Word Sense Disambiguation: An Empirical Survey

J. Sreedhar, S. Viswanadha Raju, A. Vinaya Babu, Amjan Shaik, P. Pavan Kumar

Abstract— Word Sense Disambiguation(WSD) is a vital area which is very useful in today's world. Many WSD algorithms are available in literature, we have chosen to opt for an optimal and portable WSD algorithms. We are discussed the supervised, unsupervised, and knowledge-based approaches for WSD. This paper will also furnish an idea of few of the WSD algorithms and their performances, Which compares and asses the need of the word sense disambiguity.

Index Terms—Supervised, Unsupervised, Knowledge-based , WSD.

I. INTRODUCTION

Language is communication media for the creatures among the races. The Language can able to exchange the information among the races. Language is the evolution criteria for the technology, the development of the Languages can able to exchange the thoughts, views, suggestions in an understandable way.The development in the human races has been observed when the communication among the people started to increase from the rock age. Communication is categorized into two types, verbal and nonverbal. Verbal communication is associated with alphabets, words, sentences etc. These are depends upon the language. The perfectness of the language depends upon the grammar rules associated with it.Verbal languages are of scripted and nonscripted. Verbal communication is the process of conveying orally. Examples presentations, discussions etc. Non-Verbal communication is adopted by the animal races. This communication is the process of exchange of information with signs and sounds. Non-verbal communication is the process of communication in the form of non-word messages. Examples gestures, facial expressions etc. Natural Language Processing (NLP) is a technique in which computerized approach to analyzing text is based on theories as well as technologies and both. NLP is a health research and development area in artificial intelligence (AI). NLP a theoretically motivated, multiple methods and techniques from which are selected for the accomplishment of particular type of language in analyzing and representing a human communicable at one or more level of linguistic analysis in the process of achieving human like languages processing for a range of tasks or applications. Natural language processing is the field of computer science which concerns with the interaction between the system and the User's language. NLP algorithms are grouped in statistical machine learning. Machine learning consists of two steps the training step and the evaluation step.

Word Sense Disambiguation is the common problem of NLP, which identifies the sense of the word used in the sentence or the query when it has multiple meanings. WSD is used to find the correct meaning of the sense or the word. A rich variety of techniques have been researched from dictionary-based methods that use knowledge encoded in lexical resources, supervised machine learning works on classifiers and unsupervised learning method supports clusters and many more as such.

This paper is organized as follows: first, we formalize the WSD (Section 2), and present the main approaches (Sections 3, 4, 5 and 6). Next, we turn to the evaluation of WSD (Section 7), and conclusion(Section 8) followed by the references.

II. WORD SENSE DISAMBIGUATION

Word Sense Disambiguation is the process of differentiating among the senses of words. Machine translation is one of the most former and on growing research computational linguistic. In 1940's WSD was developed as discrete field in computational linguistic due to fast research in of machine translation. In 1950's Weaver acknowledged that context is crucial and recognized the basic statistical character of the problem in proposing that statistical semantic studies should be undertaken as a necessary primary step. The automatic disambiguation of word senses has been an interest and concerned since the earliest days of computer treatment of languages in the 1950's. Then identifying work in estimating the degree of ambiguity in texts and bilingual dictionaries and applying simple statistical models. Sense disambiguation is an intermediate task which is not an end in itself, but rather is necessary at one level or another to accomplish most NLP task.

III. KNOWLEDGE BASED APPROACHES

The aim of Knowledge based approach (Dictionary based approach) WSD is to exploit knowledge resources to infer the senses of words in context. The knowledge resources are dictionaries, thesauri, ontology's, collocations etc The above methods have lower performance than their supervised alternative methods ,but they have an advantage of a wider range. In the year 1979 and 1980 the initial knowledge based approaches to word sense disambiguation taken place when experiments are conducted on extremely limited domains. Grading up these works was the main difficulty at that time. The proper evaluation comparison and exploitation of these methods in end to end application has been prevented because of large computational resources.

Manuscript received on April 26, 2012.

J. Sreedhar, Sr. Associate Professor in Computer Science & Engineering, ECET, Hyderabad, India, Mobile No: 8790423564, (e-mail: sreedharyd@gamil.com).

Dr. S. Viswanadha Raju, Professor in CSE, SIT, JNT University, Hyderabad, India, 9963701506, (e-mail: svraju.jntu@gmail.com).

Dr. A. Vinaya Babu, Professor in CSE & Principal in JNTUniversity, Hyderabad, India.

The overview of the main knowledge bases techniques, namely the overlap of sense definitions Selectional restrictions and structural approaches. A review knowledge based approaches can be found also in manning and schutze (1999) and mihalcea (2006).

Knowledge based approach have a faith on knowledge resources of machine readable dictionaries in form of corpus, WorldNet etc. they may use either grammar rules for disambiguation. A huge prominence of computer the large scale dictionaries are made available in form of MRD (machine readable dictionaries) like oxford English dictionary, Longman dictionary of ordinary contemporary English, Roget thesaurus and semantic networks which add more semantic relation like WorldNet, euro WorldNet. These are all for English [Collins,M., and Singer, Y. 1999].When it comes Indian national language that i.e. Hindi. The purpose data on which application has to be tested is provided by Central Institute of Indian Languages (CIIL Mysore), the MRD format is being is of WorldNet prepared by IIT Bombay.

For the purpose of Telugu language, the Telugu corpus is used in the thesis is prepared by Hyderabad Central University (HCU) in the format of MRD.

A. Selectional Preferences

A knowledge based algorithm is one which efforts of Selectional preferences to restrict the number of meanings of a target word occurring in context.A Selectional preferences or restrictions are constraints on semantic type that a word sense imposes on the words with which it combines usually through grammatical relationships in sentences. For example, the verbs eat as a subject entity expects an animate, when it as a direct objects an edible entity. The distinguish between selection restrictions and preferences. In the earlier rule out senses, violates the constraint but recent the Selectional of more appropriate sense which can satisfy better for the requirements' in considered.

The determination of the semantic appropriateness of the association provided by a word to word relation is the way to learn selectional preferences. The elementary measures of this kind are frequent count. p1and p2 are pair of words and syntactic relation R. the number of instances (R, p1, p2) in a corpus of parsed text.

Count (R1, p1, p2) Hindle and Rooth [1993].the other estimation of the semantic appropriates of a word to word relation is the conditional probability is the conditional probability of word p1 given to the word p2

R: P (p1/p2, R) =count (p1, p2, R)/count (p2, R)

Several techniques has been devised for measure of Selectional association [resnik 1993, 1997] for tree cut models using the minimum description length [li and Abe 1998, McCarthy and Carroll 2003] hidden markov model [abney and light 1999], class based probability [clark and weir 2002] the above approaches exploit large corpora and model the Selectional preferences of predicates by combining observed frequencies with knowledge about the semantic classes of their arguments. The disambiguation is performed with different means based on the strength of Selectional preferences towards a certain conceptual class.

B. Overlap Based Approaches

Overlap based approaches generally require a machine readable [MDR] [Duda, R. O. and Hart, P. 1973].The determination between the features of different senses of an ambiguous word (sense bag) and the features of the words in it context (control bag).Overlap based approach features may be a definition example sense hyponym. It is also given weights the sense which has maximum overlap is selected as the contextually appropriate sense.

The MRD'S like WorldNet, corpus thesaurus the relationship among the words provided by the thesaurus the disambiguation bases on thesaurus make use of semantic categorization provided by dictionary with sub categorization. The Roget's international thesaurus is highly adopted and frequently used thesaurus .the machine tractable from in 1950. The basic inference in thesaurus base disambiguation is that semantic categories of the words in context. Overlap based approach uses many algorithms the most commonly observed algorithms used for this approach are as follows :

WSD using conceptual density.
Lesk's algorithm
Walker's algorithm.

Conceptual Density of WSD:

Choose a sense based on the relatedness of the word sense to the context. Conceptual density is the measuring unit of relatedness (i.e. it represents how close the concept represented by the word and the concept represented by its context word. The conceptual distance is determined by structured hierarchical semantic net (WorldNet) conceptual distance and conceptual density is inverse proportional in nature. Higher the conceptual distance lower the density .the concept will have higher lower the density if all words in concept are strong indicator of a particular concept.

Lesk's approach:

The overlap based approach uses lesk algorithm. It can be explained as follows:

Consider a polysemous word W needing disambiguation let c the collection of a set of context words in its surrounding window.

There will be a lot of senses S for W of words each sense S of W the following

Let consider B be the bag of words obtained from the
Synonyms
Glosses
Example sentences
Hyponyms
Glosses of hyponyms
Example sentence of hyponyms
Hypernyms
Glosses of hypernyms
Example sentence of hypernyms
Meronyms
Glosses of meronyms
Example sentence of meronyms

Measure the overlap between C and D using the interaction similarity measure.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Output that the senses as the most probable sense which has the maximum overlap[Blum, A., and Chawla, S. 2001]

Walker's Approach:

In the year 1987, walker proposed an algorithm as follows. Considering a thesaurus each word is assigned to one or more subject categories in the theasurs. There are several subjects are assigned with a word then it is assumed that they correspond to different senses of the word. Black applied walker's approach to five different words and achieved accuracies of 50% [Blum, A., and Mitchell, T. 1998].

Wilk's Approach:

According to the wilks dictionary glosses are too short to result reliable disambiguation later he developed a context vector approach that expand the glosses with related words which allows for matching to be bases one more words. Then automatically results a fine distinctions in meaning than is possible with short glosses.

In the year 1990 the Longman's dictionary of contemporary English (LDOCE) because a standard work. Walker's approach has controlled definition vocabulary of app 2200 words which increase the likelihood of finding overlap among word sense.

IV. SUPERVISED DISAMBIGUATION

In the last 15 years, the NLP community has witnessed a significant shift from the use of manually crafted systems to the employment of automated classification methods [Cardie and Mooney 1999]. Such a dramatic increase of interest toward machinelearning techniques is reflected by the number of supervised approaches applied to the problem of WSD. Supervised WSD uses machine-learning techniques for inducing a classifier from manually sense-annotated data sets. Usually, the classifier (often called word expert) is concerned with a single word and performs a classification task in order to assign the appropriate sense to each instance of that word. The training set used to learn the classifier typically contains a set of examples in which a given target word is manually tagged with a sense from the sense inventory of a reference dictionary. Generally, supervised approaches to WSD have obtained better results than unsupervised methods (cf. Section 8). In the next subsections, we briefly review the most popular machine learning methods and contextualize them in the field of WSD. Additional information on the topic can be found in Manning and Sch¨ utze [1999], Jurafsky and Martin [2000], and M`arquez et al. [2006].

A. Decision Lists

A decision list [Rivest 1987] is an ordered set of rules for categorizing test instances (in the case of WSD, for assigning the appropriate sense to a target word). It can be seen as a list of weighted "if-then-else" rules. A training set is used for inducing a set of features. As a result, rules of the kind (feature-value, sense, score) are created. The ordering of these rules, based on their decreasing score, constitutes the decision list.

Given a word occurrence w and its representation as a feature vector, the decision list is checked, and the feature with highest score that matches the input vector selects the word sense to be assigned:

$$S = \operatorname{argmax}_{S_i \in Senses_{D(w)}} \text{Score}(S_i) \quad (1)$$

According to Yarowsky [1994], the score of sense Si is calculated as the maximum among the feature scores, where the score of a feature f is computed as the logarithm of the probability of sense Si given feature f divided by the sum of the probabilities of the other senses given feature f :

$$\text{Score}(S_i) = \max_f \log\left(\frac{P(S_i/f)}{\sum_{j \neq i} P(S_j/f)}\right) \quad (2)$$

The above formula is an adaptation to an arbitrary number of senses due to Agirre and Martinez [2000] of Yarowsky's [1994] formula, originally based on two senses. The probabilities P(Sj | f) can be estimated using the maximum-likelihood estimate. Smoothing can be applied to avoid the problem of zero counts. Pruning can also be employed to eliminate unreliable rules with very low weight.

B. Decision Trees

A decision tree is a predictive model used to represent classification rules with a tree structure that recursively partitions the training data set. Each internal node of a decision tree represents a test on a feature value, and each branch represents an outcome of the test. A prediction is made when a terminal node (i.e., a leaf) is reached.

In the last decades, decision trees have been rarely applied to WSD (in spite of some relatively old studies by, e.g., Kelly and Stone [1975] and Black [1988]). A popular algorithm for learning decision trees is the C4.5 algorithm [Quinlan 1993], an extension of the ID3 algorithm [Quinlan 1986]. In a comparative experiment with several machine learning algorithms for WSD, Mooney [1996] concluded that decision trees obtained with the C4.5 algorithm are outperformed by other supervised approaches. In fact, even though they represent the predictive model in a compact and human-readable way, they suffer from several issues, such as data sparseness due to features with a large number of values, unreliability of the predictions due to small training sets, etc. For instance, if the noun bank must be classified in the sentence "we sat on a bank of sand," the tree is traversed and, after following the no-yes-no path, the choice of sense bank/RIVER is made. The leaf with empty value (-) indicates that no choice can be made based on specific feature values.

C. Naive Bayes

A Naive Bayes classifier is a simple probabilistic classifier based on the application of Bayes' theorem. It relies on the calculation of the conditional probability of each sense Si of a word w given the features f j in the context. The sense S which maximizes the following formula is chosen as the most appropriate sense in context:

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

$$\hat{S} = argmax_{S_i \in Senses_{D(W)}} P\left(s_i / f_1, \ldots, f_m\right) = argmax_{S_i \in Senses_{D(W)}} \frac{P\left(f_1, \ldots f_m / s_i\right) P(s_i)}{P\left(f_1, \ldots f_m\right)} \quad (3)$$

$$\hat{S} = argmax_{S_i \in Senses_{D(W)}} P(S_i) \pi_{j=1}^{m} P\left(f_j / s_i\right) \quad (4)$$

where m is the number of features, and the last formula is obtained based on the naive assumption that the features are conditionally independent given the sense (the denominator is also discarded as it does not influence the calculations). The probabilities P (Si) and P (f j | Si) are estimated, respectively, as the relative occurrence frequencies in the training set of sense Si and feature f j in the presence of sense Si. Zero counts need to be smoothed: for instance, they can be replaced with P (Si)/N where N is the size of the training set [Ng 1997; Escudero et al. 2000c]. However, this solution leads probabilities to sum to more than 1. Backoff or interpolation strategies can be used instead to avoid this problem.

D. Neural Networks

A neural network [McCulloch and Pitts 1943] is an interconnected group of artificial neurons that uses a computational model for processing data based on a connectionist approach. Pairs of (input feature, desired response) are input to the learning program.

The aim is to use the input features to partition the training contexts into nonoverlapping sets corresponding to the desired responses. As new pairs are provided, link weights are progressively adjusted so that the output unit representing the desired response has a larger activation than any other output unit. Neural networks are trained until the output of the unit corresponding to the desired response is greater than the output of any other unit for every training example. For testing, the classification determined by the network is given by the unit with the largest output. Weights in the network can be either positive or negative, thus enabling the accumulation of evidence in favour or against a sense choice.

Cottrell [1989] employed neural networks to represent words as nodes: the words activate the concepts to which they are semantically related and vice versa. The acti- vation of a node causes the activation of nodes to which it is connected by excitory links and the deactivation of those to which it is connected by inhibitory links (i.e., competing senses of the same word). V´ eronis and Ide built a neural network from the dictionary definitions of the Collins English Dictionary. They connect words to their senses and each sense to words occurring in their textual definition. Recently, Tsatsaronis et al. [2007] successfully extended this approach to include all related senses linked by semantic relations in the reference resource, that is WordNet. Finally, Towell and Voorhees [1998] found that neural networks perform better without the use of hidden layers of nodes and used perceptrons for linking local and topical input features directly to output units (which represent senses).

In several studies, neural networks have been shown to perform well compared to other supervised methods [Leacock et al. 1993; Towell and Voorhees 1998; Mooney 1996]. However, these experiments are often performed on a small number of words. As major drawbacks of neural networks we cite the difficulty in interpreting the results, the need for a large quantity of training data, and the tuning of parameters such as thresholds, decay, etc

V. UNSUPERVISED DISAMBIGUATION

Be Unsupervised methods have the potential to overcome the knowledge acquisition bottleneck [Gale et al. 1992b], that is, the lack of large-scale resources manually annotated with word senses. These approaches to WSD are based on the idea that the same sense of a word will have similar neighboring words. They are able to induce word senses from input text by clustering word occurrences, and then classifying new occurrences into the induced clusters. They do not rely on labeled training text and, in their purest version, do not make use of any machine-readable resources like dictionaries, thesauri, ontolo- gies, etc. However, the main disadvantage of fully unsupervised systems is that, as they do not exploit any dictionary, they cannot rely on a shared reference inventory of senses.

While WSD is typically identified as a sense labeling task, that is, the explicit assignment of a sense label to a target word, unsupervised WSD performs word sense discrimination, that is, it aims to divide "the occurrences of a word into a number of classes by determining for any two occurrences whether they belong to the same sense or not" [Schutze 1998, page 97]. Consequently, these methods may not discover clusters equivalent to the traditional senses in a dictionary sense inventory. For this reason, their evaluation is usually more difficult: in order to assess the quality of a sense cluster we should ask humans to look at the members of each cluster and determine the nature of the relationship that they all share (e.g., via questionnaires), or employ the clusters in end-to-end applications, thus measuring the quality of the former based on the performance of the latter.

Admittedly, unsupervised WSD approaches have a different aim than supervised and knowledge-based methods, that is, that of identifying sense clusters compared to that of assigning sense labels. However, sense discrimination and sense labeling are both sub problems of the word sense disambiguation task [Schutze 1998] and are strictly related, to the point that the clusters produced can be used at a later stage to sense tag word occurrences.

A. Context Clustering

A first set of unsupervised approaches is based on the notion of context clustering. Each occurrence of a target word in a corpus is represented as a context vector. The vectors are then clustered into groups, each identifying a sense of the target word.

A historical approach of this kind is based on the idea of word space [Schutze 1992].

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

That is, a vector space whose dimensions are words. A word w in a corpus can be represented as a vector whose j th component counts the number of times that word w j cooccurs with w within a fixed context (a sentence or a larger context). The underlying hypothesis of this model is that the distributional profile of words implicitly expresses their semantics.

The similarity between two words v and w can then be measured geometrically, for example, by the cosine between the corresponding vectors v and w:

$$Sim(V,W) = \frac{v.w}{|v||w|} = \frac{\sum_{i=1}^{m} v_i w_i}{\sqrt{\sum_{i=1}^{m} v_i^2 \sum_{i=1}^{m} w_i^2}} \quad (5)$$

where m is the number of features in each vector. A vector is computed for each word in a corpus. This kind of representation conflates senses: a vector includes all the senses of the word it represents (e.g., the senses stock as a supply and as capital are all summed in its word vector).

If we put together the set of vectors for each word in the corpus, we obtain a cooccurrence matrix. As we might deal with a large number of dimensions, latent semantic analysis (LSA) can be applied to reduce the dimensionality of the resulting multidimensional space via singular value decomposition (SVD) [Golub and van Loan 1989]. SVD finds the major axes of variation in the word space. The dimensionality reduction has the effect of taking the set of word vectors in the highdimensional space and represent them in a lower-dimensional space: as a result, the dimensions associated with terms that have similar meanings are expected to be merged. After the reduction, contextual similarity between two words can be measured again in terms of the cosine between the corresponding vectors.

Finally, sense discrimination can be performed by grouping the context vectors of a target word using a clustering algorithm. Schutze [1998] proposed an algorithm, called context-group discrimination, which groups the occurrences of an ambiguous word into clusters of senses, based on the contextual similarity between occurrences. Contextual similarity is calculated as described above, whereas clustering is performed with the Expectation Maximization algorithm, an iterative maximum likelihood estimation procedure of a probabilistic model [Dempster et al. 1977]. A different clustering approach consists of agglomerative clustering [Pedersen and Bruce 1997]. Initially, each instance constitutes a singleton cluster. Next, agglomerative clustering merges the most similar pair of clusters, and continues with successively less similar pairs until a stopping threshold is reached. The performance of the agglomerative clustering of context vec- tors was assessed in an unconstrained setting [Pedersen and Bruce 1997] and in the biomedical domain [Savova et al. 2005]. A problem in the construction of context vectors is that a large amount of (unlabeled) training data is required to determine a significant distribution of word cooccurrences.

This issue can be addressed by augmenting the feature vector of each word with the content words occurring in the glosses of its senses [Purandare and Pedersen 2004] (note the circularity of this approach, which makes it semisupervised: we use an existing sense inventory to discriminate word senses). A further issue that can be addressed is the fact that different context clusters might not correspond to distinct word senses. A supervised classifier can be trained and subsequently applied to tackle this issue [Niu et al. 2005].

Multilingual context vectors are also used to determine word senses [Ide et al. 2001]. In this setting, a word occurrence in a multilingual corpus is represented as a context vector which includes all the possible lexical translations of the target word w, whose value is 1 if the specific occurrence of w can be translated accordingly, and zero otherwise.

B. Word Clustering

In the previous section we represented word senses as first- or second-order context vectors. A different approach to the induction of word senses consists of word clustering techniques, that is, methods which aim at clustering words which are semantically similar and can thus convey a specific meaning.

A well-known approach to word clustering [Lin 1998a] consists of the identification of words W = (w1, . . . , wk) similar (possibly synonymous) to a target word w0. The similarity between w0 and wi is determined based on the information content of their single features, given by the syntactic dependencies which occur in a corpus (such as, e.g., subject-verb, verb-object, adjective-noun, etc.). The more dependencies the two words share, the higher the information content. However, as for context vectors, the words in W will cover all senses of w0. To discriminate between the senses, a word clustering algorithm is applied. Let W be the list of similar words ordered by degree of similarity to w0. A similarity tree T is initially created which consists of a single node w0. Next, for each i ∈ {1, . . . , k}, wi ∈ W is added as a child of w j in the tree T such that w j is the most similar word to wi among {w0, . . . , wi−1}. After a pruning step, each subtree rooted at w0 is considered as a distinct sense of w0.

In a subsequent approach, called the clustering by committee (CBC) algorithm [Lin and Pantel 2002], a different word clustering method was proposed. For each target word, a set of similar words was computed as above. To calculate the similarity, again, each word is represented as a feature vector, where each feature is the expression of a syntactic context in which the word occurs. Given a set of target words (e.g., all those occurring in a corpus), a similarity matrix S is built such that Sij contains the pairwise similarity between words wi and w j .

As a second step, given a set of words E, a recursive procedure is applied to determine sets of clusters, called committees, of the words in E. To this end, a standard clustering technique, that is, average-link clustering, is employed. In each step, residue words not covered by any committee (i.e., not similar enough to the centroid of each committee) are identified. Recursive attempts are made to discover more committees from residue words. Notice that, as above, committees conflate senses as each word belongs to a single committee. Finally, as a sense discrimination step, each target word w ∈ E, again represented as a feature vector, is iteratively assigned to its most similar cluster, based on its

similarity to the centroid of each committee. After a word w is assigned to a committee c, the intersecting features between w and elements in c are removed from the representation of w, so as to allow for the identification of less frequent senses of the same word at a later iteration.

C. Cooccurrence Graphs

A different view of word sense discrimination is provided by graph-based approaches, which have been recently explored with a certain success. These approaches are based on the notion of a cooccurrence graph, that is, a graph G = (V , E) whose vertices V correspond to words in a text and edges E connect pairs of words which cooccur in a syntactic relation, in the same paragraph, or in a larger context.

The construction of a cooccurrence graph based on grammatical relations between words in context was described by Widdows and Dorow [2002] (see also Dorow and Widdows [2003]). Given a target ambiguous word w, a local graph Gw is built around w. By normalizing the adjacency matrix associated with Gw, we can interpret the graph as a Markov chain. The Markov clustering algorithm [van Dongen 2000] is then applied to determine word senses, based on an expansion and an inflation step, aiming, respectively, at inspecting new more distant neighbors and supporting more popular nodes.

Subsequently, V´ eronis [2004] proposed an ad hoc approach called HyperLex. First, a cooccurrence graph is built such that nodes are words occurring in the paragraphs of a text corpus in which a target word occurs, and an edge between a pair of words is added to the graph if they cooccur in the same paragraph. Each edge is assigned a weight according to the relative cooccurrence frequency of the two words connected by the edge. Formally, given an edge {i, j } its weight wij is given by

$$W_{ij} = 1\text{-}\mathrm{Max}\left\{P\left(W_i/W_j\right), P\left(W_j/w_i\right)\right\} \quad (6)$$

where P(wi | wj) = f reqi j/ f reqj , and freqij is the frequency of cooccurrence of words wi and wj and freq j is the frequency of w j within the text. As a result, words with high frequency of cooccurrence are assigned a weight close to zero, whereas words which rarely occur together receive weights close to 1

As a second step, an iterative algorithm is applied to the cooccurrence graph: at each iteration, the node with highest relative degree in the graph is selected as a hub (based on the experimental finding that a node's degree and its frequency in the original text are highly correlated). As a result, all its neighbors are no longer eligible as hub candidates. The algorithm stops when the relative frequency of the word corresponding to the selected hub is below a fixed threshold. The entire set of hubs selected is said to represent the senses of the word of interest.

Finally, the MST is used to disambiguate specific instances of our target word. Let W = (w1, w2, . . . , wi, . . . , wn) be a context in which wi is an instance of our target word.

A score vector s is associated with each w j ∈ W (j = i), such that its kth component sk represents the contribution of the kth hub as follows:

$$S_k = \left\{\frac{1}{1+d(h_k, w_j)}\right\} \quad (7)$$

where d (hk, w j) is the distance between root hub hk and node w j (possibly, hk ≡ w j). Next, all score vectors associated with all w j ∈ W (j = i) are summed up and the hub which receives the maximum score is chosen as the most appropriate sense for wi.

An alternative graph-based algorithm for inducing word senses is PageRank [Brin and Page 1998]. PageRank is a well-known algorithm developed for computing the ranking of web pages, and is the main ingredient of the Google search engine. It has been employed in several research areas for determining the importance of entities whose relations can be represented in terms of a graph. In its weighted formulation, the PageRank degree of a vertex vi ∈ V is given by the following formula:

$$P(V_i) = (1-d) + d\sum_{v_j \to v_i} \frac{W_{ij}}{\sum_{v_j \to v_k} W_{jk}} P(V_j) \quad (8)$$

where vj → vi denotes the existence of an edge from vj to vi, w ji is its weight, and d is a damping factor (usually set to 0.85) which models the probability of following a link to vi (second term) or randomly jumping to vi (first term in the equation). Notice the recursive nature of the above formula: the PageRank of each vertex is iteratively computed until convergence.

In the adaptation of PageRank to unsupervised WSD (due to Agirre et al. [2006]), w ji is, as for HyperLex, a function of the probability of cooccurrence of words wi and w j . As a result of a run of the PageRank algorithm, the vertices are sorted by their PageRank value, and the best ranking ones are chosen as hubs of the target word.

VI. KNOWLEDGE-BASED DISAMBIGUATION

The objective of knowledge-based or dictionary-based WSD is to exploit knowledge resources to infer the senses of words in context. These methods usually have lower performance than their supervised alternatives, but they have the advantage of a wider coverage, thanks to the use of large-scale knowledge resources.

The first knowledge-based approaches to WSD date back to the 1970s and 1980s when experiments were conducted on extremely limited domains. Scaling up these works was the main difficulty at that time: the lack of large-scale computational resources prevented a proper evaluation, comparison and exploitation of those methods in end- to-end applications.

A. Overlap of Sense Definitions

A simple and intuitive knowledge-based approach relies on the calculation of the word overlap between the sense definitions of two or more target words. This approach is named gloss overlap or the Lesk algorithm after its author [Lesk 1986]. Given a two word context (w1, w2), the senses of the target words whose definitions have the highest overlap (i.e., words in common) are assumed to be the correct ones. Formally, given two words w1

and w2, the following score is computed for each pair of word senses S1 ∈ Senses(w1) and S2 ∈ Senses(w2):

$$Score_{Lesk}(S_1S_2) = |gloss(S_1) \cap gloss(S_2)| \quad (9)$$

where gloss(Si) is the bag of words in the textual definition of sense Si of wi.

The senses which maximize the above formula are assigned to the respective words. However, this requires the calculation of | Senses(w1) | · | Senses(w2) | gloss overlaps. If we extend the algorithm to a context of n words, we need to determine n= | Senses(wi) | overlaps.

Given the exponential number of steps required, a variant of the Lesk algorithm is currently employed which identifies the sense of a word w whose textual definition has the highest overlap with the words in the context of w. Formally, given a target word w, the following score is computed for each sense S of w:

$$Score_{leskvar}(S) = |context(w) \cap gloss(S)| \quad (10)$$

where context(w) is the bag of all content words in a context window around the target word w.

As an example, in Table V we show the first three senses in WordNet of keyn and mark in italic the words which overlap with the following input sentence:

Sense 1 of key has 3 overlaps, whereas the other two senses have zero, so the first sense is selected.

The original method achieved 50-70% accuracy (depending on the word), using a relatively fine set of sense distinctions such as those found in a typical learner's dictionary [Lesk 1986]. Unfortunately, Lesk's approach is very sensitive to the exact wording of definitions, so the absence of a certain word can radically change the results.

Further, the algorithm determines overlaps only among the glosses of the senses being considered. This is a significant limitation in that dictionary glosses tend to be fairly short and do not provide sufficient vocabulary to relate fine-grained sense distinctions.

Recently, Banerjee and Pedersen [2003] introduced a measure of extended gloss overlap, which expands the glosses of the words being compared to include glosses of concepts that are known to be related through explicit relations in the dictionary (e.g., hypernymy, meronymy, pertainymy, etc.). The range of relationships used to extend the glosses is a parameter, and can be chosen from any combination of WordNet relations.

For each sense S of a target word w we estimate its score as

$$Score_{ExtLesk}(s) = \sum_{s' is \to s' or s=s'} |context(W) \cap gloss(s')| \quad (11)$$

where context(w) is, as above, the bag of all content words in a context window around the target word w and gloss(S) is the bag of words in the textual definition of a sense S which is either S itself or related to S through a relation rel . The overlap scoring mechanism is also parametrized and can be adjusted to take into account gloss length (i.e. normalization) or to include function words.

Banerjee and Pedersen [2003] showed that disambiguation greatly benefits from the use of gloss information from related concepts (jumping from 18.3% for the original Lesk algorithm to 34.6% accuracy for extended Lesk). However, the approach does not lead to state-of-the-art performance compared to competing knowledge-based systems.

B. Selectional Preferences

A historical type of knowledge-based algorithm is one which exploits selectional preferences to restrict the number of meanings of a target word occurring in context. Selectional preferences or restrictions are constraints on the semantic type that a word sense imposes on the words with which it combines in sentences (usually through grammatical relationships). For instance, the verb eat expects an animate entity as subject and an edible entity as its direct object. We can distinguish between selectional restrictions and preferences in that the former rule out senses that violate the constraint, whereas the latter (more typical of recent empirical work) tend to select those senses which better satisfy the requirements.

The easiest way to learn selectional preferences is to determine the semantic appropriateness of the association provided by a word-to-word relation. The simplest measure of this kind is frequency count. Given a pair of words w1 and w2 and a syntactic relation R (e.g., subject-verb, verb-object, etc.), this method counts the number of instances (R, w1, w2) in a corpus of parsed text, obtaining a figure Count(R, w1, w2) (see, e.g., Hindle and Rooth [1993]). Another estimation of the semantic appropriateness of a word-to-word relation is the conditional probability of word w1 given the other word w2 and the relation R:

$$P\left(W_1/W_2, R\right) = \frac{Count(w W_2, R)}{Count(W_2, R)} \quad (12)$$

To provide word-to-class or class-to-class models, that is, to generalize the knowledge acquired to semantic classes and relieve the data sparseness problem, manually crafted taxonomies such as WordNet can used to derive a mapping from words to conceptual classes. Several techniques have been devised, from measures of selectional association [Resnik 1993, 1997], to tree cut models using the minimum description length [Li and Abe 1998; McCarthy and Carroll 2003], hidden markov models [Abney and Light 1999], The scoring function presented here is a variant of that presented by Banerjee and Pedersen [2003].

Class-based probability [Clark and Weir 2002; Agirre and Martinez 2001], Bayesian networks [Ciaramita and Johnson 2000], etc. Almost all these approaches exploit large corpora and model the selectional preferences of predicates by combining observed frequencies with knowledge about the semantic classes of their arguments (the latter obtained from corpora or dictionaries). Disambiguation is then performed with different means based on the strength of a selectional preference towards a certain conceptual class (i.e., sense choice).

A comparison of word-to-word, word-to-class, and class-to-class approaches was pre- sented by Agirre and Martinez [2001], who found out that the coverage grows as we move from the former to the latter methods (26% for word-to-word preferences, 86.7% for word-to-class, 97.3% for class-to-class methods), and that precision decreases accordingly (from 95.9% to 66.9% to 66.6%, respectively).

In general, we can say that approaches to WSD based on selectional restrictions have not been found to perform as well as Lesk-based methods or the most frequent sense heuristic (see Section 7.2.2).

C. Structural Approaches

Since the availability of computational lexicons like WordNet, a number of structural approaches have been developed to analyze and exploit the structure of the concept network made available in such lexicons. The recognition and measurement of patterns, both in a local and a global context, can be collocated in the field of structural pattern recognition [Fu 1982; Bunke and Sanfeliu 1990], which aims at classifying data (specifically, senses) based on the structural interrelationships of features. We present two main approaches of this kind: similarity-based and graph-based methods.

D. Support Vector Machines (SVM) (13)

This method (introduced by Boser et al. [1992]) is based on the idea of learning a linear hyperplane from the training set that separates positive examples from negative examples. The hyperplane is located in that point of the hyperspace which maximizes the distance to the closest positive and negative examples (called support vectors). In other words, support vector machines (SVMs) tend at the same time to minimize the empirical classification error and maximize the geometric margin between positive and negative examples.

As SVM is a binary classifier, in order to be usable for WSD it must be adapted to multiclass classification (i.e., the senses of a target word). A simple possibility, for instance, is to reduce the multiclass classification problem to a number of binary classifications of the kind sense Si versus all other senses. As a result, the sense with the highest confidence is selected.

It can be shown that the classification formula of SVM can be reduced to a function of the support vectors, which—in its linear form—determines the dot product of pairs of vectors. More in general, the similarity between two vectors x and y is calculated with a function called kernel which maps the original space (e.g., of the training and test instances) into a feature space such that k(x, y) = (x) · (y), where is a transformation (the simplest kernel is the dot product k(x, y) = x · y). A nonlinear transformation might be chosen to change the original representation into one that is more suitable for the problem (the so-called kernel trick). The capability to map vector spaces to higher dimensions with kernel methods, together with its high degree of adaptability based on parameter tuning, are among the key success factors of SVM.

SVM has been applied to a number of problems in NLP, including text categorization [Joachims 1998], chunking [Kudo and Matsumoto 2001], parsing [Collins 2004], and WSD [Escudero et al. 2000c; Murata et al. 2001; Keok and Ng 2002]. SVM has been shown to achieve the best results in WSD compared to several supervised approaches [Keok and Ng 2002].

VII. Evaluation Methodology

We present here the evaluation measures and baselines employed for in vitro evaluation of WSD systems, that is, as if they were stand-alone, independent applications.

However, one of the real objectives of WSD is to demonstrate that it improves the performance of applications such as information retrieval, machine translation, etc. The evaluation of WSD as a module embedded in applications is called in vivo or end-to-end evaluation. We will discuss this second kind of evaluation in later sections.

A. Evaluation Measures

The assessment of word sense disambiguation systems is usually performed in terms of evaluation measures borrowed from the field of information retrieval, that we introduce hereafter.

Let T = (w1, . . . , wn) be a test set and A an "answer" function that associates with each word wi ∈ T the appropriate set of senses from the dictionary D (i.e., A(i) ⊆ SensesD(wi)). Then, given the sense assignments A (i) ∈ SensesD(wi) ∪ { } provided by an automatic WSD system11 (i ∈ {1, . . . , n}), we can define coverage C as the percentage of items in the test set for which the system provided a sense assignment that is:

$$C=\frac{\#\ answers\ provided}{\#\ total\ answers\ to\ provide}=\frac{|\{i\in\{1,\ldots,n\}:A'(i)\in A(i)\}|}{n} \qquad (14)$$

where we indicate with the case in which the system does not provide an answer for a specific word wi (i.e., in that case we assume that A (i) =). The total number of answers is given by n =| T |. The precision P of a system is computed as the percentage of correct answers given by the automatic system, that is:

$$P=\frac{\#\ correct\ answers\ provided}{\#\ answers\ provided}=\frac{|\{i\in\{1,\ldots,n\}:A'(i)\in A(i)\}|}{|\{i\in\{1,\ldots,n\}:A'(i)\neq\in\}|} \qquad (15)$$

We assume that the annotations to be assessed assign to each word a single sense from the inventory. We note that more than one annotation can be allowed by extending this notation.

Precision determines how good are the answers given by the system being assessed. Recall R is defined as the number of correct answers given by the automatic system over the total number of answers to be given:

$$R=\frac{\#\ correct\ answers\ provided}{\#\ total\ answers\ to\ provide}=\frac{|\{i\in\{1,\ldots,n\}:A'(i)\in A(i)\}|}{n} \qquad (16)$$

According to the above definitions, we have that R ≤ P . When coverage is 100%, we have that P= R. In the WSD literature, recall is also referred to as accuracy, although these are two different measures in the machine learning and information retrieval literature.

Finally, a measure which determines the weighted harmonic mean of precision and recall, called the F1-measure or balanced F -score, is defined as

$$F_1=\frac{2PR}{P+R} \qquad (17)$$

The F1-measure is a specialization of a general formula, the Fα-score, defined as

$$F_\alpha=\frac{1}{\alpha\frac{1}{p}+(1-\alpha)\frac{1}{R}}=\frac{(\beta^2+1)}{\beta^2P+R} \qquad (18)$$

where α = 1/(β2 + 1).

The F1-measure is obtained by choosing β = 1 (or, equivalently, α = 1), thus equally balancing precision and recall. F1 is useful to compare systems 2 with a coverage lower than 100%. Note that an easy-to-build system with P = 100% and almost-zero recall would get around 50% performance if we used a simple arithmetic mean (P+R), whereas a harmonic mean such as F1 is dramatically penalized by low 2 values of either precision or recall.

It has been argued that the above measures do not reflect the ability of systems to output a degree of confidence for a given sense choice. In this direction, Resnik and Yarowsky [1999] proposed an evaluation metric which weighs misclassification errors by the distance between the selected and correct senses. As a result, if the chosen sense is a fine-grained distinction of the correct sense, this error will be penalised less heavily than between coarser sense distinctions. Even more refined metrics, such as the receiver operation characteristic (ROC), have been proposed [Cohn 2003]. How- ever these metrics are not often used, also for reasons of comparison with previously established results, mostly measured in terms of precision, recall, and F1.

VIII. CONCLUSION

In this paper we empirically surveye the field of word sense disambiguation(WSD). WSD is a hard task as it deals with the full complexities of language and aims at identifying a semantic structure from apparently unstructured sources of text. The hardness of WSD strictly depends on the granularity of the sense distinctions taken into account. Supervised methods undoubtedly perform better than other approaches. However, re- lying on the availability of large training corpora for different domains, languages, and tasks is not a realistic assumption. This paper will also furnish an idea of few of the WSD algorithms and their performances, Which compares and asses the need of the word sense disambiguity.

REFERENCES

1. Collins,M., and Singer, Y. 1999. Unsupervised models for namedentity classification. In EMNLP/VLC-99.
2. RESNIK, P. 1997. Selectional preference and sense disambiguation. In Proceedings of the ACL SIGLEXWorkshop on Tagging Text with Lexical Semantics: Why, What, and How? (Washington, D.C.). 52–57.
3. LI, H. AND ABE, N. 1998. Generalizing case frames using a thesaurus and the MDL principle. Computat. Ling. 24, 2, 217–244.
4. ABNEY, S. AND LIGHT, M. 1999. Hiding a semantic class hierarchy in a Markov model. In Proceedings of the ACL Workshop on Unsupervised Learning in Natural Language Processing (College Park, MD). 1–8.
5. CLARK, S. AND WEIR, D. 2002. Class-based probability estimation using a semantic hierarchy. Computat. Ling. 28, 2, 187–206.
6. Duda, R. O. and Hart, P. E.: Pattern Classification and Scene Analysis. Wiley, New York, 1973.
7. Blum, A., and Chawla, S. 2001. Learning from labeled and unla-beled data using graph mincuts. In ICML-2001.
8. Blum, A., and Mitchell, T. 1998. Combining labeled and unla-beled data with co-training. In COLT-98.
9. CARDIE, C. AND MOONEY, R. J. 1999. Guest editors' introduction: Machine learning and natural language. Mach. Learn. 34, 1–3, 5–9.
10. SCHUTZE, H. 1998. Automatic word sense discrimination. Computat. Ling. 24, 1, 97–124.
11. MARTINEZ, D. 2004. Supervised word sense disambiguation: Facing current challenges, Ph.D. dissertation. University of the Basque Country, Spain.
12. MARQUEZ, L., ESCUDERO, G., MART`INEZ, D., AND RIGAU, G. 2006. Supervised corpus-based methods for WSD. In Word Sense Disambiguation: Algorithms and Applications, E. Agirre and P. Edmonds, Eds. Springer, New York, NY, 167–216.
13. RIVEST, R. L. 1987. Learning decision lists. Mach. Learn. 2, 3, 229–246.
14. YAROWSKY, D. 1994. Decision lists for lexical ambiguity resolution: Application to accent restoration in Spanish and French. In Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (Las Cruces, NM). 88–95.
15. AGIRRE, E. AND MARTINEZ, D. 2000. Exploring automatic word sense disambiguation with decision lists and the web. In Proceedings of the 18th International Conference on Computational Linguistics (COLING, Saarbr ¨ ucken, Germany). 11–19.
16. Saa KELLY, E. AND STONE, P. 1975. Computer Recognition of English Word Senses. Vol. 3 of North Holland Linguistics Series. Elsevier, Amsterdam, The Netherlands.
17. BLACK, E. 1988. An experiment in computational discrimination of English word senses. IBM J. Res. Devel. 32, 2, 185–194
18. J QUINLAN, J. R. 1986. Induction of decision trees. Mach. Learn. 1, 1, 81–106.
19. QUINLAN, J. R. 1993. Programs for Machine Learning. Morgan Kaufmann, San Francisco, CA.
20. MOONEY, R. J. 1996. Comparative experiments on disambiguating word senses: An illustration of the role of bias in machine learning. In Proceedings of the 1996 Conference on Empirical Methods in Natural
21. Language Processing (EMNLP). 82–91.
22. NG, T. H. 1997. Getting serious about word sense disambiguation. In Proceedings of the ACL SIGLEX
23. Workshop on Tagging Text with Lexical Semantics: Why, What, and How? (Washington D.C.). 1–7.
24. MCCULLOCH, W. AND PITTS, W. 1943. A logical calculus of the ideas immanent in nervous activity. Bull. Math. Biophys. 5, 115–133.
25. COTTRELL, G. W. 1989. A Connectionist Approach to Word Sense Disambiguation. Pitman, London, U.K.
26. TSATSARONIS, G.VAZIRGIANNIS, M., AND ANDROUTSOPOULOS, I. 2007. Word sense disambiguation with spreading activation networks generated from thesauri. In Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI, Hyderabad, India). 1725–1730.
27. TOWELL, G. AND VOORHEES, E. 1998. Disambiguating highly ambiguous words. Computat. Ling. 24, 1, 125–145.
28. LEACOCK, C., TOWELL, G., AND VOORHEES, E. 1993. Corpus-based statistical sense resolution. In Proceedings of the ARPA Workshop on Human Language Technology (Princeton, NJ). 260–265.
29. GALE, W. A., CHURCH, K., AND YAROWSKY, D. 1992b. A method for disambiguating word senses in a corpus. Comput. Human. 26, 415–439.
30. SCHUTZE, H. 1998. Automatic word sense discrimination. Computat. Ling. 24, 1, 97–124.
31. SCHUTZE, H. 1992. Dimensions of meaning. In Supercomputing '92: Proceedings of the 1992 ACM/IEEE Conference on Supercomputing. IEEE Computer Society Press, Los Alamitos, CA. 787–796.
32. GOLUB, G. H. AND VAN LOAN, C. F. 1989. Matrix Computations. The John Hopkins University Press, Baltimore, MD.
33. DEMPSTER, A., LAIRD, N., AND RUBIN, D. 1977. Maximum likelihood from incomplete data via the EM algorithm. J. Roy. Statist. Soc. 39, 1–38.
34. PEDERSEN, T. AND BRUCE, R. 1997. Distinguishing word senses in untagged text. In Proceedings of the 1997 Conference on Empirical Methods in Natural Language Processing (EMNLP, Providence, RI). 197–207.
35. SAVOVA, G., PEDERSEN, T., PURANDARE, A., AND KULKARNI, A. 2005. Resolving ambiguities in biomedical text with unsupervised clustering approaches. Res. rep. UMSI 2005/80. University of Minnesota Supercomputing Institute, Minneapolis, MN.
36. PURANDARE, A. AND PEDERSEN, T. 2004. Improving word sense discrimination with gloss augmented feature vectors. In Proceedings of theWorkshop on Lexical Resources for theWeb andWord Sense Disambiguation (Puebla, Mexico). 123–130.
37. NIU, C., LI, W., SRIHARI, R., AND LI, H. 2005. Word independent context pair classification model for word sense disambiguation. In Proceedings of the 9th Conference on Computational Natural Language Learning(CoNLL, Ann Arbor, MI).

38. IDE, N., ERJAVEC, T., AND TUFIS, D. 2001. Automatic sense tagging using parallel corpora. In Proceedings of the 6th Natural Language Processing Pacific Rim Symposium (Tokyo, Japan). 83–89.
39. LIN,D. 1998a. Automatic retrieval and clustering of similar words. In Proceedings of the 17th International Conference on Computational linguistics (COLING, Montreal, P.Q., Canada). 768–774.
40. LIN, D. AND PANTEL, P. 2002. Discovering word senses from text. In Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (Edmonton, Alta., Canada). 613–619.
41. WIDDOWS, D. AND DOROW, B. 2002. A graph model for unsupervised lexical acquisition. In Proceedings of the 19th International Conference on Computational Linguistics (COLING, Taipei, Taiwan). 1–7.
42. DOROW, B. AND WIDDOWS, D. 2003. Discovering corpus-specific word senses. In Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (Budapest, Hungary). 79–82.
43. VAN DONGEN, S. 2000. Graph Clustering by Flow Simulation, Ph.D. dissertation. University of Utrecht,Utrecht, The Netherlands.
44. V´ERONIS, J. 2004. Hyperlex: Lexical cartography for information retrieval. Comput. Speech Lang. 18, 3,223–252.
45. BRIN, S. AND PAGE, M. 1998. Anatomy of a large-scale hypertextual Web search engine. In Proceedings of the 7th Conference on World Wide Web (Brisbane, Australia). 107–117.
46. AGIRRE, E. AND EDMONDS, P., Eds. 2006. Word Sense Disambiguation: Algorithms and Applications.Springer, New York, NY.
47. LESK, M. 1986. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In Proceedings of the 5th SIGDOC (New York, NY). 24–26.
48. HINDLE, D. AND ROOTH,M. 1993. Structural ambiguity and lexical relations. Computat. Ling. 19, 1, 103–120.
49. RESNIK, P. S., Ed. 1993. Selection and information: A class-based approach to lexical relationships, Ph.D. dissertation. University of Pennsylvania, Pennsylvania, Philadelphia, PA.
50. LI, H. AND ABE, N. 1998. Generalizing case frames using a thesaurus and the MDL principle. Computat. Ling. 24, 2, 217–244.
51. ABNEY, S. AND LIGHT, M. 1999. Hiding a semantic class hierarchy in a Markov model. In Proceedings of the ACL Workshop on Unsupervised Learning in Natural Language Processing (College Park, MD). 1–8.
52. CLARK, S. AND WEIR, D. 2002. Class-based probability estimation using a semantic hierarchy. Computat. Ling. 28, 2, 187–206.
53. AGIRRE, E. ANDMARTINEZ, D. 2001. Learning class-to-class selectional preferences. In Proceedings of the 5th Conference on Computational Natural Language Learning (CoNLL, Toulouse, France). 15–22.
54. FU, K. 1982. Syntactic Pattern Recognition and Applications. Prentice-Hall, Engelwood Cliffs, NJ.
55. JOACHIMS, T. 1998. Text categorization with support vector machines: Learning with many relevant features. In Proceedings of the 10th European Conference on Machine Learning (ECML, Heidelberg,
56. Germany). 137–142.
57. KUDO, T. AND MATSUMOTO, Y. 2001. Chunking with support vector machines. In Proceedings of NAACL(Pittsburgh, PA). 137–142.
58. COLLINS,M. 2004. Parameter estimation for statistical parsing models: Theory and practice of distributionfree methods. In New Developments in Parsing Technology,H. Bunt, J. Carroll, andG. Satta, Eds. Kluwer, Dordrecht, The Netherlands, 19–55.
59. ESCUDERO, G., M`ARQUEZ, L., AND RIGAU, G. 2000c. On the portability and tuning of supervised word sense disambiguation. In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC, Hong Kong, China). 172–180.
60. KEOK, L. Y. AND NG, H. T. 2002. An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. In Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP, Philadelphia, PA). 41–48.

AUTHORS PROFILE

J. Sreedhar is a Research Scholar, Department of Computer Science and Engineering, JNTUK, Kakinada, India. He has received M.Tech.(Computer Science and Technology) from Andhra University. He has been published and presented good number of Research and Technical papers in International Journals , International Conferences and National Conferences. Presently Working as Sr.Assoc Professor and HOD in the Department of CSE, ECET, Hyderabad. He published and presented good number of research papers in various conferences and journals. His main research interests are NLP, IRS, Data Mining, Compiler Design, FLAT, Computational Linguistics.

Dr. S. Viswanadha Raju obtained his Ph.D in Computer Science &Engineering from ANU. He obtained his M.Tech in CSE from JNTUniversity. He has a good academic background with a very sound academic and research experience. At present he is working as a professor and HOD-CSE in JNTUniversity, Jagityala,Hyderabad. He is guiding 10 research scholors for Ph.D and also conducted several conferences/workshops/seminars with sponsored agencies such as AICTE, DST, TCS, IEEE and CST. His research interests are Information Retrieval, Databases, Image Retrieval, Data Mining and related areas. He published 25 research papers in reputed International Journals/Conferences proceedings in his research area. He is active member in different professional bodies with life membership like IETE, ISTE and CSI.

Dr. A. Vinaya Babu,. obtained his Ph.D in Computer Science & Engineering from JNTU. He obtained his M.Tech in CSE from JNTUniversity. He obtained his ME from Osmania University. He obtained his BE in ECE from Osmania University He has a good academic background with a very sound and academic research experience. At present he is working as a professor and principal in JNTUniversity, Hyderabad. He is guiding 10 research scholors for Ph.D and also conducted several conferences/workshops/seminars with sponsored agencies such as AICTE, DST, TCS, IEEE and CST. His research includes Information Retrieval, Databases, Image Retrieval, Data Mining and related areas. He published 82 research papers in reputed International Journals/Conferences proceedings in his research area. He is active member in different professional bodies with life membership like IETE, ISTE and CSI.

Amjan.Shaik is a Research Scholar, Department of Computer Science and Engineering, JNTUH, Hyderabad, India. He has received M.Tech.(Computer Science and Technology) from Andhra University. He has been published and presented good number of Research and Technical papers in International Journals , International Conferences and National Conferences. His main research interests are Software Metrics, Software Engineering, Software Testing, Software Quality, Object Oriented Design and NLP.

Dr. Pammi Pavan Kumar, is working as Assistant Professor in the Department of Telugu at University of Hyderabad, India. He teaches Computational Linguistics in the department and guiding research in its functional aspects. He has published 19 books so far, as co-author, Editor and presented several papers in National and International conferences. He is a life member in the professional bodies like LSI, DLA.