Sādhanā (2019) 44:181
https://doi.org/10.1007/s12046-019-1165-2

A novel approach to word sense disambiguation in Bengali language using supervised methodology

ALOK RANJAN PAL[1,*], DIGANTA SAHA[2], NILADRI SEKHAR DASH[3], SUDIP KUMAR NASKAR[2] and ANTARA PAL[4]

[1] Department of Computer Science and Engineering, College of Engineering and Management, Kolaghat 721171, India
[2] Department of Computer Science and Engineering, Jadavpur University, Kolkata 700032, India
[3] Linguistic Research Unit, Indian Statistical Institute, Kolkata 700108, India
[4] Department of Computer Science and Engineering, National Institute of Technology, Durgapur 713209, India
e-mail: chhaandasik@gmail.com; neruda0101@yahoo.com; nisedash@gmail.com; sudip.naskar@gmail.com; antarapal22@gmail.com

MS received 13 July 2017; revised 7 December 2018; accepted 6 June 2019; published online 17 July 2019

Abstract. An attempt is made in this paper to report how a supervised methodology has been adopted for the task of Word Sense Disambiguation (WSD) in Bengali with necessary modifications. At the initial stage, four commonly used supervised methods, Decision Tree (DT), Support Vector Machine (SVM), Artificial Neural Network (ANN) and Naïve Bayes (NB), are developed at the *baseline*. These algorithms are applied individually on a data set of 13 most frequently used Bengali ambiguous words. On experimental basis, the *baseline* strategy is modified with two extensions: (a) inclusion of lemmatization process into the system and (b) bootstrapping of the operational process. As a result, the levels of accuracy of the *baseline* methods are slightly improved, which is a positive signal for the whole process of disambiguation as it opens scope for further modification of the existing method for better result. In this experiment, the data sets are prepared from the Bengali corpus, developed in the Technology Development for Indian Languages (TDIL) project of the Government of India and from the Bengali WordNet, which is developed at the Indian Statistical Institute, Kolkata. The paper reports the challenges and pitfalls of the work that have been closely observed during the experiment.

Keywords. Natural language processing; Word Sense Disambiguation; supervised methodology; lemmatization; bootstrapping.

1. Introduction

In every natural language, there are so many words that carry different senses in different contexts of their use. These words are often recognized as ambiguous words and finding the exact sense of an ambiguous word in a piece of text is known as Word Sense Disambiguation (WSD) [1–5]. For example, the English words *head, run, round, manage,* etc. have multiple senses based on their contexts of use in texts. Finding the exact senses of the words in a given context is the main challenge of WSD. To date, we have come across three major methodologies that are used to deal with this problem, namely, supervised methodology, knowledge-based methodology and unsupervised methodology.

In supervised methodology [6–25], sense disambiguation of words is performed with the help of previously created learning sets. These learning sets contain related sentences for a particular sense of an ambiguous word. The supervised method classifies the new test sentences based on the probability distributions calculated using these learning sets.

The knowledge-based methodology [26–36] depends on external knowledge resources like online semantic dictionaries, thesauri, machine-readable dictionaries, etc. to obtain sense definitions of the lexical components.

In unsupervised methodology [37–39], the sense disambiguation happens in two phases. First, the sentences are clustered using a clustering algorithm and these clusters are tagged with relevant senses with the help of a linguistic expert. Next, a distance-based similarity measuring technique is used to find the closeness of a test data with the sense-tagged clusters. The minimum distance from a sense-tagged cluster leads to assigning the same sense to that test data.

[*] For correspondence

The present work is developed based on the four commonly used supervised methods, namely, the Decision Tree (DT), the Support Vector Machine (SVM), the Artificial Neural Network (ANN) and the Naïve Bayes (NB) for sense classification; in the *baseline* experiment, these methods generate 63.84, 76.9, 76.23 and 80.23% accurate result, respectively, when they are tested on 13 mostly used Bengali ambiguous words; next, two extensions are adopted over the *baseline* strategy to increase the level of accuracy: (a) incorporation of lemmatization process in the system that generates 68.30, 79, 78.23 and 82.30% accuracy, respectively, and (b) operation of bootstrapping on the systems (including lemmatization feature) that produces 70.92, 79.15, 79.53 and 83% accuracy, respectively. Obviously, the additional features and properties have made the proposed technique more robust and less erroneous in generation of outputs.

The organization of the paper is as follows. Section 2 presents a brief survey of this research methodology. In section 3, experimental set-up is described. The proposed approach is demonstrated in section 4. In section 5, extensions on the *baseline* methodology are described in detail. The report is concluded with future scope in section 6.

2. Survey

In the case of supervised methodology, manually created learning sets are used to train the model. The learning sets consist of example sentences relating to a particular sense of a word. The test instances are classified based on their probability distributions, calculated using the learning sets. Some commonly used approaches deployed in this methodology are discussed here.

2.1 *Decision list*

In the Decision List [36, 40]—based approach, first, a set of rules are formed for a target word. Next, a part of the example sentences are fed to the system to calculate the decision parameters like feature value, sense score, etc. When a test data comes for classification task, these feature values categorize that data to a particular class using these parameters.

2.2 *DT*

The DT [41–43]—based approach frames the rules in the form of a tree structure (figure 1) where the non-leaf nodes denote the tests and the branches represent the test results. The leaf nodes of the tree carry the different senses. If a set of rules can guide an execution to a leaf node then the sense is assigned to that word as a derived sense.

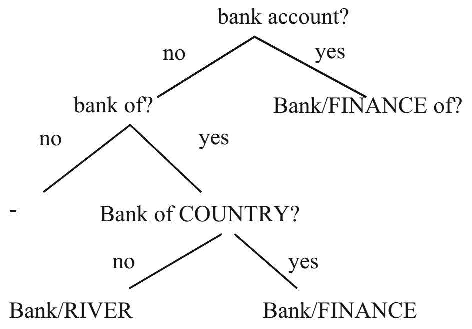


Figure 1. An example of a decision tree.

2.3 *NB classification algorithm*

NB classifier [44–46] is a powerful algorithm for the classification task based on Bayes theorem. The Bayes theorem is stated by the following equation:

$$\boldsymbol{P}(\mathbf{A}|\mathbf{B}) = \frac{\boldsymbol{P}(\mathbf{B}|\mathbf{A})\boldsymbol{P}(\mathbf{A})}{\boldsymbol{P}(\mathbf{B})}$$

where A is called the *proposition* and B is called the *evidence*. P(A) is called the prior probability of proposition and P(B) is called the prior probability of evidence. P(A|B) is called the *posterior* and P(B|A) is the *likelihood*.

The class with the highest membership probability for a data point is considered as the most likely class for that data point.

2.4 *ANN-based classification*

ANN [47–50] is a model of artificial neurons that works similar to a neural structure of brain. This model processes one input at a time and assigns it to an arbitrary class. Next, this allocation is verified with a known output. The errors from every iteration stage are fed back to the model to rectify the errors for the next iterations.

2.5 *Exemplar-based strategy*

In Exemplar-based [51] strategy, the examples are considered as points, distributed over a feature space. When a new data point comes to be categorized, any distance-based similarity measuring technique is used to find the closeness of the data point w.r.t. all the other classifiers. The minimum distance w.r.t. a particular classifier represents the sense of the test data.

2.6 *SVM-based algorithms*

In SVM-based [52–54] strategy, examples are treated as polarized points, either positive or negative. The goal of the methodology is to separate these positive and negative points w.r.t. a hyper-plane. A test data is classified by evaluating, at which side of the hyper-plane the point belongs to.

2.7 *Ensemble methods*

In the Ensemble methods [55], the classifiers are combined after every execution for a better classification result. This combination occurs according to different parameters, such as Majority Voting, Probability Mixture, Rank-based Combination, AdaBoost [56, 57], etc.

3. Experimental set-up

3.1 *The Bengali corpus*

The Bengali corpus used in this research work was developed in the Technology Development for Indian Languages (TDIL) project of the Government of India. This corpus contains text samples from 85 text categories or subject domains like Physics, Chemistry, Mathematics, Agriculture, Botany, Child Literature, Mass Media, etc. covering 11,300 number of A4 pages; 271,102 number of sentences; 3,589,220 number of words in their inflected and non-inflected forms and 199,245 number of distinct words. Each of the distinct words appears in the corpus with a different frequency of occurrence. For example, the word মাথা (māthā) "head" occurs 968 times, মাথার (māthār) "of head" occurs 398 times and মাথায় (māthāy) "on head" occurs 729 times, followed by other inflected forms like মাথাতেই (māthātei) "in head itself" occurring 3 times, মাথাটা (māthāṭā) "the head" occurring 112 times, মাথাটি (māthāṭi) "the head" occurring 13 times, মাথাগুলো (māthāgulo) "heads" occurring 3 times, etc. This corpus is exhaustively used in this work to extract sentences containing a particular ambiguous word.

3.2 *Selection of ambiguous word*

Theoretically it is possible to assume that any Bengali word can appear in a text with certain level of ambiguity. People of computational linguistics like to use several constraints from implementation perspective to select the ambiguous words. As mentioned earlier, the Bengali text corpus contains 199,245 distinct words. First, these words are arranged in decreasing order according to their term frequency in the corpus. The most frequently used words are then selected for experiment with some necessary pre-requisite conditions as discussed in section 3.3.

3.3 *Selection of senses of the ambiguous words for experiment*

After retrieving the ambiguous words, a set of steps have been defined and executed to select their multiple senses for the experiment. The range of sense variation of Bengali words is so vast that it appears as a real challenge to select a few senses from them for the experiment. For example, according to the *Sansad Banglā Avidhān*, the word "হাত" (hāt) can denote more than 80 (eighty) different senses, both in its singular and conjugate forms, whereas the Bengali WordNet lists only 14 (fourteen) distinct senses for the word. On the contrary, the TDIL Bengali text corpus provides only 4 (four) different senses of this word with some needful number of sentences.

In this experiment, a particular sense of an ambiguous word is considered for evaluation process when at least 20 sentences (threshold) are present in the corpus having that particular sense.

As the supervised methodologies depend on some learning sets initially sense-tagged for classification of test data, for an individual ambiguous word, only those senses are considered for evaluation that follow the afore-mentioned criteria.

The selected senses for the experiment are listed in table 1.

3.4 *Text normalization*

The texts stored in the TDIL Bengali corpus are non-normalized in nature. Hence, the very first job was to normalize the texts adequately by (a) removing uneven number of spaces, new lines, etc., (b) discarding comma, colon, semi colon, double quote, single quote and all other orthographic symbols, (c) converting the whole texts into Unicode-compatible single Bengali font (*Vrinda* in this work) and (d) considering all types of Bengali sentence termination symbols, such as note-of-exclamation, note-of-interrogation and *purnacched* (full stop) ("।").

3.5 *Removal of function words*

In the field of linguistic study, the nouns, verbs, adjectives and adverbs are called as *content* parts of speech (POS), and *function* words are those words that exist in a sentence to explain or create grammatical or structural relationships into which the *content* words may fit.

In the research works in Natural Language Processing (NLP), there is no specific rule or process to differentiate between the *content* word and *function* word; rather, it is more or less based on nature of the NLP work. Although theoretically all the Bengali words carry some important meaning in every sentence, in computational environment, considering all words in a text creates two problems: first, sometimes the size of the vocabulary (distinct word) goes out of the computational power of a system, and second, the context analysis of a target word cannot retrieve sufficient meaningful information from the *function* words of its surrounding. To deal with these problems, after lemmatization process, the words except nouns, verbs, adjectives and adverbs (in Bengali, adverbs are also treated as a kind of adjective) are eliminated from the texts as they are *function* words.

Table 1. Selected senses of the ambiguous words.

Word	No. of sentences	Selected senses
“মাথা” (māthā)	96	প্রান্ত (prānta: edge), মস্তক (mastak: head), চিন্তা (chintā: thought)
“হাত” (hāt)	80	হস্ত (hasta: hand), অবদান (abadān: contribution), হাতবদল (hātbadal: handover), হাতপাতা (hātpātā: beg)
“ঘণ্টা” (ghantā)	90	ষাট মিনিট (sāt minute: sixty minutes), বাদ্য যন্ত্র (bādya yantra: bell)
“ঘর” (ghar)	93	গৃহ (griha: home), সংসার করা (sansārkarā: living family life), বংশ (bansha: family culture)
“পাতা” (pātā)	90	গাছের পাতা (gācher pātā: leaf), পৃষ্ঠা (prishthā: page), অক্ষি পল্লব (akshi pallab: eye leaf), বিছানো (bichāno: unfold)
“জল” (jal)	119	বারি (bāri: water), অশ্রু (ashru: tear), ঘটনা প্রবাহ (ghatanā prabāha: flow of incident), জিভে জল (jive jal: saliva)
“নাম” (nām)	80	নাম (nām: name), সুনাম (sunām: praise), জপ (jap: chant)
“যোগ” (yog)	113	যোগদান করা (yogdān karā: participate), যোগফল (yogfal: add), সম্পর্ক (samparka: relation)
“ফল” (fal)	100	বীজকোষ (beejkosh: fruit), পরিণতি (parinati: result)
“মানুষ” (mānush)	138	নর (nar: homo sapiens), ব্যক্তি (byakti: person), লালন পালন (lālan pālan: nourishing)
“মুখ” (mukh)	90	অঙ্গ (anga: organ), খোলা অংশ (kholā ansha: opening), অভিমুখ (abhimukh: direction)
“শব্দ” (shabda)	118	ধ্বণী (dhwanee: sound), অক্ষর (akshar: word)
“সময়” (samay)	97	কাল (kāl: in time), ক্ষণ (kshan: duration of time), অবসর (abasar: leisure time)

3.6 *Performance evaluation*

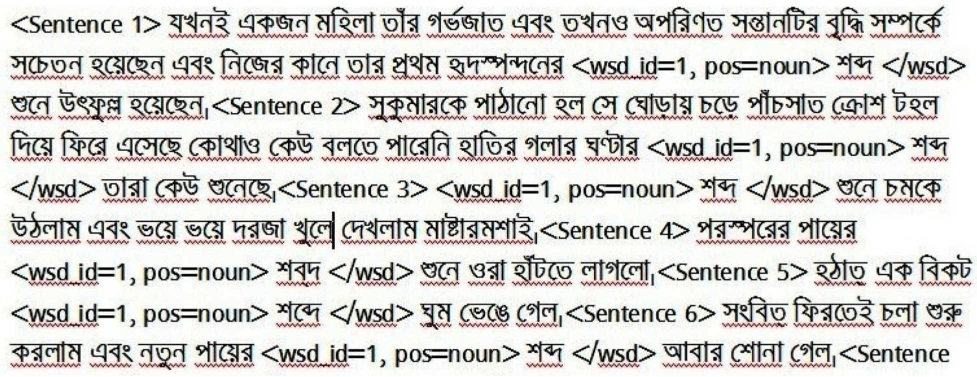
In the proposed work, the system identifies all the target words in the data set for evaluation and resolves senses for all of them either correctly or wrongly. For this reason, the performance of the systems is evaluated by the “percentage-of-accuracy” throughout the work.

3.7 *Preparation of data set*

3.7a *Annotation of input data*: After text normalization process, the input sentences are annotated for the experiment in the following way:

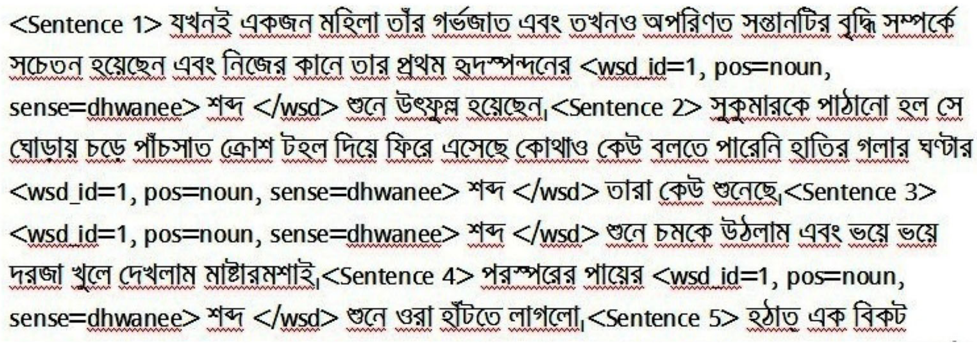
<Sentence x> tag at the beginning of each sentence represents the sentence number. The target word is bounded by two tags. In the preceding tag, “*wsd_id*” represents the ambiguous word number (as this experiment deals with single-word-wsd, *wsd_id* is considered as (1) in the sentence and “*pos*” represents the part-of-speech of the target word in that particular sentence (see figure 2).

3.7b *Preparation of reference output data*: The reference output files are prepared earlier with the help of a standard Bengali dictionary (*Sansad Banglā Avidhān*) (see figure 3). The system-generated results are verified programmatically with these reference outputs. Annotations of these sentences are similar to the input sentences, except that the actual senses of the ambiguous words are mentioned in the tag.



<Sentence 1> যখনই একজন মহিলা তাঁর গর্ভজাত এবং তখনও অপরিণত সন্তানটির বৃদ্ধি সম্পর্কে সচেতন হয়েছেন এবং নিজের কানে তার প্রথম হৃদস্পন্দনের <wsd_id=1, pos=noun> শব্দ </wsd> শুনে উৎফুল্ল হয়েছেন।<Sentence 2> সুকুমারকে পাঠানো হল সে ঘোড়ায় চড়ে পাঁচসাত ক্রোশ টহল দিয়ে ফিরে এসেছে কোথাও কেউ বলতে পারেনি হাতির গলার ঘণ্টার <wsd_id=1, pos=noun> শব্দ </wsd> তারা কেউ শুনেছে।<Sentence 3> <wsd_id=1, pos=noun> শব্দ </wsd> শুনে চমকে উঠলাম এবং ভয়ে ভয়ে দরজা খুলে দেখলাম মাষ্টারমশাই।<Sentence 4> পরস্পরের পায়ের <wsd_id=1, pos=noun> শবদ </wsd> শুনে ওরা হাঁটতে লাগলো।<Sentence 5> হঠাত্ এক বিকট <wsd_id=1, pos=noun> শব্দে </wsd> ঘুম ভেঙে গেল।<Sentence 6> সংবিত্ ফিরতেই চলা শুরু করলাম এবং নতুন পায়ের <wsd_id=1, pos=noun> শব্দ </wsd> আবার শোনা গেল।<Sentence

Figure 2. Partial view of a sample input file.



<Sentence 1> যখনই একজন মহিলা তাঁর গর্ভজাত এবং তখনও অপরিণত সন্তানটির বৃদ্ধি সম্পর্কে সচেতন হয়েছেন এবং নিজের কানে তার প্রথম হৃদস্পন্দনের <wsd_id=1, pos=noun, sense=dhwanee> শব্দ </wsd> শুনে উৎফুল্ল হয়েছেন।<Sentence 2> সুকুমারকে পাঠানো হল সে ঘোড়ায় চড়ে পাঁচসাত ক্রোশ টহল দিয়ে ফিরে এসেছে কোথাও কেউ বলতে পারেনি হাতির গলার ঘণ্টার <wsd_id=1, pos=noun, sense=dhwanee> শব্দ </wsd> তারা কেউ শুনেছে।<Sentence 3> <wsd_id=1, pos=noun, sense=dhwanee> শব্দ </wsd> শুনে চমকে উঠলাম এবং ভয়ে ভয়ে দরজা খুলে দেখলাম মাষ্টারমশাই।<Sentence 4> পরস্পরের পায়ের <wsd_id=1, pos=noun, sense=dhwanee> শব্দ </wsd> শুনে ওরা হাঁটতে লাগলো।<Sentence 5> হঠাত্ এক বিকট

Figure 3. Partial view of a reference output data.

The outputs generated by the program have the same annotation like this reference output. Therefore, the two results are compared programmatically.

4. Proposed approach

In the proposed approach, first of all, four commonly used supervised methods, DT, SVM, ANN and NB, are used as the *baseline* strategy for sense classification. These algorithms are tested on 13 mostly used ambiguous words. The data sets are prepared from the Bengali corpus and the Bengali WordNet.

In the next phase, two modifications are adopted over this *baseline* strategy: (a) lemmatization of the whole system and (b) bootstrapping. These two modifications are tested over the same data sets used in the *baseline* experiment. In the evaluation stage, it is observed that the modified approaches produce a better accuracy than the *baseline* strategy.

4.1 *Flow chart of the baseline strategy*

The *baseline* strategy can be represented by the following diagram (figure 4):

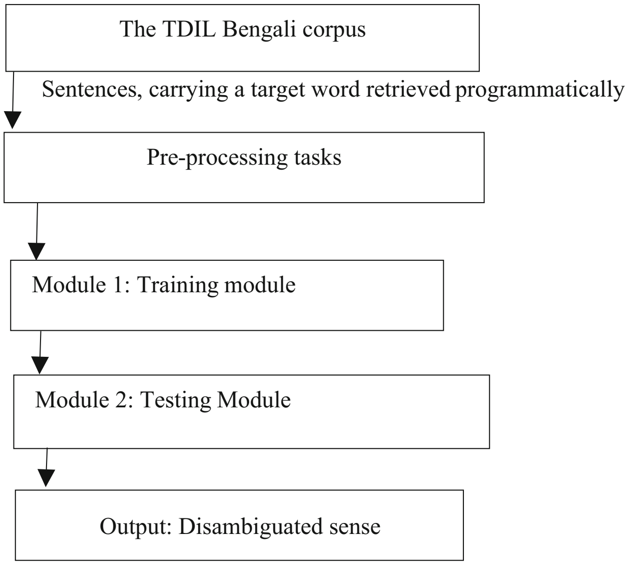

The TDIL Bengali corpus
Sentences, carrying a target word retrieved programmatically
Pre-processing tasks
Module 1: Training module
Module 2: Testing Module
Output: Disambiguated sense

Figure 4. Flow chart of the proposed *baseline* strategy.

The flowchart in figure 4 depicts the overall *baseline* strategy. First, the sentences carrying the selected ambiguous words are retrieved programmatically from the TDIL corpus. Initially, these sentences are non-normalized in nature. Hence, they are passed through a series of preprocessing steps such as normalization, annotation (see section 3.7), etc. Next, some portion of the normalized data sets is used for preparing the training module and remaining is used for testing purpose (i.e., split in 3:1 ratio of training set and test set for *4-fold cross-validation*). Finally, the sense-resolved test sentences are evaluated programmatically by comparing to a reference result (see section 3.7).

4.2 *Result in the baseline experiment*

In the *baseline* strategy, four commonly used supervised methods, DT, SVM, ANN and NB, are used for sense classification. The algorithms are tested individually on the same data sets using *4-fold cross-validation*, which effectively results in 3:1 ratio of training set and test set. The results are presented in the form of "percentage-of-accuracy", because the systems identify all the test instances for evaluation and assign a sense to each of them either correctly or wrongly. Some of the test cases produced an appreciable accuracy, but some of them did not perform up to the mark. It is due to the syntactic and semantic varieties in sentence structures, which are directly related to the lexical similarity measure and thus the varieties in accuracy as well.

Table 2 depicts the average percentage of accuracy of the four methods at the *baselin*e.

5. Extensions of the *baseline* methodology

To enhance the performance of the *baseline* methodology, the following two extensions have been adopted: (a) lemmatization of the whole system and (b) bootstrapping.

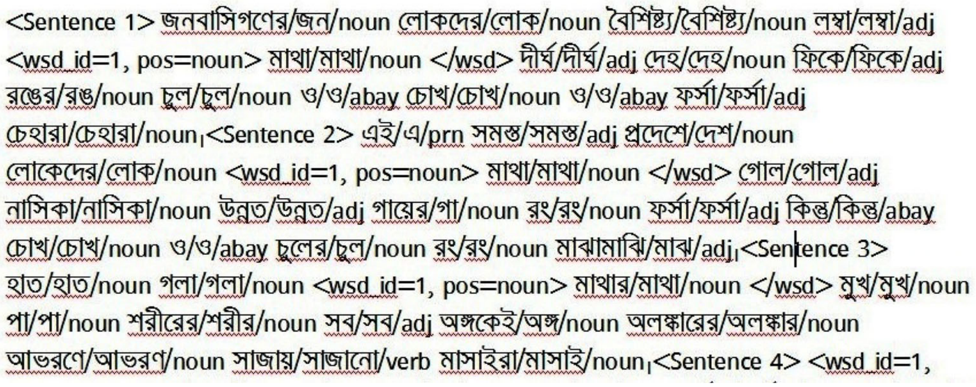
5.1 *Lemmatization of the whole system*

Since Bengali is a morphologically very strong language, the lexical matching between the inflected words is not adequate for measuring the similarity between the words. To overcome this bottleneck, the whole system has been operated on the lemmatized forms of the words [58]. The expansion of lexical coverage due to this lemmatization task generates such a situation where more number of lexical similarities are observed between the instances, which eventually leads the system to act in a robust manner to achieve higher level of accuracy. The lemmatization tool operated on the training data and test data in a uniform manner without any selectional bias.

Partial view of a sample lemmatized input data is presented in figure 5. Annotation of the sentences follows the same strategy as in the *baseline* experiment (see section 3.7); in addition, the words are in lemmatized form. Words are represented in the following format: "inflected-word/corresponding-stem-form/POS". The experiment is

Table 2. Execution of the *baseline* model.

Word	Decision Tree (%)	SVM (%)	ANN (%)	Naïve Bayes (%)
"**ঘর** "(ghar)	78	82	84	82
"**জল**" (jal)	62	83	84	84
"ঘণ্টা" (ghantā)	66	82	82	85
"**যোগ**" (yog)	54	58	58	72
"পাতা" (pātā)	71	82	81	87
"**মানুষ** "(mānush)	58	86	74	79
"নাম" (nām)	61	75	75	82
"**শব্দ**" (shabda)	82	88	86	85
"**ফল**" (fal)	58	65	63	70
"**সময়** "(samay)	78	83	81	83
"**মুখ**" (mukh)	56	65	65	75
"**মাথা** "(māthā)	56	76	83	79
"**হাত** "(hāt)	50	75	75	80
Average % of accuracy	63.84	76.9	76.23	80.23



```
<Sentence 1> জনবাসিগণের/জন/noun লোকদের/লোক/noun বৈশিষ্ট্য/বৈশিষ্ট্য/noun লম্বা/লম্বা/adj
<wsd_id=1, pos=noun> মাথা/মাথা/noun </wsd> দীর্ঘ/দীর্ঘ/adj দেহ/দেহ/noun ফিকে/ফিকে/adj
রঙের/রঙ/noun চুল/চুল/noun ও/ও/abay চোখ/চোখ/noun ও/ও/abay ফর্সা/ফর্সা/adj
চেহারা/চেহারা/noun।<Sentence 2> এই/এ/prn সমস্ত/সমস্ত/adj প্রদেশে/দেশ/noun
লোকেদের/লোক/noun <wsd_id=1, pos=noun> মাথা/মাথা/noun </wsd> গোল/গোল/adj
নাসিকা/নাসিকা/noun উন্নত/উন্নত/adj গায়ের/গা/noun রং/রং/noun ফর্সা/ফর্সা/adj কিন্তু/কিন্তু/abay
চোখ/চোখ/noun ও/ও/abay চুলের/চুল/noun রং/রং/noun মাঝামাঝি/মাঝ/adj।<Sentence 3>
হাত/হাত/noun গলা/গলা/noun <wsd_id=1, pos=noun> মাথার/মাথা/noun </wsd> মুখ/মুখ/noun
পা/পা/noun শরীরের/শরীর/noun সব/সব/adj অঙ্গকেই/অঙ্গ/noun অলঙ্কারের/অলঙ্কার/noun
আভরণে/আভরণ/noun সাজায়/সাজানো/verb মাসাইরা/মাসাই/noun।<Sentence 4> <wsd_id=1,
```

Figure 5. A sample lemmatized input data.

carried out on the root forms of the words to increase the lexical coverage of the words.

5.1a *Execution in lemmatized environment*: This expansion approach uses the same reference output files used in the *baseline* experiment. Though the inputs are prepared in lemmatized form, the outputs are generated in surface level form of the words to conduct a similar comparison with the *baseline* experiment. The same supervised methods, same ambiguous words and the same *4-fold cross-validation* technique used in the *baseline* strategy are adopted in this phase of experiment. Like the *baseline* experiment, the results are presented in the form of "percentage-of-accuracy", because the systems identify all the test instances for evaluation and assign a sense to each of them either correctly or wrongly. In table 3, the performances of the algorithms on the lemmatized data (i.e. lemmatized form of the *baseline* data set) are presented.

In table 3, it is observed that the overall accuracy has been increased due to the expansion of lexical coverage of the words. As the size of the data sets taken for the experiment is quite small, at several occasions the algorithm returns the same accuracy. In these cases, the

Table 3. Performance of the algorithms on lemmatized form of the *baseline* data set.

Word	Decision Tree (%)	SVM (%)	ANN (%)	Naïve Bayes (%)
"ঘর" (ghar)	80	83	84	84
"জল" (jal)	62	83	84	84
"ঘণ্টা" (ghantā)	66	82	82	85
"যোগ" (yog)	66	72	70	78
"পাতা" (pātā)	71	82	81	87
"মানুষ" (mānush)	68	87	79	82
"নাম" (nām)	64	77	77	83
"শব্দ" (shabda)	82	88	86	85
"ফল" (fal)	63	67	66	75
"সময়" (samay)	78	83	81	83
"মুখ" (mukh)	61	67	67	79
"মাথা"(māthā)	72	80	84	84
"হাত"(hāt)	55	76	76	81
Average accuracy (%)	68.30	79	78.23	82.30

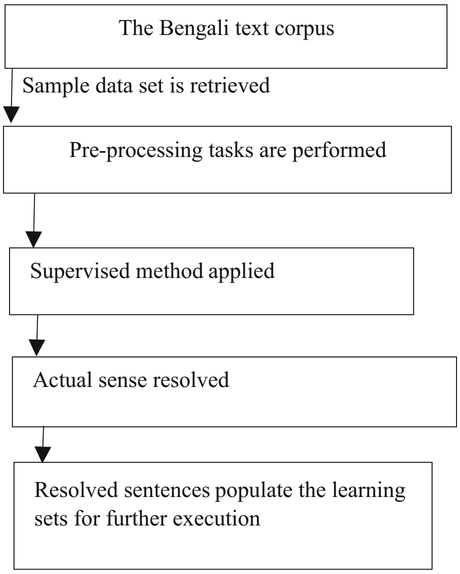

The Bengali text corpus
Sample data set is retrieved
Pre-processing tasks are performed
Supervised method applied
Actual sense resolved
Resolved sentences populate the learning sets for further execution

Figure 6. Flowchart of the proposed bootstrapping technique.

lemmatization process cannot produce any effectively new instance that can enhance the lexical overlap process.

5.2 *Bootstrapping*

In this extended methodology, the sense-resolute test data in a particular phase of execution is inserted into the training sets to enrich the learning procedure. As the training sets become stronger in every execution, the system produces a better accuracy in its next executions. A small manual intervention was mandatory in this phase. As the classification of a test data depends on the probability measures based on the training sets, the methodology demands a correctly populated training set for sense retrieval. However, the proposed model could not produce an absolute result in a particular execution. Hence, to generate an error-free training model, all the misclassified instances are further rectified manually, which leads the system towards a right direction (figure 6).

5.1b *Execution of bootstrapping technique:* In this phase of experiment, two consecutive executions are considered. As, in the previous experiment (see section 5.1) it is observed that performance of the algorithms increases due to lemmatization task, the bootstrapping strategy is also developed in lemmatized environment.

In the first phase, the module is tested on the data set used in the previous experiment (see section 5.1). In the second phase, after the training sets are auto-incremented, a new set of data is selected from the corpus for experiment. The efficiencies of the systems are measured using *4-fold cross-validation* technique, which effectively results in 3:1 ratio of training set to test set. The accuracy of the result in both the phases is presented in table 4. Like the previous two experiments (*baseline* and lemmatization), the results are presented in the form of "percentage-of-accuracy" because the systems identify the entire test instances for evaluation and assign a sense to each of them either correctly or wrongly.

It is observed in the previous two experiments (sections 5.1 and 5.2) that extensions on the *baseline* methodology can produce a better result in most of the cases (tables 3 and 4). However, in a few cases, the accuracy level has slightly dropped. Through investigation it is observed that the accuracy of the system depends on several parameters such as the following.

(a) *Same sense with no contextual similarity*: for example "যে খালের কথা তোমাকে বলেছি তাতে একফোঁটাও জল নেই।" and "এই রশ্মি জীবাণু নাশক এর দ্বারা জল জীবাণু মুক্ত করা হয়।". In these two sentences, the ambiguous word "জল" carries the same sense in every sentence, but there is no contextual similarity in the sentences. Establishing a semantic relation in this type of sentences is a big challenge in computational environment.

(b) *Occurrence of same lexical entries in semantically dissimilar sentences*: for example "সেই যুগে মানুষ ছিল যাযাবর প্রকৃতির।" and "ভূপর্যটক কলম্বাস ছিলেনযাযাবর প্রকৃতির মানুষ।". This mentioned sentence pair is composed of similar *content* words but they represent different senses for the ambiguous word "মানুষ".

(c) *Presence of multiple sense carrying contextual words in a single sentence*: for example

পান্ডুলিপির ধূসর পাতায় তাঁর আত্মজীবনী আজও এতটাই জীবন্ত যে একবার পড়তে শুরু করলে চোখের পাতা পড়েনা।.

In this sentence, while disambiguating the word "পাতা", the word "পান্ডুলিপি" is a contextual word for the sense "পৃষ্ঠা"; the word "ধূসর" is a contextual word for the sense "পৃষ্ঠা", as well as "গাছের পাতা"; and "চোখ" is a contextual word for the sense "অক্ষি পল্লব".

(d) *Sentence with sense anomaly*: for example

"সে পরীক্ষার চারিদিকে এত সংযমের বেষ্টন যে সামান্য মানুষ তেমন উপভোগ লাভ করিবার সহিষ্ণুতা সঞ্চয় করিতে পারে না।".

Table 4. Result of bootstrapping strategy.

Word	Decision Tree (%)		SVM (%)		ANN (%)		Naïve Bayes (%)	
	1st iteration	2nd iteration	1st iteration	2nd iteration	1st iteration	2nd iteration	1st iteration	2nd iteration
“ঘর” (ghar)	80	79	83	80	84	80	84	81
“জল” (jal)	62	62	83	83	84	84	84	84
“ঘণ্টা” (ghantā)	66	64	82	81	82	80	85	84
“যোগ” (yog)	66	72	72	78	70	79	78	77
“পাতা” (pātā)	71	72	82	82	81	82	87	87
“মানুষ” (mānush)	68	71	87	83	79	82	82	86
“নাম” (nām)	64	66	77	78	77	80	83	84
“শব্দ” (shabda)	82	81	88	83	86	82	85	84
“ফল” (fal)	63	64	67	67	66	68	75	78
“সময়” (samay)	78	79	83	85	81	86	83	87
“মুখ” (mukh)	61	61	67	67	67	67	79	79
“মাথা” (māthā)	72	81	80	83	84	84	84	85
“হাত” (hāt)	55	70	76	79	76	80	81	83
Average % of accuracy	68.30	70.92	79	79.15	78.23	79.53	82.30	83

For this type of sentence, it becomes very tough to tag a particular sense even by human judgment.

(e) *Very large sentence, containing a lot of irrelevant information in it*: for example

> কেহ বা দুই কানে আঙুল চাপিয়া ঝুপ ঝুপ করিয়া দ্রুতবেগে কতকগুলো ডুব পাড়িয়া চলিয়া যাইত, কেহ বা ডুব না দিয়া গামছায় জল তুলিয়া ঘন ঘন মাথায় ঢালিতে থাকিত, কেহ বা জলের উপরিভাগের মলিনতা এড়াইবার জন্য বারবার দুই হাতে জল কাটাইয়া লইয়া হঠাত্ একসময়ে ধাঁ করিয়া ডুব পাড়িত, কেহ বা উপরের সিঁড়ি হইতেই বিনা ভূমিকায় সশব্দে জলের মধ্যে ঝাঁপ দিয়া পড়িয়া আত্মসমর্পণ করিত, কেহ বা জলের মধ্যে নামিতে নামিতে এক নিশ্বাসে কতকগুলি শ্লোক আওড়াইয়া লইত, কেহ বা ব্যস্ত কোনোমতে স্নান সারিয়া লইয়া বাড়ি যাইবার জন্য উত্সুক, কাহারো বা ব্যস্ততা লেশমাত্র নাই ধীরেসুস্থে স্নান করিয়া জপ করিয়া গা মুছিয়া কাপড় ছাড়িয়া কোঁচাটা দুই তিনবার ঝাড়িয়া বাগান হইতে কিছু বা ফুল তুলিয়া মৃদুমন্দ দোদুল গতিতে স্নানস্নিগ্ধ শরীরের আরামটিকে বায়ুতে বিকীর্ণ করিতে করিতে বাড়ির দিকে তাহার যাত্রা।.

(f) *Very short sentence, containing insufficient information for computation*: for example

নে, জল আন।.

(g) *Spelling error*: dealing with the spelling errors in the words is also a big challenge in this work. The dissimilar use of ‘শ’, ‘ষ’, ‘স’ ; ‘ি’, ‘ী’;‘ু’, ‘ূ’; ‘ত্’, ‘ৎ’ and different typographical mistakes in the words create a major problem in lexical matching. These errors could be managed easily by a human-driven system, but in an automated system, these spelling errors directly affect the output.

(h) *Scarcity of information in WordNet*: The Bengali WordNet is in developing phase, so it is not a complete reference for retrieving the semantic information of the Bengali words. For example:

(i) The different sense definitions of the commonly used Bengali words are missing in this dictionary, such as মানুষ (single sense present), পড়া (absent in the dictionary), etc., and a few common words in inflected forms (such as নীচে, ধরে, ফলে, মনে, etc.) are also absent in this dictionary.

(ii) A few sense definitions are found in the WordNet that are absent in the standard lexical dictionary, as well as those unknown to the linguistic experts also, such as

Sl. no.	POS	Gloss	Example	Word
27588	Noun	সাধনা রূপে উপলব্ধ সেই সময়াবধি যা কারও নিয়ন্ত্রণে আছে	আমার খাবার খাওয়ার সময় নেই	সময়
33958	Noun	যজ্ঞ বা হোমের সেই সময় যখন বৈদিক স্তোত্রের পাঠ করা হয়	সময়ের পর সবাই হোমের সামগ্রী হোম কুন্ডে ঢেলে দিল	সময়

(iii) A few common relations among the words are not established (properly/not at all) in this online dictionary, such as hypernymy, hyponymy, holonymy, meronymy, antonymy, etc.

6. Conclusion and future scope

In this paper the work for WSD in Bengali language has been proposed using four supervised classification algorithms at the *baseline*, which is supported with two relevant extensions, namely, lemmatization and bootstrapping. Due to lemmatization, lexical coverage of the inflected words is increased, which yields more lexical similarity, causing better accuracy than the *baseline* result. In bootstrapping strategy, more enriched training sets in every iteration resolve a better result in every next iteration.

In reality, the complex linguistic nature of the South Asian languages like Hindi, Bengali, Tamil, Telugu, Punjabi, Malayalam, Marathi, etc. usually puts several challenges in the form of fonts, texts, morphological complexities, etc. At the same time the variation of senses of words, diversities in sentence structures and complex formation of content word and function words, etc. demand additional attention for achieving better result from such experiments.

A dedicated research work might be carried out on identification of the *function* words and *content* words, identification of singular form and conjugate form of the words, accurate all-word lemmatization and all-word POS tagging, handling the sense distinctions of the Bengali words, etc. for better performance from such algorithms.

References

[1] Ide N and Véronis J 1998 Word sense disambiguation: the state of the art. *Comput. Linguist.* 24(1): 1–40

[2] Cucerzan R S, Schafer C and Yarowsky D 2002 Combining classifiers for word sense disambiguation. *Nat. Lang. Eng.* 8(4): 327–341

[3] Nameh M S, Fakhrahmad M and Jahromi M Z 2011 A new approach to word sense disambiguation based on context similarity. In: *Proceedings of the World Congress on Engineering*, vol. I

[4] Xiaojie W and Matsumoto Y 2003 Chinese word sense disambiguation by combining pseudo training data. In: *Proceedings of the International Conference on Natural Language Processing and Knowledge Engineering*, pp. 138–143

[5] Navigli R 2009 Word sense disambiguation: a survey. *ACM Comput. Surv.* 41(2): 1–69

[6] Xiaojie W and Matsumoto Y 2003 Chinese word sense disambiguation by combining pseudo training data. In: *Proceedings of the International Conference on Natural Language Processing and Knowledge Engineering*, pp. 138–143

[7] Sanderson M 1994 Word sense disambiguation and information retrieval. In: *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'94*, July 03–06, Dublin, Ireland. New York: Springer, pp. 142–151

[8] Eneko Agirre E and Edmonds P (Eds.) *Word Sense Disambiguation: Algorithms and Applications*

[9] Seo H, Chung H, Rim H, Myaeng S H and Kim S 2004 Unsupervised word sense disambiguation using WordNet relatives. *Comput. Speech Lang.* 18(3): 253–273

[10] Miller G *et al* 1991 Introduction to WordNet: an on-line lexical database. *Int. J. Lexicogr.* 3(4): 235–244

[11] Kolte S G and Bhirud S G 2008 Word sense disambiguation using WordNet domains. In: *Proceedings of the First International Conference on Digital Object Identifier*, pp. 1187–1191

[12] Liu Y, Scheuermann P, Li X and Zhu X 2007 Using WordNet to disambiguate word senses for text classification. In: *Proceedings of the 7th International Conference on Computational Science*, Springer, pp. 781–789

[13] Miller G A, Beckwith R, Fellbaum C, Gross D and Miller K J 1990 WordNet: an on-line lexical database. *Int. J. Lexicogr.* 3(4): 235–244

[14] Miller G A 1993 WordNet: a lexical database. *Commun. ACM* 38(11): 39–41

[15] Cañas A J, Valerio A, Lalinde-Pulido J, Carvalho M and Arguedas M 2003 Using WordNet for word sense disambiguation to support concept map construction, string processing and information retrieval. In: *Proceedings of SPIRE 2003*, pp. 350–359

[16] Marine C and Dekai W U 2005 Word sense disambiguation vs. statistical machine translation. In: *Proceedings of the 43rd Annual Meeting of the ACL*, Ann Arbor, pp. 387–394

[17] Màrquez L, Escudero G, Martínez D and Rigau G Supervised corpus-based methods for WSD. In: *Word Sense Disambiguation. Text, Speech and Language Technology,* vol. 33, pp. 167–216

[18] Carpuat M and Wu D 2005 Evaluating the word sense disambiguation performance of statistical machine translation. In: *Proceedings of the Second International Joint Conference on Natural Language Processing*, Jeju, Korea, October

[19] Yee S C, Hwee T N and David C 2007 Word sense disambiguation improves statistical machine translation. In: *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic, pp. 33–40

[20] Mihalcea R and Moldovan D 2000 An iterative approach to word sense disambiguation. In: *Proceedings of FLAIRS 2000*, Orlando, FL, pp. 219–223

[21] Christopher S, Michael P O and John T 2003 Word sense disambiguation in information retrieval revisited. In: *Proceedings of SIGIR'03*, July 28–August 1, Toronto, Canada

[22] Sanderson M 1994 Word sense disambiguation and information retrieval. In: *SIGIR '94 Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Dublin, Ireland, July 03–06, pp. 142–151

[23] Zhi Z and Hwee Tou N 2012 Word sense disambiguation improves information retrieval. In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Jeju Island, Korea, vol. 1, pp. 273–282

[24] Tou Ng H 2011 Does word sense disambiguation improve information retrieval? In: *Proceedings of ESAIR'11*, ACM, October 28, Glasgow, Scotland, UK, pp. 17–18

[25] Jacques G, Gilles F, Saïd R and Karim E 2009 Analysis of word sense disambiguation-based information retrieval. In: *Proceedings of the 9th Workshop of the Cross-Language Evaluation Forum, CLEF 2008—Evaluating Systems for Multilingual and Multimodal Information Access*, Denmark, 17–19 September 2008. Berlin: Springer, pp. 146–154

[26] Banerjee S and Pedersen T 2002 An adapted Lesk algorithm for word sense disambiguation using WordNet. In: *Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics*, Mexico City

[27] Lesk M 1986 Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In: *Proceedings of SIGDOC*

[28] Soler S and Montoyo A 2002 A proposal for WSD using semantic similarity. In: Gelbukh A (Ed.) *Computational Linguistics and Intelligent Text Processing. Proceedings of CICLing 2002*. Lecture Notes in Computer Science, vol. 2276. Berlin–Heidelberg: Springer

[29] Mittal K and Jain A 2015 Word sense disambiguation method using semantic similarity measure and OWA operator. *ICTACT J. Soft Comput.* 05(02) (Special issue on Soft-Computing Theory, Application and Implications in Engineering and Technology)

[30] Patwardhan S, Banerjee S and Pedersen T 2003 Using measures of semantic relatedness for word sense disambiguation. In: *CICLing'03 Proceedings of the 4th International Conference on Computational Linguistics and Intelligent Text Processing*, Mexico City, February, pp. 241–257

[31] Ye P 2004 Selectional preference based verb sense disambiguation using WordNet. In: *Proceedings of the Australasian Language Technology Workshop*, December, Sydney, Australia, pp. 155–162

[32] Xuri T, Xiaohe C, Weiguang Q and Shiw Y 2010 Semi-supervised WSD in selectional preferences with semantic redundancy. In: *COLING 2010: Posters*, August, 2010, Beijing, China, pp. 1238–1246

[33] Diana M C and Carroll J Disambiguating nouns, verbs, and adjectives using automatically acquired selectional preferences. *Computat. Linguist.* 29(4): 639–654

[34] Patrick Y and Timothy B 2006 Verb sense disambiguation using selectional preferences extracted with a state-of-the-art semantic role labeler. In: *Proceedings of the 2006 Australasian Language Technology Workshop (ALTW2006)*, pp. 139–148

[35] Yarowsky D 2000 Hierarchical decision lists for word sense disambiguation. *Comput. Humanit.* 34(1–2): 179–186

[36] Parameswarappa S and Narayana V N 2013 Kannada word sense disambiguation using decision list. *Int. J. Emerg. Trends Technol. Comput. Sci.* 2(3): 272–278

[37] Palanati D P and Kolikipogu R 2013 Decision list algorithm for word sense disambiguation for Telugu natural language processing. I. *Int. J. Electron. Commun. Comput. Eng.* 4(6): 176–180

[38] Pedersen T In: Agirre E and Edmonds P (Eds.) *Word Sense Disambiguation: Algorithms And Applications*

[39] Shinnou H and Sasaki M 2003 Unsupervised learning of word sense disambiguation rules by estimating an optimum iteration number in the EM algorithm. In: *Proceedings of the Seventh CoNLL, held at HLT-NAACL 2003*, Edmonton, May–June 2003, pp. 41–48

[40] Boshra F, Al_Bayaty Z and Joshi S 2014 Sense identification for ambiguous word using decision list. *Int. J. Adv. Res. Sci. Eng.* 3(10): 109–115

[41] Singh R L, Ghosh K, Nongmeikapam K and Bandyopadhyay S 2014 A decision tree based word sense disambiguation system in Manipuri language. *Adv. Comput. Int. J.* 1.5(4): 17–22

[42] Park S, Zhang B and Kim Y T 2003 Word sense disambiguation by learning decision trees from unlabeled data. *Appl. Intell.* 19: 27–38

[43] Sarmah J and Sarma S 2016 Decision tree based supervised word sense disambiguation for Assamese. *Int. J. Comput. Appl.* 141(1): 42–48

[44] Le C and Shimazu A 2004 High WSD accuracy using Naive Bayesian classifier with rich features. In: *Proceedings of PACLIC 18*, December 8th–10th, 2004, Waseda University, Tokyo, pp. 105–114

[45] Escudero G, Màrquez L and Rigau G 2000 Naive Bayes and exemplar-based approaches to word sense disambiguation revisited. In: *Proceedings of the 14th European Conference on Artificial Intelligence, ECAI*, Berlin, Germany

[46] Aung N T T, Soe K M and Thein N L 2011 A word sense disambiguation system using Naïve Bayesian algorithm for Myanmar language. *Int. J. Sci. Eng. Res.* 2(9): 1–7

[47] Abraham A 2004 Meta learning evolutionary artificial neural networks. *Neurocomputing* 56: 1–38

[48] Azzini A and Tettamanzi A 2006 A neural evolutionary approach to financial modeling. In: *Proceedings of GECCO'06*, vol. 2. San Francisco, CA: Morgan Kaufmann, pp. 1605–1612

[49] Azzini A and Tettamanzi A 2006 A neural evolutionary classification method for brain-wave analysis. In: *Proceedings of EVOIASP'06*, pp. 500–504

[50] Yao X and Liu Y 1997 A new evolutionary system for evolving artificial neural networks. *IEEE Trans. Neural Netw.* 8(3): 694–713

[51] Ng H T 1997 Exemplar-based word sense disambiguation: some recent improvements. In: *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pp. 208–213

[52] Lee Y K, Ng H T and Chia T K 2004 Supervised word sense disambiguation with support vector machines and multiple knowledge sources. In: *Proceedings of SENSEVAL-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, Association for Computational Linguistics, Barcelona, Spain, July

[53] Buscaldi D, Rosso P, Pla F, Segarra E and Arnal ES 2006 Verb sense disambiguation using support vector machines: impact of WordNet-extracted features. In: Gelbukh A (Ed.) *Proceedings of CICLing 2006*, LNCS 3878, pp. 192–195

[54] Joshi M, Pedersen T and Maclin R 2005 A comparative study of support vector machines applied to the supervised word sense disambiguation problem in the medical domain. In: *Proceedings of the 2nd Indian International Conference on Artificial Intelligence (IICAI-05)*, December 20–22, 2005, Pune, India

[55] Brody S, Navigli R and Lapata M 2006 Ensemble methods for unsupervised WSD. In: *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, Sydney, pp. 97–104

[56] Escudero G, Màrquez L and Rigau G 2000 Boosting applied to word sense disambiguation. In: *Proceedings of the 12th European Conference on Machine Learning, ECML*, Barcelona, Catalonia

[57] Escudero V, Marquez L and Rigau G 2001 Using lazy boosting for word sense disambiguation. In: *Proceedings of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation Systems*, July, Toulouse, France, pp. 71–74

[58] Pal A R, Saha D, Naskar S and Dash N S 2015 Word sense disambiguation in Bengali: a lemmatized system increases the accuracy of the result. In: *Proceedings of the 2nd International Conference on Recent Trends in Information Systems (ReTIS)*, pp. 342–346