A Survey of Word-sense Disambiguation Effective Techniques and Methods for Indian Languages

Shallu
University Institute of Engineering & Technology, Panjab University, Chandigarh, India
Email: shallu.2.chd@gmail.com

Vishal Gupta
University Institute of Engineering & Technology, Panjab University, Chandigarh, India
Email: vishal@pu.ac.in

***Abstract*—Word Sense Disambiguation is a challenging technique in Natural Language Processing. There are some words in the natural languages which can cause ambiguity about the sense of the word.WSD identifies the correct sense of the word in a sentence or a document. The paper summarizes about the history of WSD. We have discussed about the knowledge - based and machine learning – based approaches for WSD. Various supervised learning and unsupervised learning techniques have been discussed. WSD is mainly used in Information Retrieval (IR), Information Extraction (IE), Machine Translation (MT), Content Analysis, Word Processing, Lexicography and Semantic Web. Finally, we have discussed about WSD for Indian languages (Hindi, Malayalam, and Kannada) and other languages (Chinese, Mongolian, Polish, Turkish, English, Myanmar, Arabic, Nepali, Persian, Dutch, and Italian).**

***Index Terms*— Word Sense Disambiguation (WSD), Natural Language Processing (NLP), supervised, unsupervised, knowledge, information retrieval, information extraction, machine translation, context, ambiguity, polysemous words.**

I. INTRODUCTION

There are words in Natural languages which have different meaning for different context but they are spelled same. Those words are called polysemous words. Word sense disambiguation (WSD) is the solution to the problem. Word Sense Disambiguation [1] is a task of automatically assigning a correct sense to the words which are polysemous in a particular context.

Many Natural languages like English, Hindi, Punjabi, French, Chinese, etc. are the languages which have some words whose meaning are different for same spelling in the different context. In English, Words likes Bark, Lie, book, etc. can be considered example of polysemous words. Human beings are blessed with the learning power. They can easily find out what is the correct meaning of a word in a context. But for computer it is a difficult task. So, we need to develop an automatic system which can perform like humans do i.e. the system which can find out the correct meaning of the word in particular context. Context is the text or words which are surrounding to the ambiguous word. Using the context, human can easily sense the correct meaning of the word in that context. So we also need the computer to follow some rules using which the system can evaluate the absolute meaning out of multiple meanings of the word.

If we consider a text T a sequence of words i.e. w_1, w_2, $w_{3\ldots\ldots}w_n$. Then, WSD is a task to assign the correct sense for all or some words in the text T.

Two main approaches which are used to WSD are Deep approaches and shallow approaches. Deep approaches uses some kind of knowledge related to the word and shallow approaches see the context in which the word has been used [2]. The other approaches to Word sense Disambiguation are knowledge-based approach, machine learning approach.

The conceptual Model [15] for Word Sense Disambiguation is given below:

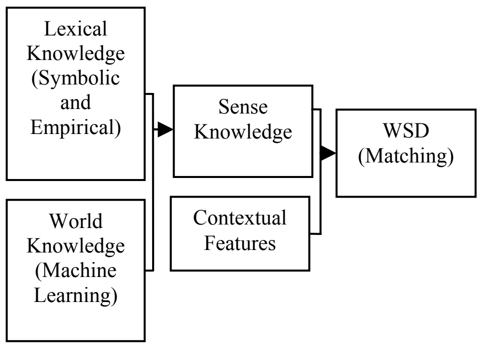

Figure 1 Conceptual Model of WSD [15]

Now, there are so many methods to assign senses, but how to measure which method provide good performance. So, the performance of the WSD can be measured by Precision and recall. Precision is defined as the proportion of correctly identifying senses of those identified, while recall is the proportion of correctly identified senses of total senses.

doi:10.4304/jetwi.5.4.354-360

It is an important and challenging technique for natural language processing (NLP). Many real world applications like machine translation (MT), semantic annotation (SA), semantic mapping (SM), and ontology learning (OL) uses WSD. Information retrieval (IR), information extraction (IE), and speech recognition (SR) are some of the applications in which WSD is used to improve the performance.

The remainder of the paper is organized as follows: in section 2, we mention various approaches for WSD, while in section 3 we present the WSD algorithms for making a word sense disambiguation system. Section 4 covers the applications where WSD is used and in Section 5, we will discuss about WSD for various Indian languages.

II. WSD Approaches

There are two approaches that are followed for Word Sense Disambiguation (WSD): Knowledge Based approach and Machine-Learning Based approach. In Knowledge based approach, it requires external lexical resources like Word Net, dictionary, thesaurus etc. In Machine learning-based approach, systems are trained to perform the task of word sense disambiguation. These two approaches are briefly discussed below

A. Machine Learning Based Approach

It adapts to new circumstances, detects and extrapolates patterns. In machine learning approach, the systems are trained to perform the task of WSD. A classifier is used to learn features and assigns senses to unseen examples. In these approaches, the initial input is the word to be disambiguated called target word, and the text in which it is embedded, called as context. Part-of-Speech tagging is used for processing, in which fixed set of features are extracted which are relevant to the task of learning called linguistic features. These linguistic features can be classified in two classes: collocation features and co-occurrence features. Collocation conceals the information about words that are located to left or right of target word at specific positions. Co-occurrence features contain the data or information about neighboring words. In this approach features are themselves served by the words. The value of feature is the number of times the word occurs in the region surrounding the target word. The region is often a fixed window with target word as center. Three types of techniques of machine learning based approaches are: supervised techniques, unsupervised techniques, and semi-supervised techniques.

Supervised Techniques: The learning here perform in supervision. Let us take the example of the learning process of a small child. The child doesn't know how to read/write. He/she is being taught by the parents at home and then by their teachers in school. The children are trained and modules to recognize the alphabets, numerals, etc. Their each and every action is supervised by the teacher. Actually, a child works on the basis of the output that he/she has to produce. Similarly, a word sense disambiguation system is learned from a representative set of labeled instances drawn from same distribution as test set to be used. Input instances to these approaches are feature encoded along with their appropriate labels. The output of the system is a classifier system capable of assigning labels to new feature encoded inputs. System is informed precisely about what should be emitted as output. In supervised learning, it is assumed that the correct (target) output values are known for each Input. So, actual output is compared with the target output, if there is a difference, an error signal should be generated by the system. This error signal helps the system to learn and reach to the desired or target output.

Unsupervised Technique: In unsupervised learning technique, no supervision is provided. Let us consider an example of a tadpole. Learning is done by itself i.e. child fish learn to swim without any supervision. It is not taught by anyone. Thus its leaning process is independent and not supervised by a teacher. Unsupervised approaches to word sense disambiguation eschew the use of sense tagged data of any kind during the training. In this technique, feature vector representations of unlabeled instances are taken as input and are then grouped into clusters according to a similarity metric. These clusters are then labeled by hand with known word senses. Main disadvantage is that senses are not well defined.

Semi-Supervised Techniques: In semi-supervised learning techniques, the information is present like in supervised but might be less information is given. Here only critic information is available, not the exact information. For example, the system may tell that only particular about of target output is correct and so. The semi-supervised or minimally supervised methods are gaining popularity because of their ability to get by with only a small amount of annotated reference data while often outperforming totally unsupervised methods on large data sets. There are a host of diverse methods and approaches, which learn important characteristics from auxiliary data and cluster or annotate data using the acquired information.

B. Dictionary Based Approach

In this style of approach the dictionary provides both the means of constructing a sense tagger and target senses to be used. An attempt to perform large scale disambiguation has lead to the use of Machine Readable Dictionaries (MRD). In this approach, all the senses of a word that need to be disambiguated are retrieved from the dictionary. These senses are then compared to the dictionary definitions of all the remaining words in context. The sense with highest overlap with these context words is chosen as the correct sense.

For example: consider the phrase 'pine cone' for selecting the correct sense of word cone, following are the definitions for pine and cone:

Pine: kinds of evergreen tree with needle-shaped leaves or waste away through sorrow or illness

Cone: solid body which narrows to a point or something of this shape whether solid or hollow or fruit of certain evergreen trees

In this example, Lesk's [11] method would select cone as the correct sense since two of the words in its entry, evergreen and tree, overlap with words in the entry for pine.

A major drawback of Dictionary based approaches is the problem of scaling.

III. WSD ALGORITHMS

A. HyperLex

The HyperLex algorithm presented in [14] is entirely corpus-based. Author has used the co-occurrence graphs. All-pair of words in the context are built in the form of co-occurrence graphs. It is a dictionary free method. The nodes in the graph are the words that co-occur with the target word. An edge is used to connect two nodes which concurrent to each other. It uses the properties of small world graph, and has the highly connected components (called hubs) in the graph. These hubs represent the senses produced by the system. These hubs identify the main word used i.e. it identifies the senses of the target word, and is used to perform word sense disambiguation.

In this, first author build the co-occurrence graph using the senses of the target word. The author considers only noun and adjectives. Verbs were also considered by the author but ended up because it was causing a notable degradation in performance. Paragraph is filtered and only nouns and adjectives are considered. All the verbs, prepositions, determiners and stop words are removed from the paragraph. Then a co-occurrence matrix from this filtered set of contexts was generated. Two words appearing in the same paragraph are called co-occur words. The HyperLex for the example used by author [14] is shown below in Fig 2:

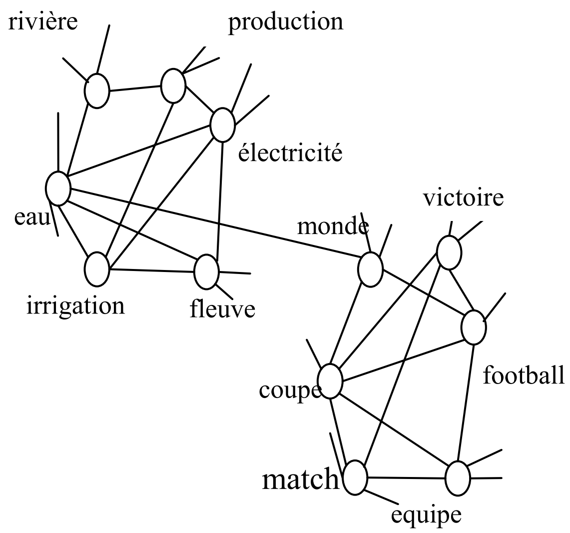

Figure 2 Graph of the co-occurrence of the French word 'barrage' [14]

After this, weights are given to the edges connecting two nodes. The co-occurrence networks are scale-free, so they contain a small number of highly connected hubs and a large number of weakly connected nodes [14]. Co-occurrence graph detects the different uses of a word and thus it amounts to isolate the high-density components. Every high-density component, one of the nodes has a higher degree than the others which is called the root hub of the component. All the root nodes are identified iteratively. For the root node, the node has to have (1) at least 6 specific neighbors (this threshold was determined experimentally), and (2) a weighted clustering coefficient large enough for it to actually be a root hub of a bundle [14]. Then a minimum spanning tree (or MST) is computed over the graph by taking the target word as the root and making its first level having the previously identified root hubs. The complexity of the graph mentioned by author [14] is O (E log (E)), where E is the number of edges in the graph. This MST is then used to construct a disambiguation system, which will tag the target word occurrences in the corpus. Each node v in the tree is assigned a score vector s with dimensions as there exist for the components. HyperLex given by the author provides a tool for domain and lexicon navigation. The results of the HyperLex algorithm were evaluated on the Web page corpus [14]. The best 25 contexts were checked for each of the 50 uses which include 1245 contexts in all. The overall precision obtained was 95.5%.

B. Extended Word Net

In the Lesk algorithm [5], word to disambiguate is given, the dictionary definition or gloss of each of its senses is compared to the glosses of every other word in the phrase. A word is assigned that sense whose gloss shares the largest number of words in common with the glosses of the other words. The algorithm begins a new for each word and does not utilize the senses it previously assigned.

A version of Lesk algorithm in combination with WordNet has been reported for achieving good word sense disambiguation results [13]. In their work, different types of relationships in WordNet have been experimented with. It showed that the best results are obtained when concatenating the descriptions of word senses with the glosses of its first and second-levels hypernyms.

This algorithm is used by Naskar and Bandyopadhyay [7], in which they have used the Word Net lexical database, because it contains different types of relationships between words. They proposed a global approach instead of local approach where all the words in the context window are simultaneously disambiguated in a bid to get the best combination of senses for all the words in the window instead of only the target word. The Lesk algorithm only work for short phrases. But the algorithm proposed by [7] takes the entire sentence under consideration.

The gloss bag is constructed for every sense of every word in the sentence. The gloss-bag is constructed from the POS and sense tagged glosses of synsets, obtained from the Extended Word Net. Once, the gloss-bag creation process is over, the comparison process starts. Each word (say W_i) in the context is compared with each word in the gloss-bag for every sense (say S_k) of every other word (say W_j) in the context. If a match is found, they are checked further for part-of-speech match. If the words match in part-of speech as well, a score is assigned to both the words: the word being matched (W_i) and the word whose gloss-bag contains the match (W_j). This matching event indicates mutual confidence towards each other, so both words are rewarded for this event. Two two-dimensional vectors are maintained: sense_vote for the word in context, and sense_score for the word in gloss-bag. Once all the comparisons have been made, sense_vote value is added with the sense_score linearly value for each sense of every word to arrive at the combination score for this word-sense pair.

Finally, for any word in the context, the value of sense index that maximizes this sum is declared the assigned sense for this particular word.

Knowledge base used by Naskar and Bandyopadhyay [7] was first 10 Semcor2.0 files. Another approach of knowledge based Disambiguation is using Word Net domains which is used for disambiguate nouns. It follows the unsupervised approach to word sense disambiguation [8] [17]. Domain is defined as set of words which contain the words with the semantic relation. This algorithm use 3 bags for solving ambiguity. Bag 3 contains the target word which we need to disambiguate and bag 3 is compared with bag 1 and bag 2. First, Domain of the word is interpreted and then the sense in that domain is the sense of the target word in bag 3. Precision of the algorithm was 85.9%, and the recall was calculated 62.1%.

C. Improved Unsupervised Learning Probabilistic Model

The purposed algorithm by the authors is a probabilistic model. The probabilistic models have parametric form and parameter estimation [15]**.** It shows an effect of one contextual feature over other contextual features and also, the effect of one contextual feature over the sense of an ambiguous word. The authors have considered the Naïve Bayes form for this purposed model. The posterior probability function, $p(S|F_1, F_2, ..., F_n)$, defined by Bayes Rule given by [15] is:

$$p(S|F_1, F_2, ..., F_n) = p(F1,F2,\ldots,Fn,S)/p(F1,F2,\ldots,Fn) = p(S)\times\prod i=1 \text{ to } n\ p(Fi \mid S))/(\sum s\ p(F1,F2,\ldots,Fn,S)$$

In this algorithm, first the word net is used. The word net will first annotate the senses of words that have single semantic item. Second step of this algorithm focuses on the part-of-speech ambiguity in which it will remove the ambiguity prior to sense disambiguation. After that, it will check for the words which are ambiguous and those words which are required for disambiguation. In this word net will define all the senses related to ambiguous words. Feature selection is the next important step in which features are selected using the Z-test. It will remove the noise in the disambiguation. Feature selection is used to increase the accuracy of WSD. In the result of this, the efficiency of WSD will also be improved.

In the proposed model [15], if *w* represents an ambiguous word and *wj* represents a contextual word, then their mutual information, *I* (*w, wj)*, is defined as:

$$I (w, wj) = \log2\ p(w,wj)/(p(w)p(wj)$$

After this, authors have estimated the initial parameters values. This algorithm proposes a statistical learning algorithm which will estimate initial parameter values of the model from raw untagged text because it is an unsupervised learning method and unsupervised learning is done strictly based on information obtained from raw untagged text [15]. The Expectation Maximization (EM) algorithm or Gibbs Sampling can be used to estimate the parameters of the probabilistic model [15].

D. Genetic Algorithm for WSD

The genetic algorithm for WSD is provided for the Arabian language due to writing structure [26]. The authors think the genetic algorithm is effective because it is very helpful in solving many NP hard optimization problems. Fig 3 below show the GAWSD prototype purposed by [26]:

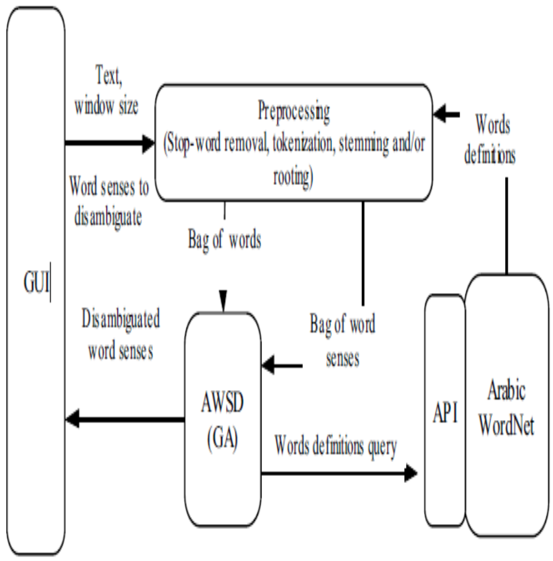

Figure 3 GAWSD prototype [26]

In this algorithm, a text T is passed through the preprocessing phase in which tokenization, stop-word removal, stemming and rooting is done. In preprocessing phase, first tokenization is done to split the text into words. After tokenization, authors have done the stop-word removal to filter out the stop words which are not important words in the text such as prepositions and articles, etc. after removing the stop words authors have performed the stemming on the remaining tokens. In stemming, it will remove the prefixes n suffixes from the word. After stemming, last step is rooting. Rooting will reduce the words to their root. Authors have used Khoja's Stemmer for rooting. The senses of each word are retrieved from Arabic Word net (AWN) as word definitions which are reduced in turn to bags of words. AWSD (GA) is used to find the most appropriate mapping from words to senses retrieved from AWN in the context T. Authors have shown that GA performs better than Naïve Bayes algorithm.

IV. APPLICATIONS

A. Information Extraction (IE)

Information Extraction is used for accurate analysis of text. Tasks like named-entity recognition (NER), acronym expansion (e.g., MP as Member of Parliament or military police), etc., can all be cast as disambiguation problems, although this is still a relatively new area. Another task is metonymy task in which systems are required to associate the appropriate metonymy with target named entities. For example, For instance, in the sentence *the BMW slowed down*, *BMW* is a car company, but here we refer to a specific car instance produced by BMW. Similarly, the Web People Search task [10] required systems to disambiguate people names occurring in Web documents, that is, to determine the occurrence of specific instances of people within texts.

B. Information Retrieval (IR)

Use either SI (MKS) or CGS as primary units. (SI units are encouraged.) If your native language is not English, try to get a native English-speaking colleague to proofread your paper. Do not add page numbers.

C. Machine Translation (MT)

WSD is important for Machine translations. There are words in one language which need to sense so that it could be translated to other language. There are some words which appear same in both language but they have different meaning. Machine translation helps in better understanding of source language and generation of sentences in target language. It also affects lexical choice depending upon the usage context.

D. Text Processing

Word Sense Disambiguation can also be used in Text to Speech translation, i.e., when words need to be pronounced in more than one way depending on their meaning. For example: “lead” can be “in front of” or “type of metal”.

E. Speech Processing and Part of Speech tagging

Speech recognition, i.e., when processing homophones words which are spelled differently but pronounced the same way. For example: “base” and “bass” or “sealing” and “ceiling”.

V. WSD FOR INDIAN LANGUAGES

Various works on WSD can be found in English and other European languages but, less amount of works in Indian languages. Various Indian Languages in which work has done are Manipuri, Tamil, Kannada, Hindi, Malyalam etc.

A. Manipuri

Manipuri is a Tibeto-Burman language, spoken in the valley of Manipur, a North-Eastern state of India. Due to the geographic location, differences in syntactic and semantic structures are noted from other Indian languages. Richard Singh and K. Ghosh[18] has recently given a proposed architecture for Manipuri Language in 2013. No work was done before this. The work presented in this paper is performed at KIIT University. The System performs WSD in two phases: training phase and testing phase. The suggested architecture to develop the Manipuri word sense disambiguation system contains five building blocks:(i) preprocessing, (ii) feature selection and generation and (iii) training, (iv) testing and (v) performance evaluation.

Raw Data is processed in the order to get the features which can be used for training and testing data efficiently. In feature Selection, a total of 6 features are taken to build feature:(i) the focus word for which the sense is to be derived,(ii) the normalized position of the word in the sentence,(iii) the previous word,(iv) the previous-to-previous word,(v) the next word,(vi) the next to next word. A 5-gram window is formed using the pair of the focus word and its context words which forms the context information. A focus word, based on the context may have different senses. Hence, in order to disambiguate the sense of the focused word, the contextual information is very much necessary and helps in predicting the correct one.

In the current study positional feature is suggested because of the lack of other relevant morphological features. As the syntactic and semantic structures of a sentence remain mostly similar for a particular language, this feature contains probable morphological information.

To generate the final input feature vector, from the database mentioned above mentioned six features are collected automatically by using the six above mentioned features and the output sense of the focus word, development of final feature vector takes place. By deriving manually the sense of the focus word, seven entries will be feed to the classifier finally. The classifier will be trained using a specified training algorithm.

During the testing, training algorithm used will be used to predict and compare the features for the test case. For predicting the sense for a test word, trained data is used and the corresponding features are generated and compared. The output generated will be tested for the accuracy and if the focus word is not found then it will be added in the training set. The predictions are later compared with the correct sense tags to perform evaluation of the current system.

B. Malyalam

Malayalam is a Dravidian language used predominantly in the state of Kerala, in southern India. It is one of the 22 official languages of India, and it is used by around 36 million people. [20] has given the first attempt for an automatic WSD in Malayalam. The author used the knowledge based approach. One approach used is based on a hand devised knowledge source and the other is using the concept of conceptual density, by using Malayalam Word Net as the lexical resource. The author has used the Lesk and Walker algorithm. In this algorithm, the author has collect all of the words from the context of a word 'w', which needs to be disambiguated and suppose this collection as 'C'. For each sense of 'w', collect the bag of words from the Knowledge source. Let it be 'B'. Measure the overlap between ‘C’ and 'B'. A score of 1 will be added to that sense if any overlap is there. Highest score sense will be selected as the winner.

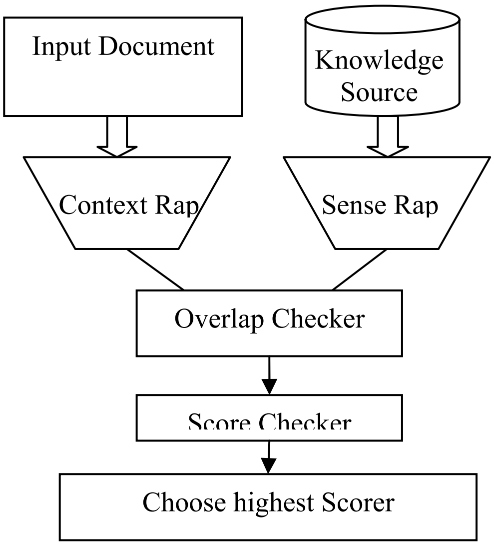

Figure 4 System Design based on Lesk and Walkers from [20]

The Second method is Conceptual density based Algorithm Design semantic relatedness between the words is taken into consideration. Semantic Relatedness can be measure in many ways. 3 metrics can be considered for measuring the semantic similarity of words using word net: Path, Depth and Information content.

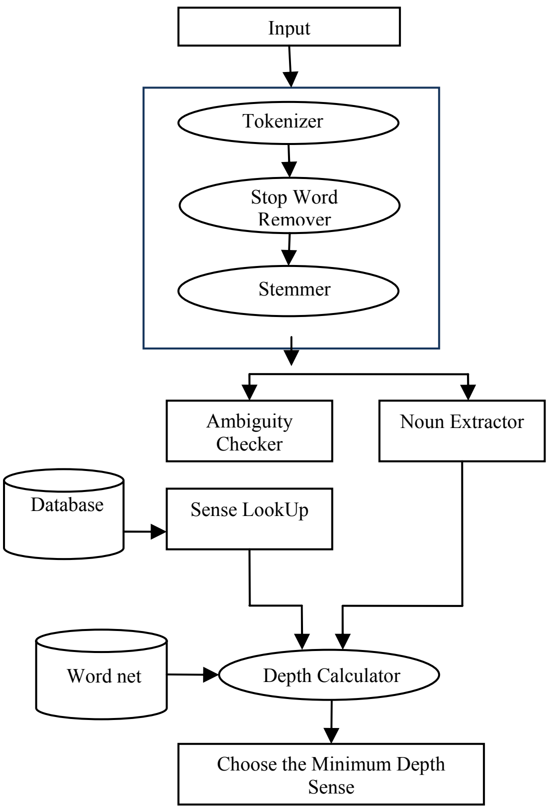

Figure 5 System Design using conceptual density from [20]

In this Algorithm depth is taken as the measurement. For each sentence: Tokenize the sentence, Remove the stop words, Perform stemming, Check for ambiguous words, If ambiguous word occurs, shift that word into one document and sense lookup is performed. Extract the nouns from the sentence and save it as a document.

For each sense in the sense lookup: Calculate the depth with each noun. If there are multiple nouns, depth of each will be added and taken as depth. The sense which results in lower Depth (highest conceptual density)is selected as the correct sense. Fig 5 is showing the system design using conceptual density given by [20].

C. Punjabi

The Punjabi language is morphologically rich. Rakesh and Ravinder [22] have given the WSD algorithm for removing ambiguity from the text document.WSD algorithm used by authors is Modified Lesk’s Algorithm. There are two hypothesis that underly this approach. The first is that words appears together in a sentence can be disambiguated by assigning to them the senses that are most closely related to their neighboring words. The second hypothesis is that related senses can be identified by finding overlapping words in their definitions

REFERENCES

[1] Christopher D. Manning, and Hinrich Schutze “Foundations of Statistical Natural Language Processing,” MIT Press, Cambridge, Massachusetts London, England 1999

[2] Wikipedia: http://en.wikipedia.org/wiki/Word-sense_disambiguation

[3] E. Agirre, and Philip Edmonds, “Word Sense Disambiguation: Algorithms and Applications (Text, Speech and Language Technology),” Springer-Verlag New York, Inc. Secaucus, NJ, USA 2006

[4] Roberto Navigli “Word Sense Disambiguation: A Survey,” Vol. 41, Universita di Roma La Sapienza, ACM Computing Surveys, 2009

[5] M. Lesk “Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from a ice cream cone,” In Proceedings of SIGDOC ’86, 1986

[6] Daniel Jurafsky and James H. Martin “An Introduction to Natural Language processing, Computational Linguistics, and Speech Recognition,” Pearson Education, 2008

[7] S.K. Naskar and S. Bandyopadhyay “word sense disambiguation using extended Word Net,” In proceedings of ICCTA’07, 2007

[8] S. G. Kolte and S. G. Bhirud “Word Sense Disambiguation using Word Net Domains”. In Proceedings of ICETET’08, 2008

[9] R. Krovetz and W. Bruce Croft “Lexical ambiguity and information retrieval,” Information System, vol. 10, pp. 115-141, 1992

[10] J. Artiles, J. Gonzalo and S. Sekine “The Semeval-2007 WEPS evaluation: Establishing a benchmark for the Web people search task,” In Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval, Prague, Czech Republic). pp. 64-69, 2007

[11] Ravi Sinha and Rada Mihalcea "Unsupervised Graph-based Word Sense Disambiguation Using Measures of Word Semantic Similarity," In International Conference on Semantic Computing, 2007

[12] G. Ramakrishnan, B. Prithviraj and P. Bhattacharyya "A Gloss Centered Algorithm for Word Sense Disambiguation," In Proceedings of the ACLSENSEVAL'04, Barcelona, Spain, pp. 217-221, 2004

[13] Jean Veronis "Hyperlex lexical cartography for information retrieval," Computer Speech & Language. Vol. 18, pp. 223-252, 2004

[14] Xu Li, Xiuyan Zhao, Fenglong Fan and Bai Liu "An Improved Unsupervised Learning Probabilistic Model Of Word Sense Disambiguation," In Information and Communication Technologies (WICT),World Congress 2012

[15] Francisco Tacoa, Danushka Bollegala and Mitsuru Ishizuka "A Context Expansion Method for Supervised Word Sense Disambiguation," In IEEE Sixth International Conference on Semantic Computing, 2012

[16] Wei Jan Lee and Edwin Mit "Word Sense Disambiguation by Using Domain Knowledge," Electronics Computer Technology (ICECT),3rd International Conference, 2011

[17] Richard L. Singh and Krishnendu Ghosh "Word Sense Disambiguation System in Manipuri Language," In International Conference on Computer Science and Information Technology, Goa, 2013

[18] Prity Bala "Word Sense Disambiguation Using Selectional Restriction". In International Journal of Scientific and Research Publications, 2013

[19] Rosna P. Haroon "Malayalam Word Sense Disambiguation," In Computational Intelligence and Computing Research (ICCIC),IEEE International Conference, 2010

[20] Jingbo Zhu, Huizhen Wang , Benjamin K. Tsou, and Matthew Ma, "Active Learning With Sampling by Uncertainty and Density for Data Annotations," In IEEE Transactions On Audio, Speech, And Language Processing, Vol. 18, pp. 1323-1331, 2010

[21] Rakesh Kumar, and Ravinder Khanna "Natural Language Engineering: The Study of Word Sense Disambiguation in Punjabi," In an International Journal of Engineering, Vol. 1, 2011

[22] Andres Montoyo, Armand Suarez, German Rigau and Manuel Palomar "Combining Knowledge- and Corpus-based Word-Sense-Disambiguation Methods," In Journal of Artificial Intelligence Research, 229-330 2005

[23] Zhang Zheng, and ZHU Shu, "A New Approach to Word Sense Disambiguation in MT System," In World Congress on Computer Science and Information Engineering , 2009

[24] Y.S. Alam "Lexical-Semantic Representation of the Lexicon for Word Sense Disambiguation and Text Understanding," In IEEE International Conference on Semantic Computing, 2009

[25] Jianping YU and Jian ZHANG "Word Sense Disambiguation of the English Modal Verb May by Back Propagation Neural Network," In Natural Language Processing and Knowledge Engineering, NLP-KE '08. International Conference, 2008

[26] Mohamed El Bachir Menai and Wojdan Alsaeedan "Genetic algorithm for Arabic word sense disambiguation," 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2013