	Sense 3	Sense 4
মাথা (*Head*)	Noun	মস্তক (*Head*)	চিন্তা (*Think about*)	প্রান্ত (*Edge*)	–
তোলা (*To pluck*)	Verb	উত্তোলন করা (*To lift*)	সৃষ্টি করা (*To create*)	সংগ্রহ করা (*To collect*)	অর্পণ করা (*To submit*)
হাত (*Hand*)	Noun	অবদান (*Contribution*)	হাতবদল (*Shifting*)	হস্ত (*Hand*)	হাতপাতা (*To beg*)
দিন (*Day*)	Noun	প্রতিদিন (*Everyday*)	দিবস (*Daytime*)	দিনকাটা (*Going on*)	দেওয়া (*To give*)

a polysemous word. A gloss of this dataset is mentioned in Table 3. The nearest English translation of each Bengali word is also mentioned.

4.2 Literature survey on WordNet-based Bengali dataset

Over the last two decades, all word WSD has been received with great importance in WordNet-based WSD database in many most spoken languages like English, Dutch, Italian, Hindi, French, German including Bengali. WordNet is a lexical database since 1985. Several types of lexical relations are present in WordNet, such that, synonymy (i.e. likeness of the meaning), antonymy (i.e. opposite meaning), hyponymy (i.e. semantic relation in meaning) and so on.

4.2.1 Dataset 5

Pal et al. (2015b) have been proposed raw corpora-based dataset. The dataset has been comprised of four polysemous words, seventeen senses, and one hundred sentences. These four words are really common in Bengali text. From these four words, three words are the noun, and one word is a verb. The total number of senses of noun and verb polysemous word is eleven and six respectively. A snapshot of this dataset is mentioned in Table 4. The number of senses of each polysemous word has been evidenced in Fig. 1.

4.2.2 Dataset 2

Pal et al. (2015a) had published Bengali WSD-based research work by utilizing the corpora of Indian Languages Corpora Initiative (ILCI), developed in Technology Development

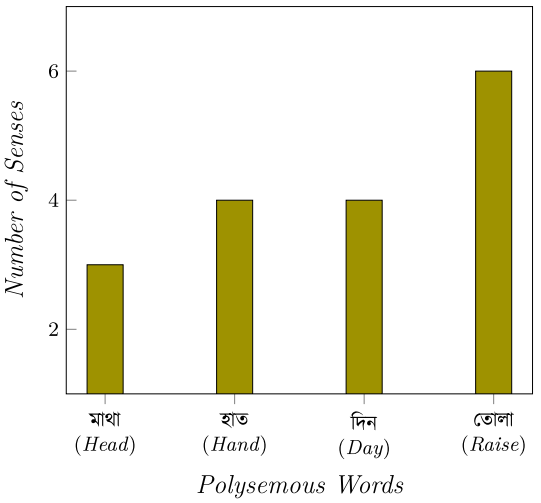


Fig. 1 Statistics of Bengali raw corpora-based *Dataset 5*

Table 5 A snapshot of *Dataset 2*

Class	Concept	Example
1	মাথার উপরের এবং সামনের অংশ (*Skull*)	রামের মাথায় আভা বিচ্ছুরিত হচ্ছে। (*Rays are being dispersed on Ram's head.*)
2	গলার সামনের অংশ (*Face*)	মাথায় আঘাত লাগার ফলে মানুষের প্রাণও যেতে পারে। (*A human being can lose life due to injury the head.*)
3	নৌকা বা জলযানের অগ্রভাগ (*The first part of a boat*)	তিনি বিশ্রাম নেওয়ার জন্য নৌকার মাথায় গিয়ে বসলেন। (*He went to the first part of a boat for taking rest.*)
4	কোন উঁচু ভবন (*Tall building*)	যে বাড়ির মাথায় চিল বানানো রয়েছে, আমি সেখানেই থাকি। (*I live on top of the building.*)

for Indian Languages (TDIL) project by Govt. of India. Apart from Bengali, this dataset is available in multiple Indian languages, such that, Hindi, English, Assamese, Bodo, Gujarati, Kannada, Kashmiri, Konkani, Malayalam, Manipuri, Marathi, Nepali, Sanskrit, Tamil, Telugu and Punjabi. It has identified each word by corresponding synset id. A word is treated as the polysemous word if the number of synsets is more than one. Gloss in English and Hindi are also available for each word in these Indian languages. Apart from these, noun relation and verb relation with POS-tagging are also available. A snapshot of this dataset is mentioned in Table 5. This dataset has been included in documents of the various category, like Accountancy, Botany, Criticism, Drawing, Economics, Folk, Game and so on. Some important categories and the corresponding number of sentences have been drawn on Fig. 2. The POS of Bengali WordNet has been analyzed systematically on Fig. 3.

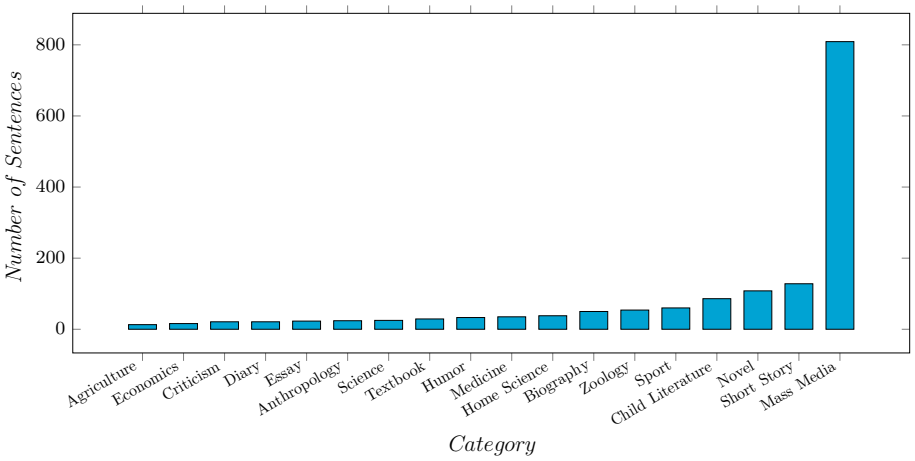
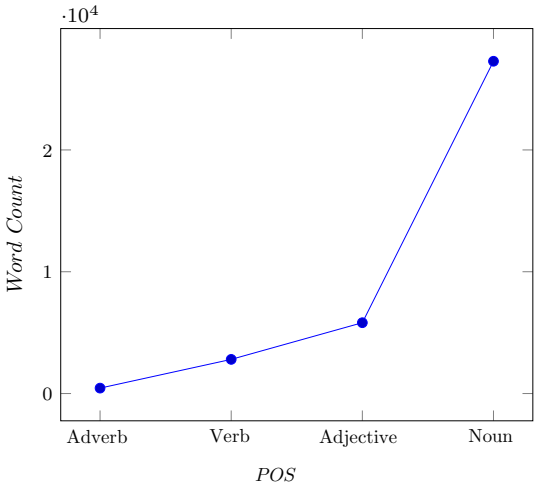


Fig. 2 Category-wise classification of WordNet-based Bengali *Dataset 2*

Fig. 3 POS categorization of WordNet-based Bengali *Dataset 2*



4.2.3 Dataset 3

Pal et al. (2017) have used WordNet dataset, which is a lexical knowledge-based machine-readable semantic dictionary, developed at the Indian Statistical Institute, Kolkata, under the Indradhanush Project of the DeitY, Government of India. The dataset is obtained from the Bengali corpus, developed under the Technology Development for the Indian Languages (TDIL) project of the Government of India. This dataset is comprised of four types of languages, such that, Indo-Aryan, Austro-Asiatic, Dravidian, and Tibeto-Burman. It also consists of a corpora management tool. The corpora management tool has been included both multilingual and parallel corpora tool. In addition, this dataset has been also provided shallow parser tool in Indian languages for morphologically analyzing and POS-tagging. A snapshot of this dataset is mentioned in Table 6.

Table 6 A snapshot of *Dataset 3*

Word	POS	Example
ঘন্টা (*Hours*)	Noun	সময় সূচিত করার জন্য যে ঘন্টা বাজানো হয় (*Bell*)
	Noun	ষাট মিনিটের সময় (*Time*)
নাম (*Name*)	Verb	এমন কিছু করা যাতে খ্যাতি বাড়ে (*Designation*)
	Noun	ঈশ্বরের নামে জপ (*Praying*)
সময় (*Time*)	Noun	ইতিহাসে প্রায় নির্দিষ্ট সময়সীমা (*Period*)
	Verb	সময় ঠিক করা (*Opportunity*)
পা (*Leg*)	Noun	সেই পরিমাণ দূরত্ব যা এক বারে যাওয়া যায় (*One step distance*)
	Verb	যাওয়ার জন্য পা উঠিয়ে অগ্রসর হওয়া (*Move forward*)

Table 7 A snapshot of *Dataset 4*

Word	Sense 1	Sense 2	Sense 3
অর্থ (*Money*)	Money	Meaning	–
চাল (*Rice*)	Rice	Maneuver	Roof
জাল (*Net*)	Forge	Net	Trap
লক্ষ্য (*Aim*)	Aim	Purpose	Observation

4.3 Literature Survey on raw corpora-based Bengali dataset

Over the span of years, target word WSD in Bengali text has been received a lot of momentum with the help of raw corpora-based dataset. Due to the sparsity, all word WSD task is not suited for this type of dataset. The techniques on both supervised and unsupervised have been examined on this dataset. In Bengali, the availability of the total number of raw corpora-based dataset is two. Each dataset has been explained briefly in the following two subsections.

4.3.1 Dataset 4

Das and Sarkar (2013) have proposed a raw corpora-based dataset. The dataset has been comprised of ten words. This dataset has been tried out on an unsupervised approach. The potentiality of this algorithm is really less. Out of ten words, two words have four numbers of the senses. A snapshot of this dataset is mentioned in Table 7. The details statistics of this dataset with each polysemous word have been drawn in Fig. 4.

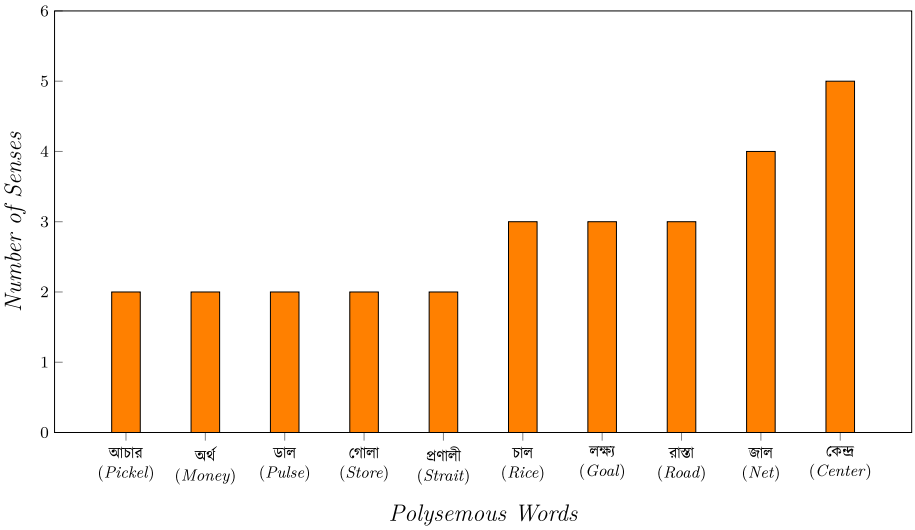


Fig. 4 Statistics of Bengali Raw Corpora-based *Dataset 4*

4.4 Literature survey on sense-annotated Bengali dataset

In most events, target word WSD has been performed with the help of supervise-based learning strategy. Polysemous word is chosen as a target word. Target word WSD is very useful in the sense-annotated dataset. Generally, the feature vector of this type of dataset has been comprised of collocational (i.e. the position of a word from the target word) and BOW feature (i.e. frequency counts within the fixed context window).

4.4.1 Dataset 6

Pandit and Naskar (2015) have published a sense-annotated dataset for evaluating a new WSD technique. This dataset has been followed by the supervised approach. Training and testing set have been designed individually for each sense of a particular polysemous word. This dataset consists of twenty-five polysemous words. As for the example, the word ফল has two meanings, such as পরিণাম (*Result*) and ফলমূল (*Fruit*). The statistical details of this dataset has been stated in Fig. 5 The POS categorization of all the polysemous word has been mentioned in Fig. 6.

5 Discussions

In the review part, more than half a dozen algorithms have been considered for the detailed interpretation. These algorithms have been classified into three modules, such as knowledge-based, supervised and unsupervised. A short analysis of each paper and corresponding approach has been tabulated on the Table 8. The critical comment section of each algorithm has been mentioned below.

In the knowledge-based algorithms, *Haque's Algorithm* has not been considered syntactic characteristics of the sentence. Since it has followed all word WSD, the collecting of syntactic

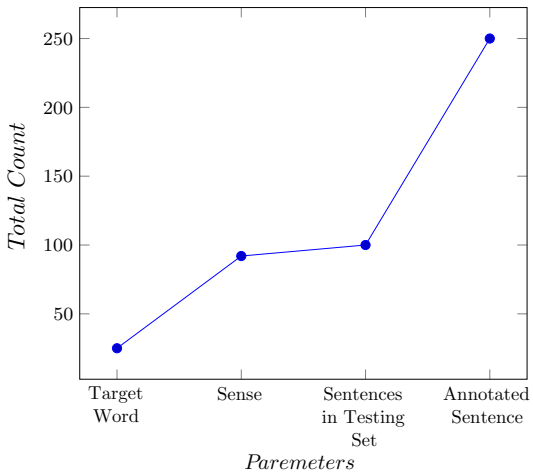


Fig. 5 Statistics of Bengali sense-annotated *Dataset 6*

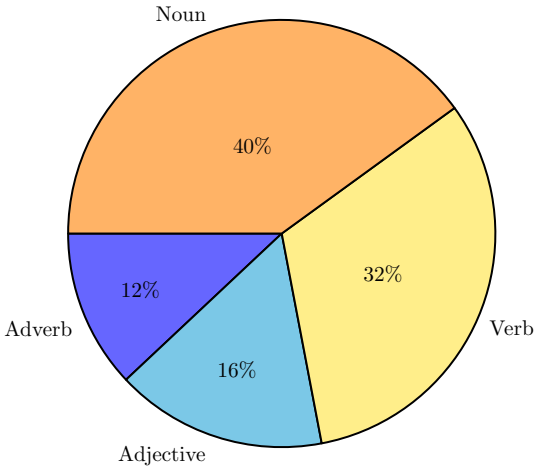


Fig. 6 POS distribution of the target words of *Dataset 6*

feature is very necessary to disambiguate the polysemous words. Moreover, *Pal's Algorithm 1* has been suffered in the lexical overlap of high degree polysemous words.

In the unsupervised-based algorithm, *Das's Algorithm* has been performed target word WSD. However, this algorithm has been followed by specialization rule of some specific dataset. Thus, a generalization of rules is needed for testing large amount dataset. In addition, *Pal's Algorithm 4* has been performed target word WSD using *Dataset 3*, mentioned in Table 6. It has been tagged POS of each word in an automated way. It has followed sentence clustering using similarity measure. However, it has not enabled to stem correctly all the inflected words in cent percentage. The algorithmic performance depends on the words lemmatization step only.

From the literature study, it has been shown that the supervised-based model is very fitting for Bengali WSD. Despite, *Pal's Algorithm 2* has been considered all the polysemous noun words. In Bengali literature, it is easy to disambiguate noun-based polysemous word, compare to a verb, adjective or adverb. In addition, *Pal's Algorithm 3* has been unable to do lemmatization of verb words in the preprocessing phase.

Table 8 Glosses of approaches with corresponding papers

Paper	WSD approach
Das and Sarkar (2013)	Unsupervised
Pal and Saha (2017)	Unsupervised
Pal et al. (2015a)	Supervised
Pandit and Naskar (2015)	Supervised
Pal et al. (2015b)	Supervised
Haque and Haque (2016)	Knowledge-based
Pal et al. (2017)	Knowledge-based

Table 9 Systematic analysis of the dataset and corresponding WSD variant

Dataset	References	WSD variant	Approach
Dataset 1	4.1.1	All word WSD	Knowledge-based
Dataset 2	4.2.2	All word WSD	Supervised
Dataset 3	4.2.3	All word WSD	Knowledge-based
Dataset 4	4.3.1	Target word WSD	Unsupervised
Dataset 5	4.2.1	Target word WSD	Supervised
Dataset 6	4.4.1	Target word WSD	Supervised

Table 10 Year-wise analysis of WSD variant with corresponding knowledge resources

Knowledge source	Year	WSD variant
Raw Corpora	2013	Target word WSD
WordNet	2015	All word WSD
Sense-annotated	2015	Target word WSD
Raw Corpora	2015	Target word WSD
MRD	2016	All word WSD
WordNet and Corpora	2017	All word WSD

All the dataset for Bengali WSD has been classified into four subsections. In general, WSD algorithms have been followed by two variants, such as all word WSD and target word WSD. The systematic analysis of WSD variant has been expressed in Table 9. This Table is shown the dataset selection in a generalized way, based on the three approaches: knowledge-based, supervised-based and unsupervised-based.

The relation between knowledge source and WSD variant has been playing a key role in designing a new database. The year wise detailed analysis of each knowledge resource and corresponding WSD variant has been mentioned in Table 10. It has shown that target word WSD has been experimented on unstructured resource while all word WSD has been tested on the structured resource.

In low resource language, knowledge resource has been received great importance for the performance analysis of the proposed algorithm. In Bengali WSD, structured and unstructured resources are available. All the mentioned algorithms of Bengali WSD have followed six different types of knowledge resources, mentioned in Table 11. The pros and cons of all the reported algorithms have mentioned in Table 12

Table 11 Interpretation of WSD algorithms with corresponding knowledge resource

Algorithm	References	Knowledge source
Pandit's Algorithm	3.2.1	Sense-annotated
Haque's Algorithm	3.1.1	MRD
Das's Algorithm	3.3.1	Raw corpora
Pal's Algorithm 3	3.2.3	Raw corpora
Pal's Algorithm 2	3.2.2	WordNet
Pal's Algorithm 4	3.3.2	WordNet
Pal's Algorithm 1	3.1.2	WordNet and Corpora

Table 12 The pros and cons of all the reported algorithms

Algorithm	Pros	Cons
Das's Algorithm	The edge-density of a graph has been used here for identifying the clusters within the graph.	It has not followed any general rule for words clustering. Thus, the overall result is varied in large scale for other dataset.
Pal's Algorithm 4	This work has followed type and token-based distributional approach for Bengali sentence clustering. The TF score of co-occurring words and synsets of collocating words have been taken into consideration.	The hierarchical method in clustering technique has not considered here. It is performed only sentence level clustering, not paragraph level. Instead of automatic text lemmatization, it has performed manual text lemmatization.
Pal's Algorithm 2	It has used Naive Bayes classification model for preparing the learning set. It has also capable to classify unknown text data.	The noun word has been only used as targeted word. So, result would be differed of other POS, specially verb. If verb is targeted word, then stemming is needed for finding the root word.
Pandit's Algorithm	It has performed majority voting scheme, which is depend upon the number of clusters and allocation of weights.	This algorithm is very slow for large database. The k-NN algorithm is called lazy learner. It takes huge time to learn the training set when the database is too large.
Pal's Algorithm 3	It has used text lemmatization tool for gaining better accuracy. It has included Laplace's estimation model for avoiding zero frequency problem.	It has worked only a very small dataset. So, there is need to use some words for various POS.
Haque's Algorithm	The major steps of this algorithm are parsing and detection. It has collected lexical and syntactic features from the textual data.	It has not used semantic feature of the sentence. It has focused only three POS, such as: noun, adjective and verb.
Pal's Algorithm 1	It has obtained better accuracy compare to the simple Lesk's approach. The algorithmic steps of this approach is better to disambiguate compare to the simple Lesk's approach.	The Bengali lemmatizer tool is not stemming well for morphological complex words to get the root word. Moreover, it is felt to handle a large number lexical overlap for a huge dataset.

In addition, some complicated issues may be present in the text. Such as:

(a) Various grammatical structures present in the sentence.
(b) Incorrect syntactic structure presents in the text.
(c) Very less amount of semantic information is present in the text.
(d) Word stemming is not possible for all the inflected words.
(e) The Bengali WordNet is now developing phase. It is not contained all the word, present in the Bengali literature.
(f) Capital and the small letter is not present in Bengali literature.
(g) Bengali is resource scaring language.
(h) POS-tagging is also difficult for this language.

6 Results analysis

All the reported papers have been examined on the various dataset. In the review section, seven WSD algorithms and corresponding six dataset have been talked about briefly. In this section, the result analysis of each algorithm has been referred in detailed.

6.1 Results analysis of unsupervised approaches

In an unsupervised approach, *Das's Algorithm* has been tested on *Dataset 4*. It has considered three types of senses, such as $|Sense| = 2, 3,$ *or* 4. The average accuracy of each sense type of this algorithm has been drawn in Fig. 7. The maximum accuracy has been obtained in the type $|Sense| = 2$. The accuracy of ten polysemous words in *Dataset 4* has been expressed in Fig. 8.

In addition, *Pal's Algorithm 4* has been tested on *Dataset 3*. This algorithm has experimented on one thousand three hundred seventy-one sentences. It has obtained 54% accuracy in the type-based method and 63% accuracy in the token-based method. The accuracy in the type-based method has been considered as baseline accuracy. The performance comparison of unsupervised approaches is mentioned in Fig. 9. Due to the presence of the word

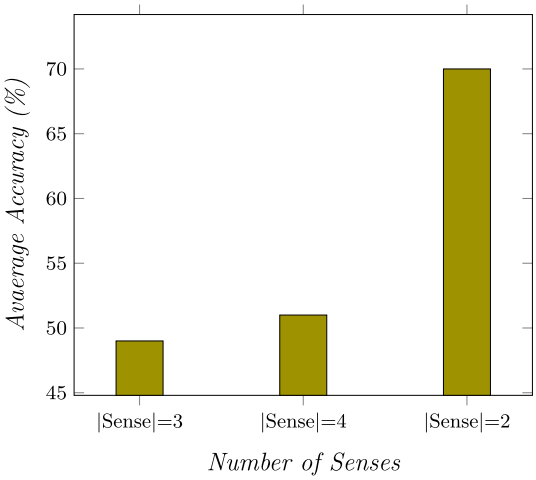


Fig. 7 Average accuracy of each sense type of *Das's Algorithm* tested in *Dataset 4*

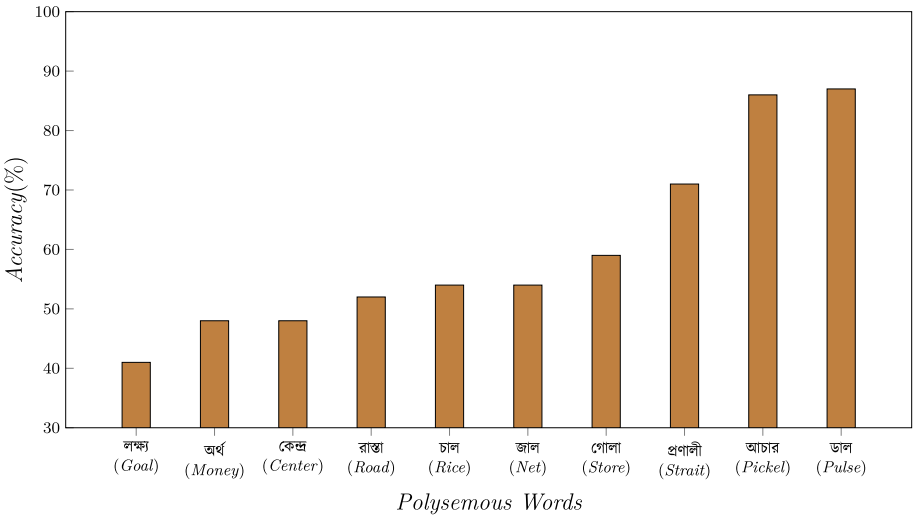


Fig. 8 Recognition accuracy of each polysemous words in *Dataset 4* tested by *Das's Algorithm*

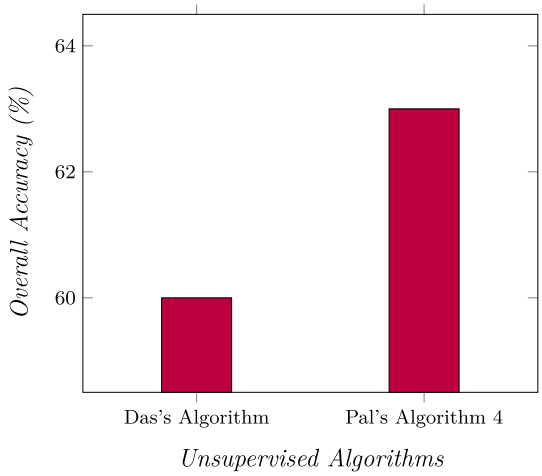


Fig. 9 Performance comparison of unsupervised approaches

lemmatization step, the performance of *Pal's Algorithm 4* is slightly better compared to *Das's Algorithm*.

6.2 Results analysis of knowledge-based approaches

In the knowledge-based approach, *Haque's Algorithm* has been tested on *Dataset 1*. It has been computed success rate by a class of total count of perfectly disambiguated sentences to the total number of processed sentences. The success rate has been delineated on the basis of three parameters, such as (a) classes of ambiguous words, (b) the total number of polysemous words in a sentence and (c) a number of ambiguous words in each sentence. The success rate with these three parameters has been depicted in Fig. 10, where Parameter 1, Parameter 2, and Parameter 3 are indicated types of ambiguous word, length of input sentences, and

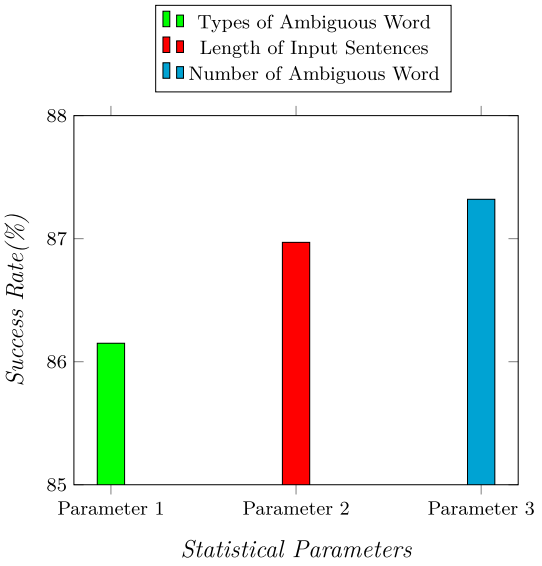


Fig. 10 Success rate of *Haque's Algorithm* in three parameters using *Dataset 1*

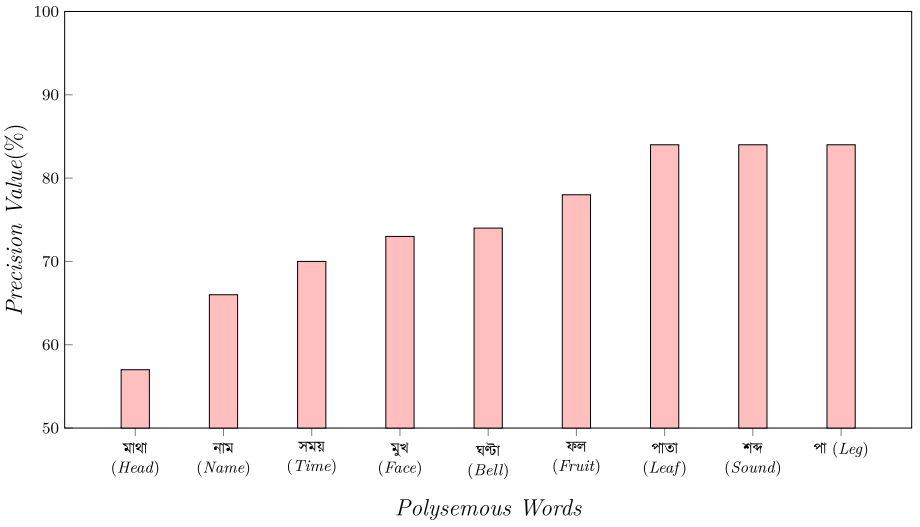


Fig. 11 Accuracy percentage of *Pal's Algorithm 1* using *Dataset 3*

number of ambiguous word respectively. It has been obtained success rate 86.67% in noun, 85.71% in adjective and 85% in verb words. In addition, the system has been also proven in several types of input sentence lengths, such as 3, 4, 5, 6, and 7. It has received optimum success rate at the sentencing length four.

In addition, *Pal's Algorithm 1* has been tested on *Dataset 3* with nine polysemous words. This algorithm has been tried out on nine polysemous words of four hundred eighty-five testing instances. It has been detected correctly over three hundred sixty-five instances, and overall obtained accuracy is 75%. Moreover, it has received 31% baseline accuracy after lemmatization of whole text. The accuracy percentage of this algorithm of nine polysemous words has been depicted in Fig. 11. It has been obtained highest detection accuracy in পাতা

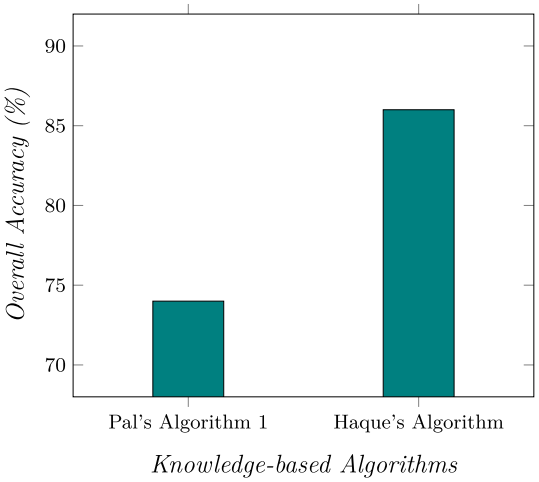


Fig. 12 Performance comparison of knowledge-based approaches

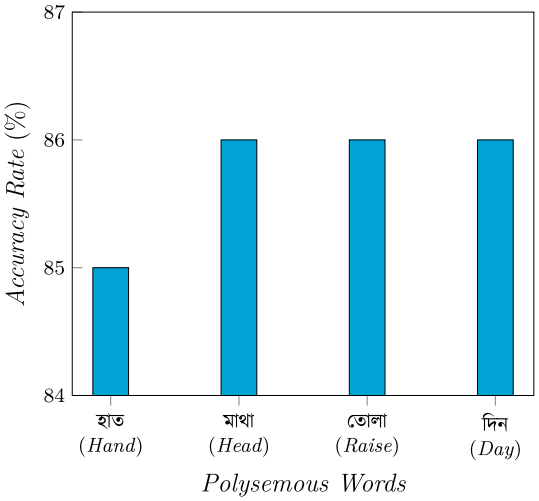


Fig. 13 Accuracy percentage of *Pal's Algorithm 3* using *Dataset 5*

(*Leaf*), শব্দ (*Sound*), and পা (*Leg*). However, মাথা (*Head*) has been received low recognition percentage due to its immense number of inflected terms present in the context.

The algorithms of knowledge-based approaches for disambiguating sense ambiguity have been executed very well in the Bengali context. The performances of both algorithms have been assessed with the help of small dataset. The comparison graph of these two algorithms has been depicted in Fig. 12. Due to the presence of lexical overlap in *Pal's Algorithm 1*, *Haque's Algorithm* has been done well compared to this.

6.3 Results analysis of supervised approaches

In this context, *Pal's Algorithm 3* has been tested on *Dataset 5*. It has been examined only four polysemous words, which are very common in Bengali literature. It has been also demonstrated that a WSD system with lemmatization always gives better results compared to the WSD system without lemmatization. Due to the presence of a huge amount of inflected

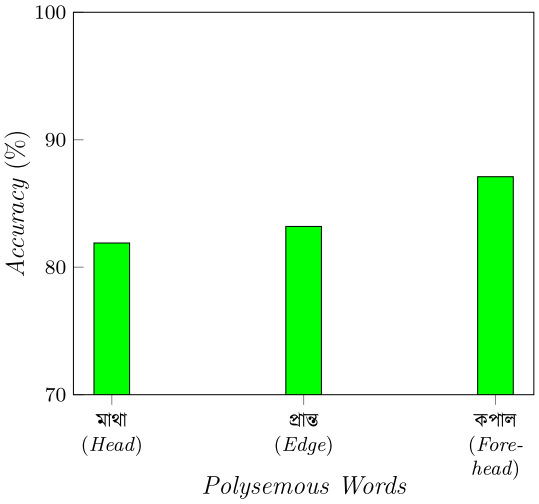


Fig. 14 Accuracy percentage of *Pal's Algorithm 2* using *Dataset 2*

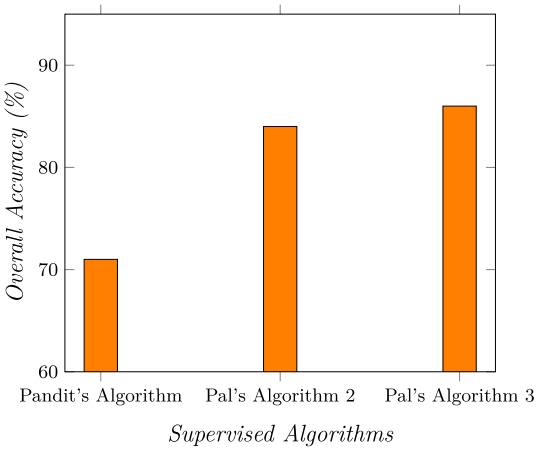


Fig. 15 Performance comparison of supervised approaches

words, vocabulary entry has not found a suitable matched word. The classification of senses of a polysemous word in huge amount inflected forms present in the context is a very tedious task. Subsequently, this algorithm has been found to mean 86% accuracy. The details result has been expressed in Fig. 13.

Consequently, *Pal's Algorithm 2* has been tested on *Dataset 2*. It has been tested on one thousand seven hundred forty-seven sentences and correctly observed on one thousand four hundred sixty-seven sentences. The potentiality of this algorithm has been tried out on three polysemous words (i.e. মাথা (*Head*), কপাল (*Forehead*), and প্রান্ত (*Edge*)) with fifty categories. Each category has been comprised of fifty-five sentences. The performance, accuracy has been remarked in Fig. 14.

In addition, *Pandit's Algorithm* has been tested on *Dataset 6*. The potentiality of this algorithm has been affirmed with the database of twenty-five polysemous words in one hundred sentences. Moreover, the execution of this algorithm has been measured on the basis of four POS (i.e. adjective, adverb, noun, and verb). It has obtained maximum accuracy on noun words and lower limit on the adverb words. The overall accuracy is 71%.

The performance comparison graph of these three supervised-based Bengali WSD algorithms has been depicted in Fig. 15. The performance of *Pal's Algorithm 3* is best to compare to other attacks. WSD with lemmatization step has been taken into consideration in a key player in gaining better recognition accuracy in *Pal's Algorithm 3*. Actually, Bengali literature is a highly morphologically complex word. Due to the presence of inflected forms of the polysemous word, word stemming has been taken on a key role in a WSD system.

7 Conclusion

This paper summarizes the various approaches of word sense disambiguation technique, based on Bengali literature. It classifies the approaches of Bengali WSD task in three subsections (i.e. supervised, knowledge-based and unsupervised). It reviews three algorithms of supervised, two algorithms of knowledge-based, and two of unsupervised technique. The step-wise description of each algorithm is discussed with proper explanation and commented critically. The drawbacks of each algorithm with proper justification are discussed in a very comprehensive way. Besides, this paper reviews a lot of survey works regarding WSD in many most spoken language.

In addition, this paper also sums up the existing dataset in Bengali. It classifies these dataset into four modules (i.e. raw corpora, WordNet, semi-annotated, and MRD). Out of six, two dataset belong to raw corpora and WordNet respectively, and one dataset belongs remaining every two categories. These dataset are discussed in detail with the proper statistical figure. Dataset categorization according to WSD variant (i.e. target word and all word WSD) are also performed.

The analysis of the experimental results of each category of Bengali WSD approaches is discussed in a precise manner. The comparative analysis of all discussed algorithms in each category is performed by graphical plots. The overall best-fitted algorithm in each category is identified with a proper logical explanation. In future, we shall plan to do a review work in various multi-linguistic WSD algorithms.

References

Alian M, Awajan A, Al-Kouz A (2017) Arabic word sense disambiguation-survey. In: 2017 international conference on new trends in computing sciences (ICTCS), pp 236–240. IEEE

Aung NTT, Soe KM, Thein NL (2011) A word sense disambiguation system using Naïve Bayesian algorithm for Myanmar language. Int J Sci Eng Res 2(9):1–7

Baldwin T, Su NK, Bond F, Fujita S, Martinez D, Tanaka T (2008) MRD-based word sense disambiguation: Further extending Lesk

Banerjee S, Pedersen T (2002) An adapted Lesk algorithm for word sense disambiguation using wordnet. In: International conference on intelligent text processing and computational linguistics, pp 136–145. Springer

Bar-Hillel Y (2003) The present status of automatic translation of languages. Read Mach Transl 1:45–77

Bhala RVV, Abirami S (2014) Trends in word sense disambiguation. Artif Intell Rev 42(2):159–171

Borah PP, Talukdar G, Baruah A (2014) Approaches for word sense disambiguation—a survey. Int J Recent Technol Eng 3(1):35–38

Carpuat M, Wu D (2005) Word sense disambiguation vs. statistical machine translation. In: Proceedings of the 43rd annual meeting on association for computational linguistics, pp 387–394. Association for Computational Linguistics

Cottrell GW, Small SL (1983) A connectionist scheme for modelling word sense disambiguation. Cognit Brain Theory 6(1):89–120

Das A, Sarkar S (2013) Word sense disambiguation in Bengali applied to Bengali–Hindi machine translation. In: International conference on natural language processing (ICON), Noida

Dhungana UR, Shakya S (2014) Word sense disambiguation in Nepali language. In: 2014 fourth international conference on digital information and communication technology and it's applications (DICTAP), pp 46–50. IEEE
Elayeb B (2019) Arabic word sense disambiguation: a review. Artif Intell Rev 52(4):2475–2532
Escudero G, Màrquez L, Rigau G (2000) Boosting applied to word sense disambiguation. In: European conference on machine learning, pp 129–141. Springer
Haque A, Haque MM (2016) Bangla word sense disambiguation system using dictionary based approach. ICAICT, Bangladesh
Haroon RP (2010) Malayalam word sense disambiguation. In: 2010 IEEE international conference on computational intelligence and computing research (ICCIC), pp 1–4. IEEE
Ide N, Véronis J (1998) Introduction to the special issue on word sense disambiguation: the state of the art. Comput Linguist 24(1):2–40
Kågebäck M, Salomonsson H (2016) Word sense disambiguation using a bidirectional LSTM. arXiv preprint arXiv:1606.03568
Kalita P, Barman AK (2015) Word sense disambiguation: a survey. Int J Eng Comput Sci 4(05):11743–11748
Lee YK, Ng HT (2002) An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. In: Proceedings of the ACL-02 conference on Empirical methods in natural language processing, vol 10, pp 41–48. Association for Computational Linguistics
Luo F, Liu T, Xia Q, Chang B, Sui Z (2018a) Incorporating glosses into neural word sense disambiguation. arXiv preprint arXiv:1805.08028
Luo F, Liu T, He Z, Xia Q, Sui Z, Chang B (2018b) Leveraging gloss knowledge in neural word sense disambiguation by hierarchical co-attention. In: Proceedings of the conference on empirical methods in natural language processing, pp 1402–141
Lupu, M, Trandabat, D, Husarciuc M (2005) A Romanian semcor aligned to the English and Italian multisemcor. In: 1st romance FrameNet workshop at EUROLAN, pp 20–27. Citeseer
Màrquez L, Escudero G, Martínez D, Rigau G (2007) Supervised corpus-based methods for WSD. In: Agirre E, Edmonds P (eds) Word sense disambiguation. Springer, Berlin, pp 167–216
McCarthy D (1997) Word sense disambiguation for acquisition of selectional preferences. In: Automatic information extraction and building of Lexical semantic resources for NLP applications
Mishra N, Yadav S, Siddiqui TJ (2009) An unsupervised approach to Hindi word sense disambiguation. In: Proceedings of the first international conference on intelligent human computer interaction, pp 327–335. Springer
Moro A, Raganato A, Navigli R (2014) Entity linking meets word sense disambiguation: a unified approach. Trans Assoc Comput Linguist 2:231–244
Navigli R (2009) Word sense disambiguation: a survey. ACM Comput Surv (CSUR) 41(2):10
Pal AR, Saha D (2015) Word sense disambiguation: a survey. arXiv preprint arXiv:1508.01346
Pal AR, Saha D (2017) Word sense disambiguation in Bengali: an unsupervised approach. In: Second international conference on electrical, computer and communication technologies (ICECCT), pp 1–5. IEEE
Pal AR, Saha D, Dash NS (2015a) Automatic classification of Bengali sentences based on sense definitions present in Bengali wordnet. arXiv preprint arXiv:1508.01349
Pal AR, Saha D, Naskar S, Dash NS (2015b) 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), pp 342–346. IEEE
Pal AR, Saha D, Pal A (2017) A knowledge based methodology for word sense disambiguation for low resource language. Adv Comput Sci Technol 10(2):267–283
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), pp 383–386. IEEE
Parameswarappa S, Narayana VN (2013) Kannada word sense disambiguation using decision list. Volume 2:272–278
Pedersen T (2007) Unsupervised corpus-based methods for WSD. In: Agirre E, Edmonds P (eds) Word sense disambiguation. Springer, Berlin, pp 133–166
Popescu M, Hristea F (2011) State of the art versus classical clustering for unsupervised word sense disambiguation. Artif Intell Rev 35(3):241–264
Raganato A, Bovi CD, Navigli R (2017) Neural sequence learning models for word sense disambiguation. In: Proceedings of the conference on empirical methods in natural language processing, pp 1156–1167
Sarmah J, Sarma SK (2016a) Survey on word sense disambiguation: an initiative towards an Indo-Aryan language. IJEM 6(3):37–52
Sarmah J, Sarma SK (2016b) Word sense disambiguation for Assamese. In: 2016 IEEE 6th international conference on advanced computing (IACC), pp 146–151. IEEE

Segond F (2000) Framework and results for French. Comput Humanit 34(1–2):49–60
Shallu, Gupta V (2013) A survey of word-sense disambiguation effective techniques and methods for Indian languages. J Emerg Technol Web Intell 5(4):354–360
Sharma, DK (2015) A comparative analysis of Hindi word sense disambiguation and its approaches. In: International conference on computing, communication & automation. IEEE, pp 314–321
Sinha R, Mihalcea R (2007) Unsupervised graph-basedword sense disambiguation using measures of word semantic similarity. In: International conference on semantic computing, 2007. ICSC 2007, pp 363–369. IEEE
Srinivas M, Rani BP (2016) Word sense disambiguation techniques for Indian and other Asian languages: a survey. Int J Comput Appl 156(8):35–41
Tatar D (2005) Word sense disambiguation by machine learning approach: a short survey. Fundam Inform 64(1–4):433–442
Weaver W (1949) The mathematics of communication. Sci Am 181(1):11–15
Wilks Y (1975) A preferential, pattern-seeking, semantics for natural language inference. Artif Intell 6(1):53–74
Wilks KR, Carter NL (1990) Rheology of some continental lower crustal rocks. Tectonophysics 182(1–2):57–77
Yunfang WU (2009) A survey of Chinese word sense disambiguation: resources, methods and evaluation. Contemp Linguist 2:005
Zhou X, Han H (2005) Survey of word sense disambiguation approaches. In: FLAIRS conference, pp 307–313. Philadelphia
Zouaghi A, Merhbene L, Zrigui M (2012) Combination of information retrieval methods with Lesk algorithm for Arabic word sense disambiguation. Artif Intell Rev 38(4):257–269