



Word Sense Disambiguation in Bangla Language Using Supervised Methodology with Necessary Modifications

Alok Ranjan Pal · Diganta Saha · Niladri Sekhar Dash · Antara Pal

Received: 10 July 2017 / Accepted: 11 May 2018 / Published online: 24 May 2018
© The Institution of Engineers (India) 2018

Abstract An attempt is made in this paper to report how a supervised methodology has been adopted for the task of word sense disambiguation in Bangla with necessary modifications. At the initial stage, the Naïve Bayes probabilistic model that has been adopted as a *baseline* method for sense classification, yields moderate result with 81% accuracy when applied on a database of 19 (nineteen) most frequently used Bangla ambiguous words. On experimental basis, the *baseline method* is modified with two extensions: (a) inclusion of lemmatization process into of the system, and (b) bootstrapping of the operational process. As a result, the level of accuracy of the method is slightly improved up to 84% accuracy, 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. The data sets that have been used for this experiment include the Bangla POS tagged corpus obtained from the Indian Languages Corpora Initiative, and the Bangla

WordNet, an online sense inventory developed at the Indian Statistical Institute, Kolkata. The paper also reports about the challenges and pitfalls of the work that have been closely observed and addressed to achieve expected level of accuracy.

Keywords Natural language processing · Word sense disambiguation · Naïve Bayes method · Lemmatization · Bootstrapping

Introduction

In every natural language there are so many words which carry different senses in different contexts of their use. These words are often recognized as ambiguous words and finding the exact contextual sense of an ambiguous word in a piece of text is known as called 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. Till date we have come across three major methodologies that are used to deal with this problem, namely, Supervised Method, Knowledge based Method and Unsupervised Method.

In Supervised Method [4, 6–24], sense disambiguation of words is performed with the help of previously created learning data 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 Method [25–35] depends on external knowledge-based resources like online semantic

A. R. Pal (✉)
Department of Computer Science and Engineering, College of Engineering and Management, Kolaghat, India
e-mail: chhaandasik@gmail.com

D. Saha
Department of Computer Science and Engineering, Jadavpur University, Kolkata, India
e-mail: neruda0101@yahoo.com

N. S. Dash
Linguistic Research Unit, Indian Statistical Institute, Kolkata, India
e-mail: nisedash@gmail.com

A. Pal
Department of Computer Science and Engineering, NIT, Durgapur, India
e-mail: antarapal22@gmail.com

dictionaries, thesauri, Machine readable dictionaries, etc. to obtain sense definitions of the lexical components.

In Unsupervised Method [34, 36, 37], the sense disambiguation happens in two phases. First, 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 assigns the sense to that new test data.

The present work is based on Naïve Bayes probabilistic model which is used as a baseline method for sense classification. This baseline method generates 81% accurate result when the algorithm tested on 900 instances of 19 ambiguous words. Next, two extensions are adopted to increase the level of accuracy: (a) incorporation of lemmatization in the system that generates 84% accuracy, and (b) operation of Bootstrapping on the system that produces 83% accuracy.

The organization of the paper is as follows: Sect. 2 presents a brief Survey in this research methodology; Proposed Approach is demonstrated in Sect. 3; Results and Discussion is presented in Sect. 4; in Sect. 5, Extensions on the Baseline Methodology is described in detail. The report is concluded with future scope in Sect. 6.

Survey

In case of the Supervised Method, 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 distribution calculated using the learning sets. Some commonly used approaches are deployed in this method, which are discussed below:

Decision List

In Decision List [35, 36] based approach, first, a set of rules are formed for a target word. Next, few 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.

Decision Tree

The Decision Tree [38–40] based approach frames the rules in the form of a tree structure 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.

Naïve Bayes

The Naïve Bayes [41–43] probabilistic model classifies the instances based on few parameters. These parameters calculate the probability distribution of a particular instance w.r.t. the different classifiers. The classifier, for which the probability value is the maximum for a test instance, categorizes the instance accordingly. The formula for the Naïve Bayes classification is as follows:

$$\begin{aligned}\hat{S} &= \underset{S_j \in \text{SenseD}(w)}{\text{ARGMAX}} P(S_i | f_1, \dots, f_m) \\ &= \underset{S_j \in \text{SenseD}(w)}{\text{ARGMAX}} \frac{P(f_1, \dots, f_m | S_i) P(S_i)}{P(f_1, \dots, f_m)}\end{aligned}$$

where ‘ S_i ’ represents different senses of the ambiguous word (w), the parameter ‘ f_j ’ represents the features of the word (w) in the context (S_i) and m is the number of features.

Neural Network

In Neural Network based approach [44–47], the artificial neurons act as the data processing units. The artificial neurons categorize the features into a number of non-overlapping sets. While designing a network using artificial neurons, these are arranged in different layers and the data is passed through these layers to reach the destination layer. In a network, words are treated as nodes and relations among the words are considered as links. In a network when data proceeds, only those links get activated where the two words at the two end points of an edge are semantically related.

Exemplar-Based Method

In Exemplar-Based [48] method, 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.

Support Vector Machine

In Support Vector Machine based [49–51] method, 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.

Ensemble Methods

In Ensemble Method based [52] approach, 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 [53, 54] etc.

Proposed Approach

The proposed approach adopts the Naïve Bayes (NB) probabilistic model as a baseline strategy. This model classifies the instances based on few predefined parameters.

Module 1: Training Module

Development of the training model depends on the following parameters:

- a. |V| which represents the number of vocabulary,
- b. P(c_i), to calculate the priori probability of each class,
- c. n_i, carries the total numbers of word frequencies of each class,
- d. P(w_i|c_i) which represents the conditional probability of a keyword in a given class.

The “zero frequency” problem is resolved using the Laplace Estimation in the following way:

$$P(w_i|c_i) = (\text{Number of occurrences of each word in a given class} + 1) / (n_i + |V|).$$

Module 2: Testing Module

A test data is classified with the help of “posterior” probability, P(c_i|W) w.r.t. each class using the following formula:

$$P(c_i|W) = P(c_i) \times \prod_{j=1}^{|V|} P(W_j|c_i)$$

The highest probability measure assigns a test data to a particular class.

Flow Chart of the Baseline Method

The baseline method can be represented through the following diagram (Fig. 1).

Results and Discussion

The following steps have been executed to run the system on the database:

Text Normalization

The texts stored in the TDIL Bangla corpus are non-normalized in nature. So, the very first task was to normalize

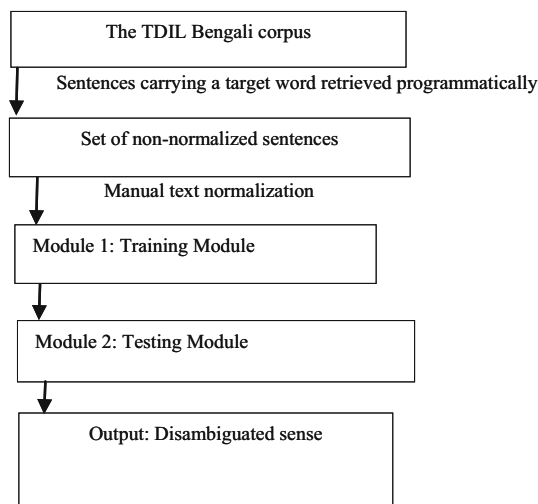


Fig. 1 Flow chart of the proposed baseline approach

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 Bangla font (*Vrinda* in this work), (d) considering all types of Bangla sentence termination symbols as note-of-exclamation, note-of-interrogation and *purnacched* (full stop) (“।”).

Removal of Non-functional Words

In NLP works there is not any specific rule or process for differentiating between functional word and non-functional words. Rather, it is more or less based on the nature of application of a NLP work. Although, in practical sense, all Bangla words are useful in some contexts or the other, while preparing the data sets for the present work, few Bangla words have been ignored to keep the number of words within a manageable length. After lemmatization, words except nouns, pronouns, adjectives, verbs and adverbs (in Bangla, adverbs are also treated as a kind of adjective) are considered functional words.

Selection of Ambiguous Word

Theoretically it is possible to assume that any Bangla word can appear in a text with certain level of ambiguity. People of computational linguistics like to use a few constraints from implementation perspective to select the ambiguous words. The Bangla text corpus used in this work consists of 35,89,220 inflected and non-inflected words, among which 199,245 words may be treated as distinct lexical units. These words are first 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 later.

Annotation of an Input Data

The sentences in the test data set are annotated in the following way:

<Sentence x> tag at the beginning of each sentence represents the sentence number in the paragraph and <wsd_id=y, pos=z> tag carries the ambiguous word number and Part-of-Speech of the target word in that particular sentence (Fig. 2).

Preparation of a Reference Output Data

The reference output files have been generated with the help of a standard Bangla dictionary (*Sansad Banglā Avidhān=Samsad Bangla Dictionary*) (Fig. 3). The reference files are used by the system to verify the system generated outputs using a separate program.

In the first phase of the work, the *baseline method* is applied on 900 sentences containing mostly used 19 Bangla ambiguous words.

Selection of Senses of the Ambiguous Words for Evaluation

After retrieving 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 Bangla words is so vast that it appeared as a real challenge to select a few senses from them for 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 inflected forms, whereas the on-line Bangla

WordNet sites only 14 (fourteen) distinct senses for the word. On the contrary, the TDIL Bangla text corpus provides only 4 (four) different senses of the word with some needful numbers of sentences. Taking all these variations into consideration the threshold value has been considered as 5 for the present work.

The following algorithm evaluates the multiple senses of an ambiguous word for the experiment:

Algorithm: Sense-Selection

Input: Sentences from a corpus containing an ambiguous word.

Output: Multiple senses of the ambiguous words.

Step 1: Sentences, classified based on contextual words.

Step 2: Misclassified sentences are rectified by an expert.

Step 3: Sense inventory is prepared for the ambiguous word based on *Sansad Bangla Abhidhan* and Bangla WordNet.

Step 4: Specific senses are tagged to the sentence classes from the sense inventory.

Step 5: Sense tagged classes are rearranged according to the decreasing number of sentences in it.

Step 6: Classes containing sentences more than a threshold value are selected.

Step 8: Senses associated with the classes are considered for evaluation.

The selected senses obtained by this algorithm are listed in Table 1.

Parameters for Evaluating the Performance

The performances of the algorithms have been measured using the conventional parameters: Precision, Recall, and F-Measure.

Precision (P) = (number of correctly evaluated instances according to human decision)/(total number of solved instances by the system).
Recall (R) = (number of correctly evaluated instances according to human decision)/(total number of data instances), and
F - Measure = $2 * P * R / (P + R)$.

Fig. 2 Partial view of a sample input file

<Sentence 1> মার্কিন যুক্তরাষ্ট্রে ফিল্ম তৈরির সঙ্গে সঙ্গেই এদেশে হোটেল ব্যবসাতেও <wsd_id=1, pos=noun> হাত </wsd> লাগিয়েছেন বিজয় অমৃত রাজ।<Sentence 2> কোথাও সহানুভূতি নেই সহযোগিতার <wsd_id=1, pos=noun> হাত </wsd> এতটুকু প্রসারিত নেই একটি ভাল কথা কারও মুখ থেকে শোনা যায় না।<Sentence 3> এই সব কয়টি দেশেরই জাতীয় মুক্তি ও সাম্রাজ্যবাদ বিরোধিতার আন্দোলনে ভারত একনিষ্ঠভাবে পাশে দাঁড়াইয়াছে সাহায্য ও সহযোগিতার <wsd_id=1, pos=noun> হাত </wsd> প্রসারিত করিয়াছে।<Sentence 4> সঠিক ও বাস্তব মূল্যায়নের অভাব খাপছাড়াভাবে

Fig. 3 Partial view of a reference output data

<Sentence 1> মার্কিন যুক্তরাষ্ট্রে ফিল্ম তৈরির সঙ্গে সঙ্গেই এদেশে হোটেল ব্যবসাতেও <wsd id=1, pos=noun, sense=abadaan> হাত </wsd> লাগিয়েছেন বিজয় অমৃত রাজ, <Sentence 2> কোথাও সহানুভূতি নেই সহযোগিতার <wsd id=1, pos=noun, sense=abadaan> হাত </wsd> এতটুকু প্রসারিত নেই একটি ভাল কথা কারও মুখ থেকে শোনা যায় না, <Sentence 3> এই সব কয়টি দেশেরই জাতীয় মুক্তি ও সাম্রাজ্যবাদ বিরোধিতার আন্দোলনে ভারত একনিষ্ঠভাবে পাশে দাঁড়াইয়াছে সাহায্য ও সহযোগিতার <wsd id=1, pos=noun, sense=abadaan> হাত </wsd> প্রসারিত করিয়াছে, <Sentence 4> সঠিক ও বাস্তব মূল্যায়নের অভাব খাপছাড়াভাবে কলোনি উন্নয়নের কাজে <wsd id=1, pos=noun, sense=abadaan> হাত </wsd> দেওয়া যথেষ্ট ব্যবস্থা না করে শিবির বন্ধ

Table 1 Selected senses of the ambiguous words

Word	Selected senses
āthā	mastak (head), chintā (thought), prānta (edge)
ghar	griha (home), sansār karā (to live family life), bansha (family)
mane	manan (mind), bodh haoyā (assuming), mane dharā (liking)
pā	pā (leg), padaksep (step), padārpan (to keep foot), eksange calā (move together), phense yāoyā (to be trapped)
tolā	uttolan karā (pick), utthāpan karā (propose), arpan karā (give), pratyāhār karā (withdraw), sristi karā (design), sangraha karā (collet)
jal	bāri (water), ashru (tear), jive jal (saliva), ghatanā prabāha (flow of event)
mānush	byakti (person), nar (homo sapiens), lālan pālan (nourish)
parā	parāshunā karā (study), patan (fall)
hāt	hasta (hand), abadān (contribution), hāt pātā (beg), hāt badal (exchaneg)
yog	yogfal (add), samparka (relation), yogdān karā (participate)
mukh	badan (face), mukh bibar (mouth), prānta (opening)
shabda	akshar (word), dhwane (sound)
din	din (day), deyā (give), pratidin (everyday), din kātāno (life-living)
kāj	kārya (work), kartabya (duty), hasta-shilpa (handicraft)
nām	nām (name), sunām (fame), nāmgān (chant)
samay	ksan (moment), kāl (in time), abasar (leisure time)
dhare	dhāran karā (hold), ākraman karā (attack), yābat (during)
kāchhe	kāchākāchi (near), prati (to), bicāre (according to)
niche	neecu (down), kam (less), parabarti ansha (next section)

Through the work, the systems evaluated all the test instances either correctly or wrongly which result the same Precision and the Recall value for each data.

Baseline Result

The typical Naïve Bayes algorithm has been developed as a baseline for this work. The algorithm has evaluated 19 mostly used Bangla ambiguous words with the same Precision and Recall value of 81% on an average (Table 2).

Table 2 Execution of the baseline model

Word	No. of sentences	Accuracy achieved (%)
ghar	50	84
mane	55	80
pā	50	82
tolā	30	82
jal	77	83
mānush	50	78
parā	50	86
hāt	20	80
yog	50	84
mukh	50	72
shabda	50	86
din	20	80
kāj	50	82
nām	50	86
samay	50	82
dhare	53	82
kāchhe	56	83
niche	51	74
māthā	30	83
Total	892	81.5

Extensions on the Baseline Methodology

To enhance the performance of the baseline methodology, the following two extensions have been adopted: (a) Lemmatization of inflected forms the whole system, and (b) Bootstrapping.

Lemmatization of the Whole System

Since Bangla is morphologically very strong, only lexical matching is not adequate enough for measuring the similarity of senses between the words. To overcome this bottleneck, the whole system has been operated on the lemmatized forms of words. The expansion of lexical coverage due to lemmatization generates situation where more number of lexical similarity are observed between the

Fig. 4 A sample lemmatized input data

<Sentence 1> মার্কিন/মার্কিন/noun যুক্তরাষ্ট্রে/রাষ্ট্র/noun ফিল্ম/ফিল্ম/noun তৈরির/তৈরি/adj
 সঙ্গে/সঙ্গে/adj সঙ্গেই/সঙ্গে/adj এদেশে/দেশ/noun হোটেল/হোটেল/noun ব্যবসাতেও/ব্যবসা/noun
 <wsd_id=1, pos=noun> হাত/হাত/noun </wsd> লাগিয়েছেন/লাগানো/verb বিজয়/বিজয়/noun
 অমৃত/অমৃত/noun রাজ/রাজ/noun<Sentence 2> কোথাও/কোথাও/prn সহানুভূতি/অনুভূতি/noun
 নেই/থাকা/verb সহযোগিতার/সহযোগিতা/noun <wsd_id=1, pos=noun> হাত/হাত/noun </wsd>
 এতটুকু/এত/prn প্রসারিত/প্রসারণ/noun নেই/থাকা/verb একটি/এক/adj ভাল/ভাল/adj
 কথা/কথা/noun কারও/কার/prn মুখ/মুখ/noun থেকে/থেকে/adj শোনা/শোনা/verb যায়/যাওয়া/verb
 না/না/abay<Sentence 3> এই/এ/prn সব/সব/adj কয়টি/কয়েক/adj দেশেরই/দেশ/noun |

Table 3 Performance of the algorithm on a regular data and its corresponding lemmatized form

Word	No. of sentences	Accuracy of result in non lemmatized system (%)	Accuracy of result in lemmatized system (%)
jal	77	83	86
samay	50	82	84
hāt	20	80	85
jog	50	84	85
mānuSh	50	78	78
din	20	80	85
ghar	50	84	84
māthā	30	83	86
dhare	53	82	82
shabda	50	86	86
tolā	30	82	86
fale	50	83	82
Total	892	82	84

instances which, eventually, leads the system to act in a far more robust manner to achieve higher level of accuracy. The lemmatization tool operated on the training sets, test data, and vocabulary (features) in a uniform manner without any selectional bias. However, since the tool could not produce accurate results for all the words, which is bound to happen due to complexities involved in the surface forms of many inflected Bangla words, manual intervention has been necessary for rectifying some of the errors in the eventual output database. A glimpse of the sample lemmatized input data is presented below (Fig. 4) where annotation of the text follows the same strategy as in the baseline method in addition to the words derived from lemmatization. Words are represented in the following format: “word-in-surface-level/stem-form/POS”.

This expansion approach uses the same standard output files used in the baseline experiment. Though the inputs have been prepared in lemmatized form, the outputs have been generated in surface level forms of the words to conduct a similar comparison with the baseline approach. In the following table (Table 3) the performance of the

algorithm on a regular data and its corresponding lemmatized form is presented.

It is observed that the overall accuracy has been increased due to the expansion of lexical coverage of the words. Since the size of the data sets taken for the experiment is quite smaller in number, at several occasions, the algorithm has returned the same accuracy. As mentioned earlier, the Precision and Recall both the values are 84% in this phase over a baseline accuracy of 82% on a same data set.

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 can produce a better accuracy in its next executions. A small manual intervention was mandatory in the phase also. Since the classification of a data set depends on the probability measures based on the training sets, the methodology requires a correctly populated training set for sense retrieval. Since the proposed model could not produce an absolute result in a particular execution, the misclassified instances have been further rectified by manual intervention to lead the system towards a right direction (Fig. 5).

In this phase two consecutive executions have been considered. In the first phase, the module has been tested on a selected set of data from the Bangla corpus. After the training sets are auto-incremented, a new set of data has been selected for the experiment for the second phase. The accuracy of the result in both the phases is presented in Table 4. The Precision and Recall values are same as 83% over a base line accuracy of 81.5%.

It is observed that extensions on the baseline methodology can produce a better result in most of the cases (Tables 3, 4). However, at a few cases, the accuracy level has slightly dropped. Through investigation it is observed that the accuracy of the system depends on a few predefined parameters such as, vast varieties in sentence representation of any particular sense, occurrence of same lexical entries in semantically dissimilar sentences, and many more.

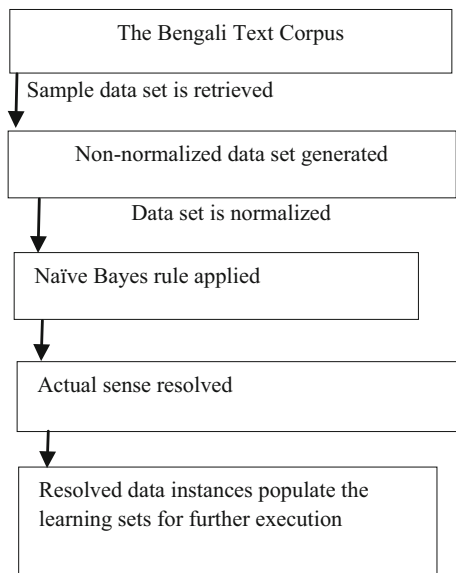


Fig. 5 Flowchart of the proposed bootstrapping method

Table 4 Result of bootstrapping method

Word	No. of sentences	Accuracy achieved in 1st execution (%)	Accuracy achieved in 2nd execution (%)
ghar	50	84	85
mane	55	80	84
pā	50	82	78
tolā	30	82	84
jal	77	83	86
mānush	50	78	70
parā	50	86	86
hāt	20	80	81
yog	50	84	85
much	50	72	79
shabda	50	86	85
din	20	80	82
kāj	50	82	83
nām	50	86	86
samay	50	82	85
dhre	53	82	84
kāchhe	56	83	83
neeche	51	74	80
māthā	30	83	84
Total	892	81.5	83

Conclusion and Future Scope

In this paper the work for Word Sense Disambiguation in Bangla language has been proposed using the Naïve Bayes algorithm as a baseline method supported with two extensions, namely, lemmatization and bootstrapping. The results obtained from this work, although not exact to our

expectation, may be accepted for the time being on the ground that this is the first attempt of this kind and this method may help us to devise new strategies for achieving our goals. In reality the complex linguistic natures of the South Asian languages like Hindi, Bangla, Tamil, Telugu, Punjabi, Malayalam and Marathi etc. usually put before us several challenges in the form of fonts, texts, morphological complexities, etc. due to which achieving even slight breakthrough in computation of these languages become a real challenge for many of us. At the same time the variation of senses of words, diversities in sentence structures, and complex formation of functional and nonfunctional words etc. demand additional attention for achieving better result from such experiments.

References

1. N. Ide, J. Véronis, Word sense disambiguation: the state of the art. *Comput. Linguist.* **24**(1), 1–40 (1998)
2. R. Florian, S. Cucerzan, C. Schafer, D. Yarowsky, Combining classifiers for word sense disambiguation. *Nat. Lang. Eng.* **8**(4), 327–341 (2002)
3. M.S. Nameh, M. Fakhrahmad, M.Z. Jahromi, A New approach to word sense disambiguation based on context similarity, in *Proceedings of the World Congress on Engineering*, vol. I (2011)
4. W. Xiaojie, Y. Matsumoto, Chinese word sense disambiguation by combining pseudo training data, in *Proceedings of The International Conference on Natural Language Processing and Knowledge Engineering* (2003), pp. 138–143
5. R. Navigli, Word sense disambiguation: a survey. *ACM. Comput. Surv.* **41**(2), 1–69 (2009)
6. M. Sanderson, 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 (Springer, New York, 1994), pp. 142–151
7. E. Agirre, P. Edmonds (eds.), *Word Sense Disambiguation, Algorithms and Applications*, Text Speech and Language Technology, vol 33 (Springer, Netherlands, 2007)
8. H. Seo, H. Chung, H. Rim, S.H. Myaeng, S. Kim, Unsupervised word sense disambiguation using WordNet relatives. *Comput. Speech Lang.* **18**(3), 253–273 (2004)
9. G.A. Miller, R. Beckwith, C. Fellbaum, D. Gross, K. Miller, WordNet: an on-line lexical database. *Int. J. Lexicogr* **3**, 235–244 (1990)
10. S.G. Kolte, S.G. Bhirud, Word sense disambiguation using WordNet domains, in *1st International Conference on Digital Object Identifier* (2008), pp. 1187–1191
11. Y. Liu, P. Scheuermann, X. Li, X. Zhu, Using WordNet to disambiguate word senses for text classification, in *Proceedings of the 7th International Conference on Computational Science* (Springer, Berlin, 2007), pp. 781–789
12. G.A. Miller, R. Beckwith, C. Fellbaum, D. Gross, K.J. Miller, WordNet an on-line lexical database. *Int. J. Lexicogr.* **3**(4), 235–244 (1990)
13. G.A. Miller, WordNet: a lexical database. *Commun. ACM* **38** (11), 39–41 (1993)
14. A.J. Cañas, A. Valerio, J. Lalinde-Pulido, M. Carvalho, M. Arguedas, Using WordNet for Word Sense Disambiguation to

- Support Concept Map Construction. In: *String Processing and Information Retrieval*, eds. by M.A. Nascimento, E.S. de Moura, A.L. Oliveira. SPIRE 2003. Lecture Notes in Computer Science, vol 2857 (Springer, Berlin, Heidelberg, 2003) pp. 350–359
15. C. Marine, W.U. Dekai, Word sense disambiguation vs. statistical machine translation, in *Proceedings of the 43rd Annual Meeting of the ACL* (Ann Arbor, 2005), pp. 387–394
 16. <http://www.ling.gu.se/sl/Undervisning/StatMet11/wsd-mt.pdf>. 14 May 2015
 17. <http://nlp.cs.nyu.edu/sk-symposium/note/P-28.pdf>. 14 May 2015
 18. S.C. Yee, T.N. Hwee, C. David, Word sense disambiguation improves statistical machine translation, in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics* (Prague, 2007), pp. 33–40
 19. R. Mihalcea, D. Moldovan, An iterative approach to word sense disambiguation, in *Proceedings of Flairs 2000* (Orlando, FL, 2000), pp. 219–223
 20. S. Christopher, P.O. Michael, T. John, *Word Sense Disambiguation in Information Retrieval Revisited, SIGIR'03, July 28–Aug 1, 2003* (Canada, Toronto, 2003)
 21. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.65.6828&rep=rep1&type=pdf>. 14 May 2015
 22. <http://www.aclweb.org/anthology/P12-1029>. 14 May 2015
 23. <https://www.comp.nus.edu.sg/nght/pubs/esair11.pdf>. 14 May 2015
 24. <http://cui.unige.ch/isi/reports/2008/CLEF2008-LNCS.pdf>. 14 May 2015
 25. S. Banerjee, T. Pedersen, 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, 2002)
 26. M. Lesk, Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone, in *Proceedings of SIGDOC* (1986)
 27. <http://www.dlsi.ua.es/projectes/srim/publicaciones/CICLing-2002.pdf>. 14 May 2015
 28. K. Mittal, A. Jain, Word sense disambiguation method using semantic similarity measures and owa operator. *ICTACT J. Soft Comput, Special .Issue .Soft. Comput. Theor. Appl. Implications. Eng. Technol.* **05**(02), 896–904 (2015)
 29. <http://www.d.umn.edu/tpederse/Pubs/cicling2003-3.pdf>. 14 May 2015
 30. <http://www.aclweb.org/anthology/U04-1021>. 14 May 2015
 31. <http://www.aclweb.org/anthology/C10-2142>. 14 May 2015
 32. M.C. Diana, J. Carroll, Disambiguating nouns, verbs, and adjectives using automatically acquired selectional preferences. *Comput. Linguist.* **29**(4), 639–654 (2003)
 33. Y. Patrick, B. Timothy, 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)* (2006), pp. 139–148
 34. <http://link.springer.com/article/10.1023/A%3A1002674829964#page-1>. 14 May 2015
 35. S. Parameswarappa, V.N. Narayana, Kannada Word sense disambiguation using decision list. *Inter. J. Emerg. Trends. Technol. Comput. Sci.* **2**(3), 272–278 (2013)
 36. http://www.academia.edu/5135515/Decision_List_Algorithm_for_WSD_for_Telugu_NLP. Accessed 10 Mar 2015
 37. T. Pedersen, in *Unsupervised Corpus-Based Methods for WSD*, eds. by E. Agirre, P. Edmonds. *Word Sense Disambiguation. Text, Speech and Language Technology*, vol 33. (Springer, Dordrecht, 2007), pp. 133–166
 38. R.L. Singh, K. Ghosh, K. Nongmeikapam, S. Bandyopadhyay, A decision tree based word sense disambiguation system in Manipuri language. *ACIJ* **5**(4), 17–22 (2014)
 39. http://wing.comp.nus.edu.sg/publications/theses/2011/low_wee_urop.pdf. 14 May 2015
 40. <http://www.d.umn.edu/tpederse/Pubs/naacl01.pdf>. 14 May 2015
 41. C. Le, A. Shimazu, High WSD accuracy using Naive Bayesian classifier with rich features, in *PACLIC 18*, Dec 8th–10th, 2004 (Waseda University, Tokyo, 2004), pp. 105–114
 42. <http://www.cs.upc.edu/escudero/wsd/00-ecai.pdf>. 14 May 2015
 43. N.T.T. Aung, K.M. Soe, N.L. Thein, A word sense disambiguation system using Naive Bayesian algorithm for Myanmar Language. *Int. J. Sci. Eng. Res.* **2**(9), 1–7 (2011)
 44. <http://crema.di.unimi.it/pereira/his2008.pdf>. 14 May 2015
 45. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.13.9418&rep=rep1&type=pdf>. 14 May 2015
 46. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.154.3476&rep=rep1&type=pdf>. 14 May 2015
 47. <http://www.aclweb.org/anthology/W02-1606>. 14 May 2015
 48. <http://www.aclweb.org/anthology/W97-0323>. 14 May 2015
 49. <https://www.comp.nus.edu.sg/nght/pubs/se3.pdf>. 14 May 2015
 50. D. Buscaldi, P. Rosso, F. Pla, E. Segarra, E.S. Arnal, *Verb Sense Disambiguation Using Support Vector Machines: Impact of WordNet-Extracted Features*, ed. by A. Gelbukh. *CICLing 2006, LNCS 3878* (2006), pp. 192–195
 51. <http://www.cs.cmu.edu/maheshj/pubs/joshipedersenmaclin.iicai2005.pdf>. 14 May 2015
 52. S. Brody, R. Navigli, M. Lapata, Ensemble methods for unsupervised WSD, in *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL* (Sydney, 2006), pp. 97–104
 53. <http://arxiv.org/pdf/cs/0007010.pdf>. 14 May 2015
 54. <http://www.aclweb.org/anthology/S01-1017>. 14 May 2015