MINOR PROJECT

WORD SENSE DISAMBIGUATION USING GRAPH THEORY

Submitted in partial fulfillment of
the requirements for the award of the

Degree of Bachelor of Technology
in
Information Technology

Submitted by

DTU/2K14/IT/013 Ayush Aggarwal
DTU/2K14/IT/016 Chhavi Sharma

Under the supervision of
Ms Minni Jain

Department of Information Technology
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Bawana Road, Delhi-110042 Winter Semester 2013

DECLARATION

I hereby certify that the work which is presented in the Minor Project titled ”Word Sense Disambguation using Graph Theory” in fulfillment for the award of the Degree of Bachelor of Technology and submitted to the Department of Information Technology, Delhi Technological University (formerly Delhi College of Engineering), New Delhi is an authentic record of my own, carried out during a period of January 2017- May 2017, under the supervision of Ms. Minni Jain.
The matter presented in this report has not been submitted by me for the award of any other degree of this or any other Institute/University.

Signature

Ayush Aggarwal DTU/2K14/IT/013
Chhavi Sharma DTU/2K14/IT/016

Date:

CERTIFICATE

This is to certify that **Ayush Aggarwal(DTU/2K14/IT/013)** and **Chhavi Sharma (DTU/2K14/IT/016)** the bonafide students of **Bachelor of Technology in Information Technology** of **Delhi Technological University** (formerly Delhi College of Engineering), New Delhi of **2014-2018 batch** have completed their major project titled **Word Sense Disambiguation using Graph Theory** under my supervision and guidance.

It is further certified that the work done in this dissertation is a result of candidates' own efforts.

Ms. Minni Jain

(Assistant Professor)

ACKNOWLEDGEMENTS

Firstly, we express our heartiest gratitude towards the authorities who gave us a chance to explore the intricacies of various aspects of Machine Learning. We are grateful to **Dr.Kapil Sharma**, HOD (Department of Information Technology), Delhi Technological University, Delhi and all other faculty members of our department for their astute guidance throughout the project.

We would also sincerely thank our esteemed mentor, **Ms.Minni Jain**, who lent a huge helping hand in the process of making this project with her valuable guidance and blessings.

In the end, we would like to thank our families for their extended support throughout the project.

ABSTRACT

Word sense disambiguation (WSD), the task of identifying the intended meanings (senses) of words in context, has been a long-standing research objective for natural language processing.

This report presents an unsupervised learning algorithm for sense disambiguation, specifically a graph based algorithm. We find the most important "node" in the sense graph which was constructed by using two words at a time from the sentence. We introduce a graph-based WSD algorithm which has few parameters and does not require sense-annotated data for training.

In the end, we also examine how the chosen lexicon and its connectivity influences WSD performance by calculating various local and global centrality measures.

CONTENTS

List of Figures

List of Tables

CHAPTER 1

Problem Definition

Word Sense Disambiguation has been the pinnacle of the evolution of Natural Language Processing. Basic Artificial Intelligence tests such as Turing Test[1]. It also serves as an understating work for Speech Recognition, Voice AI, Personal Assistant and numerous AI applications.

Given an input "I want to greet him with a diamond", to disambiguate it becomes a threefold problem. First we need to identify the tokens that give meaning to the sentence such as greet and diamond. The next step is to iterate through all the meanings of the given word using some lexicon such as Wordnet[2] and gather all possible senses to it. Then we need a Supervised or Unsupervised learning algorithm which can help disambiguate the collection of senses to just a pair of such meanings which satisfy the context the best.

Unsupervised Learning is chosen for our project because it does not require sense annotated data which has to be done manually, thus reducing a lot of setup time. Secondly, when chosen along with correct relation to draw parallels between the different tokens, these algorithms work as good as the supervised learning ones.

CHAPTER 2

Introduction

2.1 Background

Word Sense Disambiguation (WSD) is the process of identifying the sense of a polysemic[1] word.

In modern WSD systems, the senses of a word are typically taken from some specified dictionary. These days WordNet is the usual dictionary in question. WSD has been investigated in computational linguistics as a specific task for well over 40 years, though the acronym is newer. The SENSEVAL conferences have attempted to put Word Sense Disambiguation on an empirically measurable basis by hosting evaluations in which a given corpus of tagged word senses are created using WordNet's senses and participants attempt to recognize those senses after tuning their systems with a corpus of training data.

2.1.1 Recent Research

Recent advances in WSD have been due to the availability of the corpora annotated with word senses. Most accurate WSD systems to date exploit supervised methods which automatically learn cues useful for disambiguation from hand-labelled data.

2.1.2 Literature Survey

Measures of graph connectivity have been studied extensively in the social sciences, especially in the field of Social Network Analysis (SNA).

Due to the expansion of the World Wide Web, there has been development of link analysis algorithms for information retrieval. Among these, PageRank[3] and HITS[4] have been extremely influential.PageRank assigns a numerical weighting to each element of a hyper

[1] polysemic: association of one word with two or more distinct meanings

linked set of documents, with the purpose of measuring its relative importance with the set, whereas HITS rates web pages for their authority and hub values. Graph algorithms have also been applied to word sense induction[2]. For example, in the HyperLex algorithm [5], words are taken as the nodes of the graph and word co-occurrence represents an edge between two nodes. Detecting the different senses of a word thus amounts to isolating the high-density components in this co-occurrence graph. Although we focus here primarily on unsupervised methods, it is worth pointing out that graph algorithms such as Label Propagation have been successfully employed in supervised WSD. Beyond WSD, graph-based methods have been adopted in many NLP tasks such as summarizing, keyword extraction, sentiment analysis, sentence retrieval for question answering, ontology learning, human conversation analysis, and for estimating word dependency distributions.

2.2 Motivation

Despite the popularity of graph-based methods in NLP, there have been virtually no studies assessing how graph connectivity and the different ways of measuring it affects different tasks. A large number of graph connectivity metrics have been proposed within social network analysis and applied to different networks. Previous work has used almost exclusively two metrics, either variants of degree centrality or PageRank. This is in marked contrast with similarity-based approaches, where several studies have evaluated the effect of similarity measures on WSD performance Our inspiration stemmed from R. Navigli's work [6] where he has explored some of these issues, in a rather restricted setting. Specifically, they used the graph algorithm to build the sentence representations used to assess the performance of graph connectivity measures. Their algorithm builds the sense graph by consulting a hand-constructed grammar that determines which graph edge sequences are valid. After implementation of their algorithm, we realized that better results could be obtained. After researching on WSD, we discovered that better results could be obtained by certain modifications and implementing different libraries.

[2]word sense induction: the task of automatically inferring the senses of a given target word without recourse to a dictionary

CHAPTER 3

Work Done

3.1 Graph Based WSD

In order to isolate the impact of graph connectivity measures on WSD, we devised a fairly general disambiguation algorithm that has few parameters and relies almost exclusively on graph structure for inferring word senses. We use WordNet sense inventory[2] but neither our graph-based algorithm nor our connectivity measures are limited to this lexicon. Resources with alternative sense distinctions and structure have also serve as input to our method. In the following, we first provide a brief introduction to WordNet. Next, we describe our WSD algorithm and show our working example.

3.1.1 The Lexicon

WordNet[1] is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations. WordNet is also freely and publicly available for download. WordNet's structure makes it a useful tool for computational linguistics and natural language processing.

Structure

The main relation among words in WordNet is synonymy, as between the words shut and close or car and automobile. Synonyms–words that denote the same concept and are interchangeable in many contexts–are grouped into unordered sets - synsets.
Each of WordNets 117,000 synsets is linked to other synsets by means of a small number of conceptual relations. Additionally, a synset contains a brief definition (gloss) and, one or more short sentences illustrating the use of the synset members.

[1] WordNet is available from http://wordnet.princeton.edu

Relations

The most frequently encoded relation among synsets is the super-subordinate relation (also called hyperonymy, hyponymy or IS-A relation).

$$\text{furniture, piece of furniture} \rightarrow \text{bed}$$

It links more general synsets like furniture to increasingly specific ones like bed. Thus, WordNet states that the category furniture includes bed, conversely, concepts like bed make up the category furniture.

1. **Hyponymy** relation is transitive: If an armchair is a kind of chair, and if a chair is a kind of furniture, then an armchair is a kind of furniture. WordNet distinguishes among Types (common nouns) and Instances (specific persons, countries and geographic entities).

2. **Meronymy**, the part-whole relation holds between synsets like chair and back, backrest, seat and leg. Parts are inherited from their super ordinates: if a chair has legs, then an armchair has legs as well.

3. **Troponyms** are arranged into hierarchies as well: verbs towards the bottom of the trees express increasingly specific manners characterizing an event. The specific manner expressed depends on the semantic field.

4. **Antonymy** describes a relation of opposites. Pairs of direct antonyms like wet-dry and young-old reflect the strong semantic contract of their members. Each of these polar adjectives in turn is linked to a number of semantically similar ones.

Cross-POS relations The majority of the WordNets relations connect words from the same part of speech (POS). Thus, WordNet really consists of four sub-nets, one each for nouns, verbs, adjectives and adverbs, with few cross-POS pointers. Cross-POS relations include the morphosemantic links that hold among semantically similar words sharing a stem with the same meaning.

3.1.2 WSD Algorithm

Our disambiguation algorithm proceeds incrementally on a sentence. Initially, we build a graph $G = (V, E)$ for each target sentence which we induce from the graph of the reference lexicon as show in Figure 3.1

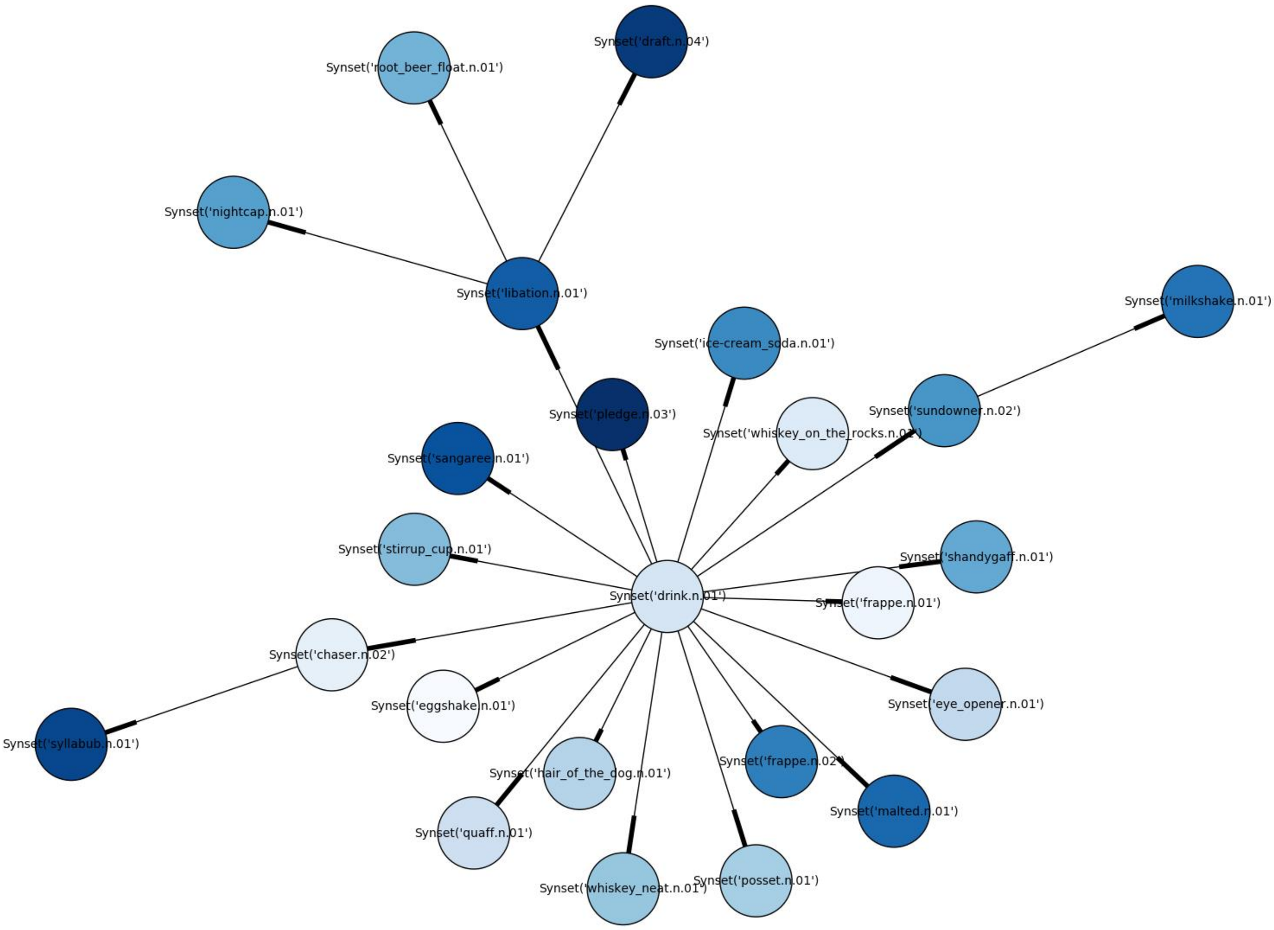

Figure 3.1: Graph extracted from Word net using hyponymy

Our Algorithm takes input a sentence whose sense is to be disambiguate. It then uses a keyword extraction algorithm to extract content words only (i.e., nouns, verbs, adjectives, and adverbs). As explained above, nodes in the graph are word senses and the edges the semantic relations between them.

We then select a sense u ϵ senses and execute the following algorithm on it :-

- Generate sets of three relations , hypernymy meronymy and closure for the sense u

- We perform a Depth First Search for all the words v in those sets. Every time we encounter a node belong to destination along the path , we add the path to the senses and plot it on the graph.

In DFS, edges are explored out of the most recently discovered vertex v that still has unexplored edges leaving it. When all of vs edges have been explored, the search backtracks to explore edges leaving the vertex from which v was discovered. This process continues until we have discovered all the vertices that are reachable from the original source vertex.

Step 1:- A query Q is entered in the form of an English sentence.

Step 2:- **Tokens** , the most important word phrases are extracted using **Rake Keyword Extractor** algorithm

Step 3:- Pick two words from the set of generated keywords and generate a subgraph G using **Wordnet**

To create a subgraph G
Initialise the Graph G to null
Create a set of synsets of both the keywords and add all those nodes to the graph and call them source and destination sets
Initialise a set of words/nodes that have been discovered yet
For every element x in syset of source:
Generate 3 sets of relations- **hypernymy** , **hyponymy** and **closure**
DFS the three subsets
(i) If the DFS length is greater than K , return
(ii) If the DFS discovered a node that was in the destination set , then
- Add all the nodes of the current path to the graph
- Add all the edges between all possible pair of the nodes
- Add all the nodes in the current path to the destination set

else follow through the DFS by repeating the process

Step 4:- For each Sub Graph G , calculate the **Local Centrality Measures** such as
- Degree Centrality
- Betweenness
- Hits
- Page Rank
- Closeness

Pick out the best source and destination context using all the above mentioned methods and pick out the ones

Step 5:- For each Sub Graph G , calculate the **Global Centrality Measures** such as
- Entropy
- Compactness
- Edge Density

Use the global measures to weightage the chosen source and destination context and find out the final two source and destination synsets

Figure 3.2: WSD Algorithm

After the extraction of Graph G from Wordnet , we apply various local centrality measures to extract the best possible meaning of the given two keywords. Global Centrality measures are applied to all such graph which gives us the importance of that graph to disambiguate the meanings of the words they represent.

The disambiguation algorithm presented above has a limited notion of context, and only neighboring words within the same sentence contribute to the meaning of an ambiguous word. In our case, the senses of a word can vary across sentences and documents. There is nothing inherent in our algorithm that restricts us to sentences. We could just as well build and disambiguate a graph for a document.

We consider an example sentence- "She drank some Milk from the bottle" . The keyword extractor returns the tokens ["drink" "milk" "bottle"]. The graph extracted after running it on the pair (drink, milk) is:-

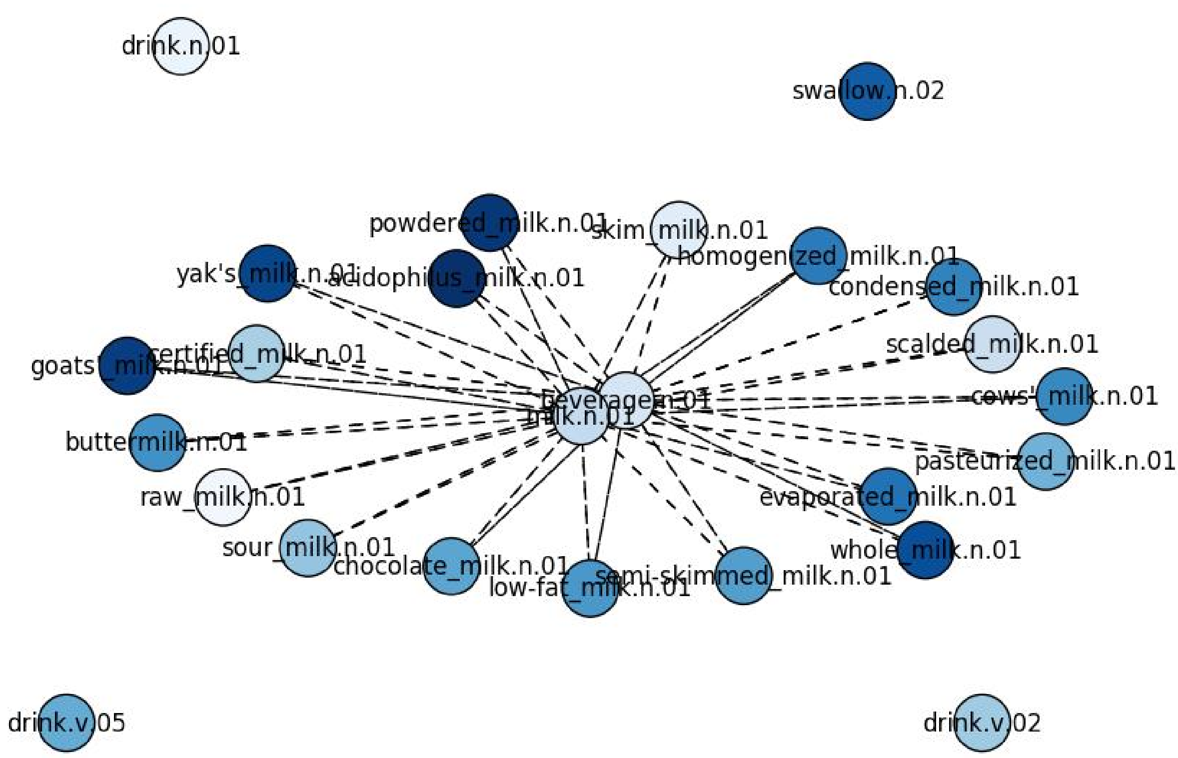

Figure 3.3: Graph extracted from keywords 'drink' and 'milk'

3.2 Centrality Measures

There are certain local measures and global, which determine the degree of relevance of a vertex v in graph G and the influence of a node over the network. They are helpful in determining the graph connectivity and can be used for both directed and undirected graphs.

We first introduce the distance function $d(u, v)$ which is used by some of the measures discussed below:

$$d(u,v) = \begin{cases} \text{length of shortest path} & \text{if } u \to v \\ K & otherwise \end{cases}$$

where $u \to v$ denotes that a path exist between u and v and K is a conversion constant which replaces ∞.

3.2.1 Local Measures

Local measures of graph connectivity determine the degree of relevance of a single vertex v in a graph G. They can thus be viewed as measures of the influence of a node over the network.Formally, we define a local measure l as:

$$l : V \to [0, 1]$$

A value close to one indicates that a vertex is important,whereas a value close to zero indicates that the vertex is not influential.In the following report, we consider three best known measures of centrality, namely, degree, closeness, and betweenness[7]

Degree Centrality

It is the simplest way to determine a vertex importance by its degree[8]. The degree of a vertex refers to the number of edges incident on that vertex. For an undirected graph, the number of outgoing edges and number of incoming edges are same; i.e., in-degree is equal to out-degree. However, for directed graphs it is different. The degree of a vertex is given by:

$$deg(v) = |(u, v) \in E : u \in V|$$

If a vertex is present in the center of the graph, it has high degree. The degree centrality is the degree of a vertex normalized by the maximum degree and calculated as:

$$C_d(v) = \frac{deg(v)}{|V| - 1}$$

Betweennes Centrality

It is defined in terms of how "in between" a vertex is among the other vertices in the graph. The betweenness centrality of a node 'v' is the ratio of the shortest paths from one node to another node that are passing from 'v' and the number of shortest paths between two nodes.

$$Betweenness(v) = \sum \frac{\sigma_{i,j}(v)}{\sigma_{i,j}}$$

Closeness

Closeness or Key Player Problem(KPP) considers the importance of a vertex by its relative closeness with all the other vertices[10]. It is calculated as reciprocal of total shortest distance from a given node to all other nodes.It is calculated as:

$$KPP(v) = \frac{\sum_{u \in V: u \neq v} \frac{1}{d(u,v)}}{|V| - 1}$$

where, the inverse of the shortest distance between v and all other nodes is the numerator, and denominator is the nodes in the graph.

PageRank

PageRank is one of the popular algorithms to rank the nodes or find the importance of a node in a network[11]. The vertex with highest number of votes casted will have higher importance or relevance. Moreover, the importance of vertex casting a vote determines how important a vote is. All the nodes that link to v contribute towards determining its relevance.

$$PageRank(v) = \frac{1-d}{|V|} + d \sum_{u,v \in E} \frac{PR(u)}{outdegree(u)}$$

where d is the damping factor introduced, which has therole of integrating into the model the probability of jumping given vertex to another random vertex and its value is set between zero and 1. A value for zero means that the ranking of the page does not depend on its outgoing links, and 1 indicates that the score is exclusively determined by the links with neighboring pages. The typical value of d is 0.85.

Hypertext Induced Topic Selection

HITS is similar to PageRank but the only main difference is, it makes a distinction between authority and hubs; i.e, in this method two values are determined for a node v i.e., authority (a(v)) and hub value (h(v)). The authority corresponds to the pages that are good and reliable sources and have numerous incoming links, whereas hub value corresponds to the pages having many outgoing links[12].

For every vertex, HITS produce two set of scores: authority score and hub score.

$$HITSA(Vi) = \sum_{V_j \in In(Vi)} HITS_H(Vj)$$

$$HITSH(Vi) = \sum_{V_j \in Out(Vi)} HITS_HA(Vj)$$

For each iteration, these scores are normalized, so that the authority scores for all vertices add up to 1. HITS can also be applied to undirected graphs.

3.2.2 Global Measures

Global connectivity measures are concerned with the structure of the graph as a whole rather than with individual nodes. Here, we discuss three well-known measures, namely, compactness, graph entropy, and edge density.

Graph Entropy

Entropy measures the amount of information (or, alternatively, uncertainty) in a random variable. In graph-theoretic terms, high entropy indicates that many vertices are equally important, whereas low entropy indicates that only a few vertices are relevant. We define a simple measure of graph entropy as:

$$H(G) = -\sum_{u \in V} p(v) log(p(v))$$

where the vertex probability $p(v)$ is determined by the degree distribution $\frac{deg(v)}{2|E|}_{(v \in V)}$.

Compactness

This measure represents the extent of cross referencing in a graph [18]: When compactness is high, each vertex can be easily reached from other vertices. The measure is defined as:

$$CO(G) = \frac{Max - \sum_{u \in V} \sum_{v \in V} d(u,v)}{Max - Min}$$

where $Max = K|V|(|V| - 1)$ is the maximum value the distance sum can assume (for a completely disconnected graph) and $Min = |V|(|V| - 1)$ the minimum value (for a fully connected graph). $CO(G)$ is zero when G is completely disconnected and one when G is a complete graph.

Edge Density

Finally, we propose the use of edge density as a simple global connectivity measure. Edge density is calculated as the ratio of edges in a graph to the number of edges of a complete graph with $|V|$ vertices (given by $\frac{|V|}{2}$). Formally,

$$ED(G) = \frac{|E(G)|}{\frac{|V|}{2}}$$

where ED(G) has a [0,1] range, with 0 corresponding to a totally disconnected graph and 1 to a complete graph.

3.3 Experimental Results

Our results for WSD using WordNet are summarized in Table 3.1 for local measures and Table 3.2 for global measures. The local measures show the best nodes of a graph whereas the global measures depict the best graph out of all. Using the global measures, we give the weight-age to the best graph

The degree centrality gives the most consistent output for all nodes and hence does not help much in choosing the best node as all have equal values. HITS performs worse as compared to KPP or closeness. PageRank performs the best. This is not entirely surprising; the PageRank value of a node is proportional to its degree in undirected graphs.

Among the global measures, Compactness and Edge Density are significantly better than Graph Entropy ($p \leq 0 : 01$).

In the above-stated example , our example returns the result with **beverage n.01** and **milk n.01** as the context of drink and milk in the sentence.

Graph Name	Entropy	Edge Density	Compactness
(drink, milk)	0.0371	0.0785	0.3255
(milk, bottle)	0.5270	0.0785	0.6255
(drink, bottle)	0.2321	0.0035	0.2565

Table 3.1: Global Centrality Measures for 'She drink some milk from the bottle'

Node	Degree	Betweenness	Hits	PageRank	Closeness
drink.n.02	0.0	0.0	0.0	0.006	0.0
raw milk.n.01	0.171	0.0	0.0	0.028	0.300
drink.n.01	0.0	0.0	0.038	0.006	0.0
skim milk.n.01	0.171	0.0	0.038	0.006	0.300
toast.v.02	0.0	0.0	0.0	0.006	0.0
beverage.n.01	1.2	0.143	0.131	0.17	0.57
sour milk.n.01	0.171	0.0	0.038	0.028	0.300
milk.n.01	1.2	0.143	0.131	0.176	0.57
certified milk.n.01	0.171	0.0	0.130	0.028	0.300
drink.v.02	0.0	0.0	0.130	0.006	0.0
scalded milk.n.01	0.171	0.0	0.38	0.028	0.300
pasteurized milk.n.01	0.171	0.13	0.028	0.028	0.300
chocolate milk.n.01	0.171	0.0	0.038	0.028	0.300
semi-skimmed milk.n.01	0.171	0.0	0.130	0.028	0.300
low-fat milk.n.01	0.171	0.0	0.130	0.028	0.300
buttermilk.n.01	0.171	0.0	0.0	0.028	0.300
cows' milk.n.01	0.171	0.0	0.0	0.028	0.300
condensed milk.n.01	0.171	0.0	0.0	0.028	0.300
homogenized milk.n.01	0.171	0.0	0.0	0.028	0.300
evaporated milk.n.01	0.171	0.0	0.038	0.028	0.300
whole milk.n.01	0.171	0.0	0.037	0.028	0.300
yak's milk.n.01	0.171	0.0	0.038	0.028	0.300
goats' milk.n.01	0.171	0.0	0.038	0.028	0.300
powdered milk.n.01	0.171	0.0	03.038	0.028	0.300
acidophilus milk.n.01	0.171	0.0	0.038	0.028	0.300

Table 3.2: Local Centrality Measures for drink and milk

CHAPTER 4

Future Work

Graph based WSD opens a possibility of applying WSD algorithm for which handpicked data such as WordNet is not that extensive as WordNet is. Since we use only the basic relations , still able to achieve comparative results as compared to the sense-annonated Supervised algorithms , they can be used in different lexicons such as Hindi Wordnet to further extend it's reach.

Further , WSD is the under the hood algorithm for various AI technqiues and improvement to this can cause a huge improvement in those systems. Graph Based tehcnique can not only be extended to Word Sense Disambguation but to other fields of NLP such as text summarization and Query Expansion and Keyword Extraction.

Using Naive Bayesian approach in addition to Graph Algorithms, can prove a mettle in applications such as text prediction and suggestions.

CHAPTER 5

Conclusion

In this project, we presented a study of graph connectivity measures for unsupervised WSD. We evaluated a wide range of local and global measures with the aim of isolating those that are particularly suited for this task. Our results indicate that local measures yield better performance than global ones. The best local measures are Degree and PageRank.
We also find that the employed reference dictionary critically influences WSD performance. We obtain a large improvement (in the range of 10 percent) when adopting a version of WordNet enriched with thousands of relatedness edges. This indicates that graph-based WSD algorithms will perform better with more densely connected sense inven- tories, with more incident edges for every node. Our experiments show that the performance could potentially increase when the right connectivity measure is chosen. The proposed measures are independent of the adopted reference lexicon; they induce a sense ranking solely by considering graph connectivity, and can thus be ported across algorithms, languages, and sense inventories.

Our experiments focused primarily on graph connectivity measures and their suitability for WordNet-like sense inventories. For this reason, we employed a relatively generic WSD algorithm without extensive tuning and obtained state-of-the-art performance when assessing our system on standard evaluation data sets. More research is needed to assess whether our results extend to other NLP tasks, besides WSD. An obvious application would be summarization, where graph-based methods have met with reasonable success and eigenvector centrality measures are a popular choice.
However, their performance against other graph connectivity measures has not yet been studied in detail.

Bibliography

[1] Bieri, P. (1988), "Thinking Machines: Some Reflections on the Turing Test ", Poetics Today 9(1), pp. 163-186.

[2] C. Fellbaum, ed. MIT Press, 1998, "WordNet: An Electronic Lexical Database"

[3] S. Brin and M. Page, "Anatomy of a Large-Scale Hypertextual Web Search Engine", Proc. Seventh Conf. World Wide Web, pp. 107- 117, 1998.

[4] J.M. Kleinberg, "Authoritative Sources in a Hyperlinked Environment, " Proc. Ninth Symp. Discrete Algorithms, pp. 668-677, 1998.

[5] J. Vronis, "Hyperlex: Lexical Cartography for Information Retrieval "Computer, Speech and Language, vol. 18, no. 3, pp. 223-252, 2004.

[6] Roberto Navigli and Mirella Lapata, "An Experimental Study of Graph Connectivity for Unsupervised Word Sense Disambiguation"

[7] L.C. Freeman, "Centrality in Social Networks: Conceptual Clarification " Social Networks, vol. 1, pp. 215-239, 1979.

[8] Navigli, R., Lapata, M.: An experimental study of graph connectiv- ity for unsupervised word sense disambiguation, IEEE transaction on pattern analysis and machine learning, Vol. 32 No. 4, April (2010).

[9] Sinha, R., Mihalcea, R.: Unsupervised graph-based word sense disambiguation using measures of word semantic similarity. In: Proceedings of ICSC (2007)

[10] Borgatti, S.P.: Identifying Sets of Key Players in a Network. In: Proceedings Conference Integration of Knowledge Intensive MultiAgent Systems, pp. 127131 (2003)

[11] Brin, S., Page, M.: Anatomy of a Large-Scale Hypertextual Web Search Engine. In: Proceedings Seventh Conference World Wide Web, pp. 107117 (1998)

[12] Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. In: Proceedings Ninth Symposium Discrete Algorithms, pp.668677 (1998)