STUDY OF WORD NETWORKS

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
OF INDIAN INSTITUTE OF TECHNOLOGY, KHARAGPUR
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF TECHNOLOGY

Joy Deep Nath

May 2008

I certify that the thesis titled *Study of Word Networks* submitted by *Joy Deep Nath, Roll. no. 03CS3021* to the partial fulfillment of the requirements of the Degree of Masters of Technology in Computer Science and Engineering is a bonafide record of work carried out by him under my supervision and guidance. I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Master of Technology.

(Dr. Niloy Ganguly) Masters Project Adviser
Dept. of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Approved for the University Committee on Graduate Studies.

Abstract

A study of the word interaction networks of Bengali in the framework of complex networks is first done. The topological properties of these networks reveal interesting insights into the morpho-syntax of the language, whereas clustering helps in the induction of the natural word classes leading to a principled way of designing POS tagsets. Then, different network construction techniques and clustering algorithms based on the cohesiveness of the word clusters measured against two gold-standard tagsets by means of the novel metric of *tag-entropy*. The approach is then extended to any five other languages- English, German, Hindi, Hebrew and Finnish to find their word network properties. Since the clusters on manual inspection reveal word classes hinting at morpho-syntax, we propose a framework for the named entity recognition (NER) and a creation of semi-supervised NER engine. And finally, we create a word network from part-of-speech (POS) tagged corpus of 2 languages to study the network structure and verify the POS induction in word networks.

Acknowledgments

I would like to profusely thank Monojit Choudhury for being there with me at all stages. For Monojit Da's his guidance, support and inspiration, I am greatly indebted. I would like to thank Animesh Mukherjee for his continued support and valuable inputs at various stages. I would like to thank Chris Beimann for helping with ideas and program codes in the beginning. And last, but not the least, I would like to thank my Adviser Prof. Niloy Ganguly, who not only inspired me throughout and provided crucial inputs in all the stages. I consider myself lucky to have his unfailing support, not only in this project but throughout my stay in Kharagpur, as my association with him goes beyond the realms of the project.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Parts-of-speech (POS, also known as *word class* or *lexical category*) are the linguistic categories of words defined by their morphological and syntactic properties. The word categories that are distinctive in one language may feature identical behavior in another language. Linguists identify the lexical categories through a manual inspection of the morpho-syntactic patterns present in a language. Can there be a principled and computational approach to this problem of identification of the lexical categories? The answer turns out to be 'yes', thanks to the concept of "distributional hypothesis" [21].

In fact, this hypothesis is the underlying (implicit or explicit) assumption of all computational approaches to POS tagging which is a very important preprocessing task for several NLP applications. Ironically, compared to the work done in the area of POS tagging, the volume of research dedicated to POS tagset (i.e., the set of lexical categories) design is quite small, even though the tagset is largely responsible for the efficiency as well as the effectiveness of a POS tagger.

The two basic questions that need to be answered while designing a POS tagset are: (a) which lexical categories are distinguishable in a language? and (b) does making a distinction between two categories help us in further NLP applications such as chunking and parsing? In other words, a tagset is always dependent on the language under consideration as well as the end application to which the POS-tagger caters. In fact, often the natural word classes present in a language are those that are easy to distinguish as well as sufficient in facilitating deeper linguistic processing. A key to the identification of these natural word classes is to understand the syntactic structure of a language, which is captured through the complex interaction of the words. This is arguably an outcome of a self-organizing process governing the dynamics of language and grounded in the cognitive abilities of human beings [29]. In this context, language can be viewed as a network of words and formation of lexical categories an emergent property of this network. Thus, understanding the structure and function of this network will help us in procuring deeper insight into the nature of word classes in a given language.

In this work, we present a study of the lexical classes of Bengali obtained through the analysis of the word interaction networks. Although the scheme presented here is not essentially novel and has been motivated by several work on unsupervised induction of POS based on the distributional hypothesis [16, 26, 27, 18, 4, 25, 3], our main contributions reside in – (a) a comparative study of various approaches to POS tagset induction on Bengali, (b) rigorous linguistic analysis of the word classes and suggestions for a Bengali tagset design, (c) introduction of a novel metric, called tag entropy, to evaluate the goodness of the induced word classes, and most importantly, (d) analysis of the word interaction networks within the framework of complex network theory to understand the syntactic structure of Bengali. The analytical scheme presented here is a generic one and can be readily applied to any language for which a raw text corpus of moderate size is available.

Sec. 1.1 puts this work in the context of previous research in the areas of unsupervised POS induction and complex network theory. In Sec. 2.1 we define the word interaction networks and analyze their topological properties. Sec. 3.1 introduces the POS tagset induction models, experimental settings and the metric of tag-entropy, followed by a quantitative comparison of the results obtained from these models. In Sec. 3.3 we present a linguistic analysis of the induced word classes. Sec. 6 concludes the paper by summarizing our observations.

1.1 Background

The present work is based on two different lines of research. On one hand, from the perspective of NLP applications, it is based on the use of unsupervised machine learning techniques for induction of POS categories, and on the other hand it models and analyzes the syntactic distribution of the words in the framework of complex networks. In this section, we present a brief survey of both these research areas.

1.1.1 Unsupervised Induction of POS

Unsupervised induction of syntactic categories or POS tags involves use of machine learning techniques to automatically cluster the words of a given raw text corpus into syntactic classes. The formation of syntactic classes can be governed by providing a seed lexicon or can be left at the discretion of the learning algorithm. These techniques, especially the latter ones, help us to (a) create a partial tagging dictionary[1] automatically from a raw text corpus, which can then be used for developing a POS tagger, and (b) acquire important insights into the natural syntactic classes present in a language, which in turn helps in deciding on a tagset for the language.

There are a number of approaches to derive syntactic categories. All of them employ a syntactic version of Harris' distributional hypothesis [21], which states that words of similar parts of speech can be observed in the same syntactic contexts. Since the function words form the syntactic skeleton of a language and almost exclusively contribute to the most frequent words in a corpus, contexts in that sense are often restricted to the most frequent words [22]. The words used to describe syntactic contexts are further called *feature words*[2].

The general methodology [16, 26, 27, 18, 4, 25, 3] for inducing word class information can be outlined as follows, (a) collect global context vectors of target words by counting how often feature words appear in the neighboring positions, and, (b) apply a clustering algorithm on these vectors to obtain word classes.

Throughout, feature words are the most frequent 50-250 words. Some authors employ a much larger number of features and reduce the dimensions of the resulting matrix using Singular Value Decomposition [26, 25]. The choice of high frequency words as features is motivated by Zipf's law: these few stop words constitute the bulk of the tokens in a corpus.

[1] A tagging dictionary consists of the distinct words (possibly inflected) of a language and their corresponding POS tags.

[2] *Target words*, as opposed to this, are the words that are to be grouped into syntactic clusters. Note that usually, the feature words form a subset of the target words.

Contexts are the feature words appearing in the immediate neighborhood of a word. The word's global context is the sum of all its contexts. Clustering consists of a similarity measure and a clustering algorithm. [16] uses the Spearman Rank Correlation Coefficient and a hierarchical clustering, [26, 27] use the cosine of the angle between the vectors and Buckshot clustering, [18] uses cosine on Mutual Information vectors for hierarchical agglomerative clustering and [4] applies Kullback-Leibler divergence.

Slightly different variations of the above generic scheme can be found in [5], [17] and [9]. For small size raw corpora, Bayesian approaches are known to be capable of producing good results [20, 19]. However, these approaches rely on a predefined set of tags and a small annotated corpus or a partial lexicon. A further related work is [8], which proposes an unsupervised morphological analysis to create a soft clustering on word classes in their weakly supervised word class induction system for English and Bengali.

1.1.2 Syntax as a Self-organizing Phenomenon

Recently, there has been several studies on the structural patterns of human languages within the framework of *complex network theory* (see [23] for a review). A complex network is a collection of entities (represented as nodes) and their interactions (represented as links or edges between the nodes). Such networks have been successfully used to explain the structure, function and evolutionary dynamics of a variety of natural systems found in the domains of biology, economics, physics, social sciences and information sciences. See [6] for a survey on applications of networks in various areas. In the context of syntax, studies on *word collocation networks* and *syntactic dependency networks* have revealed several interesting cross-linguistic universals and their possible explanations in terms of human cognition.

In word collocation networks, words are the nodes and two words are linked if they are neighbors, that is they collocate, in a sentence [14, 13]. Such networks, constructed for various languages, have been found to exhibit small world properties. The average path length between any two nodes is small (around 2 to 3) and the clustering coefficients are high (around 0.69). However, the most striking observation regarding these networks is that the degree distributions follow a two regime power-law. The degree distribution of the 5000 most connected words follow a power-law with an exponent -3.07, which is surprisingly close to that of the Barabási-Albert preferential attachment based growth model [1]. These findings led the authors to argue that the word usage of the human languages is preferential in nature, where the frequency of a word defines the comprehensibility and production capability. In essence, the authors conclude that evolution of language has resulted in an optimal structure of the word interactions that facilitate easier and faster production, perception and navigation of the words.

Although collocation networks are easier to construct, they do not necessarily capture the syntactic and semantic relationships between the words. This is because syntactic and semantic relations often extend beyond the local neighborhood of a word. Ferrer-i-Cancho and his co-authors [11, 15] defined the *syntactic dependency network* (SDN) where the words are the nodes and there is a directed edge between two words if in any of the sentences of a given corpus there is a directed dependency relation between them. The SDNs were constructed from the dependency treebanks for three languages: Czech, German and Romanian, and found to exhibit strikingly similar characteristics. All the networks exhibit power-law degree distributions, small world structure, disassortative mixing and a hierarchical organization. More recently, [12] showed that spectral clustering of SDN puts words belonging to the same syntactic categories in the same cluster.

Thus, word collocation as well as the syntactic dependency networks unfurl various interesting facts about the nature of word interactions and syntactic patterns.

Chapter 2

Word Networks

2.1 Word Networks

The definition and the construction of the word networks presented here are primarily based on the work by [3]. Nevertheless, we also explore some variations while defining the network as well as their construction for Bengali data. Moreover, we study the topological properties of these networks, which provides us with insights into the syntactic structure of Bengali. We also conduct a comparative study of two different clustering algorithms.

2.1.1 Feature words, Context Vectors and Similarity Metric

We take a raw Bengali text corpus consisting of n tokens and compute the unigram frequency counts for each of the types observed in the corpus. We select the first m types that have the highest unigram frequencies as the *feature words*. The intuition is that since the function words have a very high frequency, the feature words selected on the basis of frequency will largely correspond to the function words of the language.

However, we observe that for corpora pertaining to specific domains (e.g., only news articles), several content words also creep into the list of top few words deemed here as feature words. Therefore, to ensure the absence of any content word in the set of feature words, we also construct networks where the this set is manually selected from a frequency-based sorted list of words. We shall refer to the former (i.e., frequency based feature word selection) networks by a prefixed superscript *fr* and the latter networks by another prefixed superscript *ms*.

Let $w_{-2}w_{-1}ww_1w_2$ be a window of 5 tokens around the target word w. A *context vector* for the target word w is defined as a vector of dimension $4m$ in which the entries $(4i+1), (4i+2), (4i+3)$ and $(4i+4)$ correspond to the number of occurrences of the $(i-1)$th feature word at the w_{-2}, w_{-1}, w_1 and w_2 positions respectively.

In [3], the distributional similarity between two words w and v is defined as $sim(w,v) = \frac{1}{1-cos(\vec{w},\vec{v})}$, where $\vec{w}$ and $\vec{v}$ represent the context vectors of the words w and v computed from a large raw text corpus; $cos(\vec{x},\vec{y})$ is the normalized dot product of the vectors $\vec{x}$ and $\vec{y}$, i.e., the cosine of the angle between them. An alternative definition of the similarity could be simply the cosine of the angle between $\vec{w}$ and $\vec{v}$, that is $sim(w,v) = cos(\vec{w},\vec{v})$. We shall denote the networks constructed using the metric proposed in [3] by a prefixed superscript *b* (for Biemann) and the latter ones by another prefixed superscript c (for cosine).

2.1.2 Definition and Construction of the Networks

The word network is a weighted undirected graph $G = \langle V, E \rangle$, where V consists of 5000 nodes corresponding to the most frequent 5000 types excluding the feature words. The number of nodes in V has been decided based on the fact that with a corpus of size around 10M words, enough context information is available only for the top few words. The weight of the edge between any two nodes representative of the words w and v is given by $sim(w,v)$ and this edge exists if $sim(w,v)$ exceeds a threshold τ. Thus, considering all the variations in definition of feature words and similarity metric, we can construct four different networks for a given corpus: ${}^{fr,b}G$, ${}^{fr,c}G$, ${}^{ms,b}G$ and ${}^{ms,c}G$.

Figure 2.1 presents a hypothetical illustration of the word network.

We have used the newspaper corpus[1] Ananda Bazaar Patrika for the creation of word networks. This corpus has around 17M words. We shall represent a network constructed from a corpus of size n using m feature words as $G_{n,m}$. Therefore, for a frequency-based selection of feature words and cosine similarity metric, the networks will be denoted as

[1] The authors thank ISI Kolkata for providing this corpora for the purpose of the experiments.

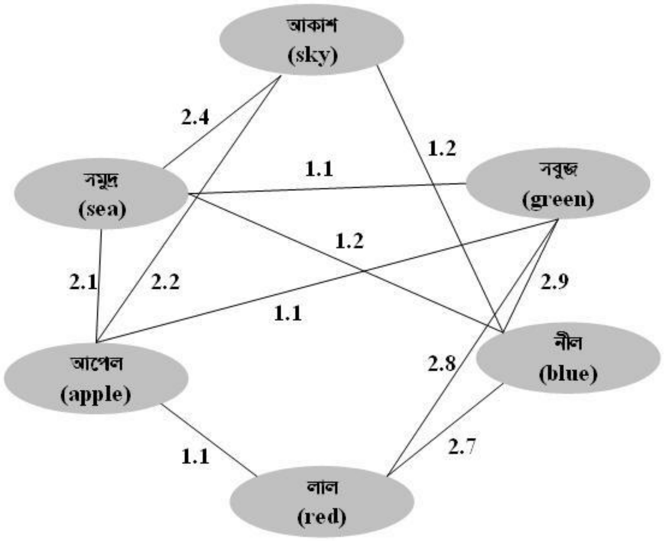

Figure 2.1: A hypothetical illustration of the word network. The English gloss for each of the Bengali words is provided within parentheses. Note that the edge weights are hypothetical and do not correspond to any of the similarity metrics.

${}^{fr,c}G_{n,m}$. Also, we shall drop the superscripts or subscripts whenever we refer to the networks corresponding to all the combinations for the part dropped.

We construct 20 word networks for all possible combinations of $n = \{$ 1M, 2M, 5M, 10M, 17M $\}$ and $m = \{25, 50, 100, 200\}$. In order to construct $G_{n,m}$ for $n < 17$, we have randomly selected a subset of documents from the original corpus. Note that in our experiments we consider the different inflected forms of a root morpheme as different types.

2.2 Properties of the Word Networks

In this section we present some of the important topological properties of the word networks. Interestingly, the four basic variations in network construction give rise to networks that have very similar topological properties. Therefore, we shall present all the results for ${}^{fr,b}G_{n,m}$, which might be generalized to the other cases as well. Note that the calculation of the degree distribution and the clustering coefficient is done on the unweighted version of the networks (all edges below the threshold τ are deleted).

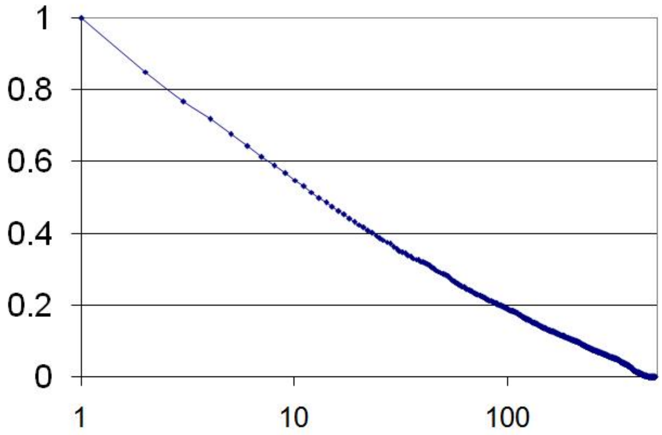

Figure 2.2: Cumulative Degree Distribution for the word network ${}^{fr,b}G_{17M,50}$. x-axis: $\log(k)$, y-axis: P_k

2.2.1 Degree Distribution

The cumulative degree distribution (CDD) of a network, P_k, is the probability that a randomly chosen node has degree greater than or equal to k. CDD provides important information about the topology of the network. Figure 2.2 shows the CDD for the word network ${}^{fr,b}G_{17M,50}$. We observe that the CDD follows a logarithmic distribution (i.e., $P_k \propto log(k)$), which means that $-\frac{dP_k}{dk} = p_k$ (probability that a randomly chosen node has degree equal to k) or the non-cumulative degree distribution is proportional to k^{-1} (popularly known as power-law or Zipfian distribution, but it is not clear whether this is a consequence of Zipf's law). Similar results have been observed for the networks with varying m and n.

Power-law networks are believed to have a self-similar hierarchical structure. In this case, the hierarchy is a reflection of syntactic ambiguities. Highly ambiguous words that belong to several lexical categories have the highest degrees. The next level of hierarchy is manifested by words that belong to a few lexical categories, whereas the last level of hierarchy is represented by the words that are unambiguous in nature. The power-law indicates that there are few words that belong to a large number of lexical categories, while the most of the words belong to only one lexical category.

2.2.2 Clustering Coefficient

The clustering coefficient of a node is the probability that a randomly chosen pair of its neighbors are themselves neighbors. We observe that there is a positive correlation between the degree of a node and its clustering coefficient. In particular, high degree nodes (i.e., the most ambiguous ones) have a high clustering coefficient. This implies that the network is very dense (clique-ish) around the high degree nodes. As we shall see later, this has a significant effect on the cluster size distribution and the efficacy of this method as such. The mean clustering coefficient for ${}^{fr,b}G_{17M,50}$ is 0.53, which is much higher than that of random graphs. This again points to the fact that there is a strong community structure in the networks reflecting the presence of natural word classes.

2.3 Community Structure

In order to gain insight into the topology of the network we cluster them using the following two different approaches.

Chinese Whispers: The Chinese Whispers (CW) algorithm [2] is a non-parametric random-walk based clustering algorithm, where initially each node is in a separate cluster. In every iteration, the nodes propagate information about their current cluster to all the neighbors, and in turn, decide upon their own cluster labels based on a weighted majority voting of the cluster information received from the neighbors. The algorithm terminates when the labels do not change considerably over successive iterations.

Agglomerative Hierarchical Clustering: In this approach [25], initially all the words are in separate clusters. At every iteration, two clusters closest to each other (where "closeness" between the centroids of the two clusters is measured by $sim(w, v)$) are merged to form a new cluster. The algorithm terminates after obtaining a predefined number of clusters.

We plot the cluster size distributions for ${}^{fr,b}G$ in Fig. 2.3 for various values of n and m following the CW algorithm. In fact, the distributions are identical for both the clustering approaches and all the other networks. The cluster size distributions (CSD) show a power-law behavior, which gets better as n increases. Thus, there are a few giant clusters, as is expected from the presence of the nodes with high degree and high clustering coefficient

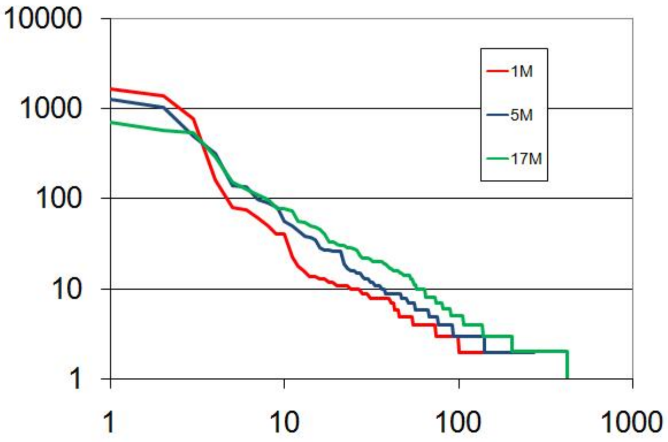

Figure 2.3: Rank (x-axis) versus cluster size (y-axis) in doubly logarithmic scale for ${}^{fr,b}G_{n,50}$ where n is 1M, 5M and 17M. The clusters are assigned a rank in descending order of their size (i.e. the number of words in the cluster), so that the largest cluster gets rank 1.

in the networks. Thus, the giant clusters consist of words that belong to multiple POS categories. In fact, these are the words that make POS tagging a non-trivial and challenging task. It would be interesting to devise techniques that can break the giant component into further clusters. We also observe that the words belonging to the giant clusters need not have high frequency in the corpus.

In this section, we have analyzed the word networks from a complex network perspective, which has revealed several significant properties underlying the syntactic structure of Bengali. We shall revisit these issues in Sec. 6, but before that we shall analyze the word clusters from the perspectives of NLP and linguistics in general.

2.4 Other word networks

The generic analytical framework is then used on five more languages viz. English, Finnish, German, Hebrew and Hindi to obtain the degree distribution, clustering coefficient of these languages. The clustering coefficients the six languages are shown in Table 2.1. Also, the cumulative degree distributions of the word netwroks for the six languages, it is evident from the values that word networks share similar structure. And hence, the framework is

Language	Corpus Size (in sentences)	Clustering Coefficient
Bengali	0.5 M	0.533
English	6.0 M	0.449
Finnish	11.0 M	0.469
German	40.0 M	0.486
Hebrew	1.7 M	0.498
Hindi	2.5 M	0.522

Table 2.1: Clustering Coefficients of word networks of six languages. All the networks were created using Chinese Whispers Clustering Algorithm with the 10,000 target words and 200 features

indeed applicable to any other languages.

Chapter 3

Experiments and Evaluation of Word Networks

3.1 Evaluation

Evaluation of the word clusters is challenging and there are two different ways in which this can be done. One way would be to compare the word clusters against a pre-designed set of lexical categories, in which case we are biased towards some gold standard tagset and consequently, contradicting the objective of automatic induction of the categories. Moreover, this method is incapable of evaluating the goodness of the clusters that are finer than the standard tagset. A better way is to resort to some task completion method for evaluation. Unfortunately, in absence of any standard task completion based evaluation strategy for the current work, we compare the clusters against two gold standard tagsets for Bengali described in [7] and [8].

3.1.1 Tag Entropy

Given a word w, a morphological analyzer returns all the possible segmentation of the word w along with the corresponding lexical categories[1]. For example, the Bengali word *kare*

[1] For the tagset presented in [7], we use the morphological analyzer for Bengali described in the same paper. However, for the purpose at hand, it suffices to have a lexicon with all the inflected forms of the root words and their categories. This is what we perform for the tagset presented in [8].

has three possible categories: NN (noun), gloss: palm - locative; VF (finite verb), gloss: do - present, simple, third person; and VN (non-finite verb), gloss: having done.

Let $cat_1, cat_2, \ldots cat_T$ be the universal set of lexical categories, where T is the total number of categories. We define a T-dimensional binary vector Tag_w for a word w as the *tag-vector*, where the value of $Tag_w(i)$ is 1 if and only if according to the morphological analyzer cat_i is a possible category for w. Thus, the tag-vector of *kare* will have 1 only in three positions (corresponding to the categories NN, VF and VN) and rest $T - 3$ positions have 0s.

Given a cluster $c = \{w_1, w_2, \ldots w_s\}$, the cluster is perfectly cohesive if the tag-vectors of all the words in c are identical. On the other hand, the cluster is incohesive if the 1s and 0s are distributed randomly across them. Our objective is to define a metric over the tag vectors of the words in c, which will be able to quantify the cohesiveness of the cluster. Since binary entropy [28] measures the disorderedness of a system, we define the (in)cohesiveness of a cluster c of size s as

$$TE(c) = -\sum_{i=1}^{T} \left(p_i(c) \log_2 p_i(c) + q_i(c) \log_2 q_i(c)\right) \tag{3.1}$$

where

$$p_i(c) = \frac{1}{s}[\text{\# words in } c \text{ for which } Tag_w(i) = 1]$$

and $q_i(c) = 1 - p_i(c)$.

In words, $TE(c)$ is the sum of the binary entropies of the cluster over each of the categories. We call $TE(c)$ the *tag entropy* of the cluster c. For a perfectly cohesive cluster, $p_i(c)$ is 1 or 0 for all i, and therefore, $TE(c) = 0$. For a perfectly incohesive cluster, $TE(c)$ is T. This happens when $p_i(c) = 0.5$ for all the categories. The lower the tag entropy, the higher the cohesiveness of the cluster.

3.1.2 Evaluation Metrics

The clustering algorithm splits the 5000 words into several clusters. Let $C = \{c_1, c_2, \ldots c_r\}$ be the set of word clusters for a particular experimental setup. Based on tag entropy, we define two metrics for evaluation of C: *mean tag entropy* $MTE(C)$ and *weighted mean* *tag entropy* $WMTE(C)$, as follows.

$$MTE(C) = \frac{1}{r}\sum_{i=1}^{r} TE(c_i) \tag{3.2}$$

$$WMTE(C) = \frac{1}{5000}\sum_{i=1}^{r} |c_i| TE(c_i) \tag{3.3}$$

where $|c_i|$ is the number of words in cluster c_i.

We define our baseline as the case when all the 5000 words are in the same cluster. Thus, the baseline MTE is equal to the baseline WMTE, which in turn is equal to $TE(V)$, where V is set of nodes in the network[2]. The motivation behind the definition of baseline is as follows. The quantity $TE(V) - WMTE(C)$ gives an estimate of information gain with respect to the standard tagset by splitting V into set of clusters C. Therefore, the higher the value of this quantity, the better the clustering.

3.2 Experiments

We use the 17M word *Anandabazaar Patrika* (a Bengali daily: http://www.anandabazar.com/) corpus for all our experiments. We have 4 different methods for network construction, 20 different combinations of m and n, 2 different clustering algorithms and 2 gold standard tagsets. This together gives rise to $4 \times 20 \times 2 \times 2 = 320$ possible experiments. It is quite a formidable task to report all these experiments here. Therefore, we divide our experiments into three sets, where we systematically investigate certain parameters.

3.2.1 Set I

In this set of experiments, we fix the network to ${}^{fr,b}G_{n,m}$, use CW clustering algorithm and compare our results for the [7] tagset. Thus, we have 20 experiments corresponding to the various combinations of m and n, the results of which are summarized in Table 3.1. The aim of this set of experiments is to study the behavior of the clusters as we increase the

[2] This is a slight abuse of notation because V is the set of nodes, whereas TE is defined on set of words. Nevertheless, the notation is unambiguous as every node in V correspond to one and only one word.

n	m	Baseline	$MTE(C)$	$WMTE(C)$	% gain for MTE	% gain for $WMTE$	m	Baseline	$MTE(C)$	$WMTE(C)$	% gain for MTE	% gain for $WMTE$
1M	25	4.09 (4.02)	1.75 (1.09)	3.51 (3.30)	57 (73)	14 (18)	100	4.10 (4.03)	1.61 (1.11)	3.57 (3.38)	61 (72)	13 (16)
	50	4.08 (4.01)	1.69 (1.10)	3.53 (3.32)	59 (72)	13 (17)	200	4.11 (4.05)	1.77 (1.12)	3.60 (3.44)	57 (72)	12 (15)
2M	25	4.13 (4.09)	1.60 (0.99)	3.48 (3.30)	61 (76)	16 (19)	100	4.12 (4.08)	1.56 (1.00)	3.51 (3.36)	62 (75)	15 (18)
	50	4.11 (4.08)	1.58 (1.01)	3.49 (3.31)	62 (75)	15 (19)	200	4.14 (4.10)	1.55 (0.96)	3.55 (3.39)	63 (76)	14 (17)
5M	25	4.08 (4.06)	1.52 (1.04)	3.23 (3.04)	63 (74)	21 (25)	100	4.04 (4.01)	1.46 (0.94)	3.22 (3.04)	64 (77)	20 (24)
	50	4.03 (4.01)	1.49 (0.95)	3.21 (3.04)	63 (76)	20 (24)	200	4.03 (4.01)	1.36 (0.81)	3.21 (3.05)	66 (80)	20 (24)
10M	25	4.06 (4.07)	1.41 (0.88)	3.16 (2.94)	65 (78)	22 (28)	100	4.08 (4.10)	1.35 (0.83)	3.17 (2.97)	67 (80)	22 (27)
	50	4.05 (4.07)	1.38 (0.88)	3.16 (2.95)	66 (78)	22 (28)	200	4.07 (4.09)	1.28 (0.77)	3.20 (3.04)	69 (81)	21 (25)
17M	25	4.04 (4.04)	1.53 (1.04)	3.03 (2.83)	62 (74)	25 (30)	100	3.96 (3.97)	1.38 (0.85)	2.97 (2.78)	65 (79)	25 (30)
	50	3.95 (3.96)	1.45 (0.99)	2.93 (2.74)	63 (75)	**26 (31)**	200	3.98 (3.99)	1.32 (0.76)	2.98 (2.81)	67 (81)	24 (29)

Table 3.1: Results for $CW_{n,m}$ model. The values in parentheses refer to the case where the words unknown to the morphological analyzer have been manually corrected. Best results in bold font.

corpus size and number of feature words. There are 450 to 500 clusters (including singletons) per graph found by the CW algorithm[3]. There were a large number of named entities among the target words that were unknown to the morphological analyzer. These words, around 1900 in number, have been manually assigned the appropriate POS categories and included for computation of WMTE.

The best results are obtained for $n = 17M$ and $m = 50$. As is expected, the goodness of the induced lexicon increases rather significantly with the corpus size. For a given corpus, using more feature words does not necessarily improve the results. In general, the ideal value of m seems to be a monotonically increasing function of n.

3.2.2 Set II

In this set of experiments, we investigate the effectiveness of the four different graph construction methods. For this set, we only use the hierarchical clustering method. The evaluations are made against the [7] tagset and all the graphs are constructed for $n = 17M$ and $m = 50$, for which the best results are obtained in Set I.

The primary observation is that the hierarchical clustering gives better result than the CW algorithm. Nevertheless, unlike CW, the WMTE is lower (or the information gain is higher) for hierarchical clustering when the named entities are manually corrected. This implies that CW is able to cluster the named entities more efficiently than hierarchical clustering. Among the graph construction methods, the best results are obtained for ${}^{ms,c}G$,

[3] Some of the example clusters can be found at *http://banglaposclusters.googlepages.com/home*

Metric	${}^{fr,b}G$	${}^{fr,c}G$	${}^{ms,b}G$	${}^{ms,c}G$
WMTE	36.2 (25.3)	37.7 (30.1)	36.7 (26.1)	**39.2 (38.1)**
MTE	86.7 (87.4)	64.0 (75.2)	**87.9 (88.9)**	70.5 (75.5)

Table 3.2: Percentage gain in MTE and WMTE for the 4 different graph construction and agglomerative hierarchical clustering. Best results are in bold fonts. The values in parentheses refer to the case where the words unknown to the morphological analyzer have been manually corrected.

which shows that manual selection of feature words has a positive impact on the word clusters. This revalidates the fact that function words are better suited for POS tag induction.

3.2.3 Set III

As we have mentioned earlier, it is not appropriate to evaluate the goodness of the word clusters that emerge after clustering based on a predefined set of tags. One way to circumvent this problem is to evaluate across multiple tagsets. The previous two sets of experiments are based on the tagset defined in [7]. In the third set of experiments, we use the tagset described in [8] and the dataset made available by the authors (*http://www.hlt.utdallas.edu/∼sajib* consisting of 5000 Bengali words and their corresponding tags to evaluate our clusters. Since we do not have an access to the training corpus used in [8], we have filtered our clusters obtained during the experiments in Set I and Set II, so that they contain only words present in the Dasgupta and Ng dataset. Consequently, the clustered networks now contain around 800 words.

The best results have been obtained for the combination of ${}^{fr,b}G_{17M,50}$ and CW algorithm, for which the entropy reduction is 89% and 57% for MTE and WMTE respectively. Note that these figures are 75% and 31% in the case of Dandapat et al. tagset. The best results for hierarchical clustering is obtained for ${}^{fr,c}G_{17M,50}$, where the respective reductions are 88% and 42%. Although it is tempting to reason that the vast improvement in the results for the Dasgupta and Ng dataset is because of the small number of tags, in reality this might not be the case as the baseline tag entropies for both the datasets are close (around 4). In the next section, we shall discuss the possible reasons behind this improvement.

3.3 Linguistic Analysis and Tagset Design

Bengali is an Indo-Aryan language spoken in Bangladesh and the eastern parts of India. The syntax of the language is morphologically rich and the word order is relatively free. The case relations between the verb and its arguments are usually marked by inflectional suffixes on the nouns. There are a handful of overloaded suffixes that mark various cases depending on the context. Verbs inflect for tense, aspect, mood and person. There are three non-finite verb forms that act as participles and gerund. Bengali has a small repertoire of verb roots and a large number of compound verbs are formed by noun-verb and adjective-verb combinations. Use of "do-support" verbs are also extremely common. Bengali makes use of classifiers (a word/morpheme used to classify nouns according to meaning, number, definiteness etc.), but does not distinguish between gender. Although number distinctions are sometimes reflected through nominal classifiers or suffixes, it is not marked on the verbs.

There has been very few work towards POS tagging in Bengali and consequently there are no standard and well-accepted tagset for the language. For instance, the two tagsets that we have used as gold standards differ substantially in their design principles. The tagset presented in [7] has 40 tags covering the nouns (2 classes), verbs (6 classes), adjectives and quantifiers (6 classes), pronouns (11 classes) and other function words. This tagset is heavily influenced by the English Penn Treebank tagset and words are tagged primarily based on their syntactic function, rather than morphological form. Thus, except for the verbs, the different morphological variations of a root word are not placed into different lexical categories. On the other hand, the tagset described in [8] consists of only 11 tags that partially covers the lexical categories of Bengali. Nouns are divided into 7 classes based on proper vs. common, singular vs. plural and different case-marker (genitive, locative, accusative and nominative) distinctions. There is one class each for adjectives and adverbs. Verbs are divided into two classes based on their morphological form (finite or non-finite). Hence, this tagset has been designed based on the forms of the words rather than their functions.

Let us investigate the nature of the clusters that emerged during our experiments. As discussed earlier, in all the experiments we observe the presence of a few (typically 2 to 4) giant clusters that mainly consist of ambiguous words and thus are “bad” clusters. In fact, it has been observed that by filtering the top few large clusters one can considerably reduce the tag entropy of the clustering. Manual inspection reveals that the medium to small size clusters are “good” and mostly composed of words belonging to similar morpho-syntactic category. There are, however, a few clusters formed on the basis of semantic similarity between the constituent words. See Table 3.3 for some example clusters[4].

The trends in which clusters are formed and merged during the hierarchical clustering provides us useful information about the distinguishabilty between the various lexical classes. We enumerate some of the natural classes that emerged out of our experiments and the categorical distinctions that seem needless for Bengali.

Nouns: Possessive nouns and pronouns (e.g. *gharera* ‘of house’, *tomAra* ‘your’) form a separate cluster and are similar to adjectives in their distribution than other nouns. Although nouns with locative (e.g. *ghare* ‘in house’) and accusative (e.g. *pradhAnamantrIke* ‘to the prime minister’) case-markers form separate clusters initially, they merge with other nouns at a later stage of clustering. We further observe that there is no distinction between the distributions of plural and singular nouns.

Proper Nouns: Different clusters emerge for the different types of proper nouns, such as names of person, location, organization, month and days. Moreover, first and last names of persons show up as separate clusters.

Verbs: In all the models we observe that finite (e.g. *kareChena* ‘have done’), modal (e.g. *pAre* ‘can do’), non-finite (e.g. *uThe* ‘having stood up’) and infinitive (e.g. *karate* ‘to do’) verbs emerge as four basic categories. Non-finites and infinitives merge at a later stage. Verbal nouns (e.g. *khAoyA* ‘to eat’) form a separate cluster initially and later merge with nouns.

Adjectives and Numbers: The distinctions between quantifiers, intensifiers and numbers are observable, though in the later stages of clustering the former two categories merge with other adjectives.

Other Categories: We also observe the question words (e.g. *kI* ‘what’, *kemana* ‘how’), relative pronouns (e.g. *ye* ‘whoever’, *yakhana* ‘whenever’), punctuation marks, conjuncts

[4] In this article, we use Romanized script to represent Bengali words following the ITRANS (*http://www.aczoom.com/itrans/*) convention.

Size	Example Words	Remarks
596	*aruNa, buddhabAbu, saurabha, rAkesha, siddhArtha*	Proper nouns (names of person)
352	*golamAlera* 'of problem', *dAbira* 'of demand', *phalera* 'of result', *Agunera* 'of fire', *dUShaNera* 'of pollution'	Nouns with possessive marker
133	*badalAno* 'to change', *AmAnya* 'disregard', *AkramaNa* 'attack', *sAhAyya* 'help',*guli* 'bullet',	Nouns/verbal nouns that form compound verbs with 'do' or 'be'
44	*sAtaTi* 'seven', *tinaTe* 'three', *anekguli* 'many', *3Ti* 'three', *11Ti* 'eleven'	Quantifiers (mainly cardinal)
13	*adhibeshane* 'during the session', *bhAShaNe* 'in the speech', *baktRRitAYa* 'in the speech', *dalei* 'in the party', *pratibedane* 'in a report'	A semantic cluster related to parliamentary affairs

Table 3.3: Examples of clusters from the ${}^{fr,b}G_{17M,50}$ using CW algorithm.

(e.g. *o* 'and', *bA* 'or') forming separate clusters. However, since these are closed-classes with a very few representative words, it is difficult to make any strong claims about their naturalness.

Therefore, one should take into account the aforementioned factors while designing a tagset for Bengali. Despite the fact that the tagset of [8] makes a larger number of distinctions between the noun forms, this partial tagset, as reflected in our experiments in Set III, has a better correlation with the natural word classes obtained. On the other hand, the Dandapat *et al.* tagset scores poorly on this dimension, primarily because of the finer distinctions made for the verbs and pronouns based on their function. Nevertheless, advanced stages of NLP like chunking and other applications might require such finer distinctions that are not apparent from the natural word classes.

Chapter 4

Application to NER

Named entity recognition (NER) (also known as entity identification (EI) and entity extraction) is a subtask of information extraction that seeks to locate and classify atomic elements in text into predefined categories such as the names of persons, organizations, locations, expressions of times, quantities, monetary values, percentages, etc.

4.1 A Semi-supervised NER Engine

In Chapter 3 we found on maunal inspection of the clusters obtained in the experiments in Sec. 3.2 that the natural word classes are being captured in the clusters. Inspired by this fact, we design a framework to obtain basic Named Entity Recognition (NER) in a semi-supervised fashion. The aim of this basic NER engine is to be able to classify a word given in a text to *name of a person* (NPE), *name of a location* (NPL) and *not a name* (NN). The inputs to the framework are (a)a large corpus of natural language sentences (b) clustering system used in Sec. 2.1.2 (c) list of top 20 most frequent NPE's and NPL's. Additionally, we also employ a the POS tagger for tagging the corpus in (a). In the Sec. 4.1.1, we explain how the inputs are utlised to achieve the goal of the framework mentioned above.

4.1.1 Framework

The framework constitutes of the following divisons on the basis of the tasks performed-

1. Obtain word clusters from the given untagged corpus.
2. Augment the original list of NPE and NPL using clusters obtained in above step
3. Tag the/a corpus sufficiently large using augmented list of NPE,NPL obtained in above step and the POS tagger
4. Extract suitable features from the tagged corpus of above step and using a suitable machine learning algorithm, learn the NER engine (classifier).

We now elaborate these divisons to explain the framework:

Obtain Word Clusters

We repeat the procedure given in Sec. 2.1.2 using the given corpus and perfom Chinese Whispers clustering or agglomerative clustering to and obtain clusters. As observed in Sec. 3.3, these clusters are morpho-syntactic in nature and we exploit this property in next step.

Augmenting the NPE,NPL List

Since, we have morpho-syntactic clusters, we can assume that the words of similar nature or *named entity* to be more specific will lie in the same clusters. Hence, using the top 20 manually hand-picked NPE's and NPL's, we will identify the clusters which these belong to. These clusters will have more NPE's and NPL's using which we can now automatically augment the NPE and NPL word list. Hence, using the clustering obtained from previous step and the list of 20 manually chosen NPE's and NPL's we obtain a larger list of words which are tagged NPE and NPL.

Tagging the Corpus

Now that we have a larger list of tagged words with NPE and NPL. We take the a suitably large corpus and tag the words in the corpus with NPE, NPL or NN according to the following scheme-

```
    foreach word in Corpus
if word in list o $NPE$ words
$word_{tag} \Rightarrow NPE$
else if word in list of $NPL$ words
$word_{tag} \Rightarrow NPL$
else
$word_{tag} \Rightarrow$ NN
end-of-foreach
```

Next, we tag each word using it's POS tag using a standard POS tagger for the language of reasonable accuracy. We choose to tag each word with it's POS tag since we want to increase the number of features available to learn the classifier in the next step. Hence, in this step, we obtain a suitably large tagged corpus.

Learning NER Classifier

In this final step, we create a suitable feature representation for each word of the tagged corpus obtained in the previous step. We then employ a suitable classifer to learn the basic NER classification from the feature representation. Once the classifier is learnt, we can test it on any test input sentence, which is POS tagged.

4.2 NER Implementation

In this section we will go through the implementation specifics of the NER framework. We pick Bangla for the implementation, so that we can readily start with the heavily analysed clustering data we already have in obtained in Chapter 2. We now go through the implementation of each of the steps discussed in the previous section.

Obtain Word Clusters

We choose the Chinese Whispers Clustering of Bengali *Anand Bazaar Patrika* corpus ($G_{17M.50}$ - 17 million words and top 50 frequent words as feature words) as obtained in 2.1. [1]

[1] the morpho-syntactic clusters were observed to be best in this

Augmenting the NPE,NPL List

From the lexicon of the above used corpus, 20 NPE and NPL's were manually picked. Then, the clusters are automatically identified which contain the words of the NPE/NPL list and each word of those clusters are added to the NPE/NPL list respectively.

Tagging the Corpus

Using the algorithm provided by the framework, 1 Million words of the corpus are then tagged with the NPE and NPL taglist. A POS tagger [2] for bangla [7] is employed to obtaind the POS tagging for each of the words. Hence, a sentence of the corpus in ITRANS is given in example below.

Example 1 *aphisabA.Dite:NN:NN Dhuke:VM:NN ghaNTA:NN:NN de.Deka:QO:NN dhare:NN:NN abAdhe:VM:NN luThapATa:VAUX:NN chAlAla:XC:NN 10:QC:NN janera:NN:NN sashastra:NN:NN dala.:NN:NN budhabAra:VM:NN gabhIra:JJ:NN rAte:NN:NN ghaTanATi:NN:NN ghaTeChe:VM:NN mallikabAjAre.:NNP:NN chAra:QO:NN dAroYAnake:NN:NN be.Ndhea,:JJ:NN moTa:QF:NN pA.NchaTi:QC:NN aphisa:NN:NN theke:PSP:NN prAYa:RB:NN 10:QC:NN lakSha:NN:NN TAkA:NN:NN hAtiYe:VM:NN champaTa:JJ:NN deYa:NN:NN durbRRitterA.:JJ:NN DAkAtadera:NN:NN ChorAYa:NN:NN eka:QC:NN aphisa-rakShI:JJ:NN gurutara:JJ:NN Ahata:VM:NN haYeChena.:VAUX:NN Di:NNP:NN si:NNP:NN* ***sa njaYa:XC:NPE mukhopAdhyAYa:NN:NPE*** *jAniYeChena,:VM:NN duShkRRitIrA:NN:NN sambhabata:JJ:NN shaharera:NN:NN bAire:NST:NN theke:PSP:NN eseChila.:NN:NN tAdera:PRP:NN kho.Nje:NN:NN* ***goYendArA:NN:NPE*** *kalakAtAra:NNP:NN bAire:NST:NN giYeChena.:NN:NN shyAmilTana:XC:NN hAusa:NN:NN nAme:NN:NN mallikabAjArera:NNP:NN oi:DEM:NN pA.NchatalA:JJ:NN bA.Dite:NN:NN antata:RP:NN 15Ti:QC:NN besarakAri:JJ:NN sa.nsthAra:NN:NN aphisa:NN:NN AChe.:NN:NN budhabAra:VM:NN rAta:VAUX:NN A.DAiTA:QC:NN nAgAda:JJ:NN piChanera:NN:NN pA.Nchila:JJ:NN Di NiYe:VM:NN Dhuke:VM:NN DAkAterA:NN:NN chAra:NST:NN dAroYAnera:NN:NN mAthAYa:NST:NN ribhalabhAra:NN:NN ThekiYe:VM:NN tA.Ndera:PRP:NN piChamo.DA:VM:NN kare:VAUX:NN be.N.Ndhe:VM:NN phele.:VAUX:NN chAra:QO:NN janakei:NN:NN ekatalAYa:QC:NN ATake:NN:NN DAkAterA:NN:NN*

[2] We thank Sandipan et al. for lending their POS tagging system [7]

NN	NPE	NPL	←Classified as
960398	1009	43	NN
23130	3178	1	NPE
7668	267	758	NPL

Table 4.1: Confusion matrix for the stratified 10 fold testing of 1 Million tagged corpus

upare:NST:NN uThe:VM:NN yAYa.:VAUX:NN bhora:NN:NN 4Te:VM:NN
paryanta:PSP:NN luThapATa:JJ:NN chAlAYa.:NN:NN

Note that the first tag is the POS tag given by POS tagger [7] and the NPE tagged by using the list has been made bold.

Learning NER Classifier

Now that we have a million words tagged with a basic NER tag and it's POS tag, we plan to learn using a decision tree classifier $C4.5$ [24]. We used the popular implementation of the $C4.5$, the $J48$ classifier with the tool Weka [30]. At first, we obtained the following feature for each word using the tagged dataset-

$$Pre_3Tag, Pre_2Tag, Pre_1Tag, CurrPOS, Pre_3POS, Pre_2POS, Pre_1POS, \\ Next_1POS, Next_2POS, Next_3POS$$

where Pre_iTag stands for the tag of the i^{th} word before the current word in the sentence, Pre_iPOS stands for the POS of the i^{th} word before the current word in the sentence and $Next_iPOS$ stands for the tag of the i^{th} word after the current word. The feature is then converted to the format required by Weka and the decision tree classifier is trained with default parameters of the J48 classifier in Weka. Test is performed using the standard stratified 10 fold cross validation process and an accuracy of 97.78%. An analysis of the confusion matrix of Table 4.1 shows that a good percentage of NPE's and NPL's have changed tags [3]. This shows that the classifier has probably learnt from the tagged corpus as there was misinformation and generalisation in the tagged data, since the tagging was very rudimentary and it might not be overfitting.

[3] ideally all of the should have been along the diagonal of the matrix

NN	NPE	NPL	←Classified as
95600	0	0	NN
0	800	0	NPE
0	0	400	NPL

Table 4.2: Confusion matrix for the testing $100K$ contiguous words from the $17M$ corpus

With this NER classifier, we evaluated a test set of contiguous $100K$ words generated from the same corpus from a different location and obtained an accuracy of 100%. The confusion matrix for the test set is given in Table 4.2, which shows that the classifier is not underfitting either. Both overfitting and underfitting are bad inorder to achieve a good classifier.

The NER classifier now can be used on a manually tagged data and compared againse the existing NER engines for bangla like in [10]

Chapter 5

POS Tagged Word Network

Inspired by the strong associativity of POS tags to clusters derived from the word network in one hand and existence of clusters that are associated with the multiple POS tags from Chapter 2, we delve to analyse word networks that are created when POS of the word is known before hand. Hence, we define and examine a new network - ‘pos-word network’.

5.1 Definition and Construction of Pos-word Networks

The pos-word network is a weighted undirected graph $G_{POS} = \langle V, E \rangle$, where V consists of 10000 nodes corresponding to the most frequent 10000 types occuring a suitably large lexicon. The number of nodes in V has been decided based on the fact that with a corpus of size around 10M words, enough context information is available only for the top few words. The weight of the edge between any two nodes representative of the words w and v is given by POS-similarity $sim_{POS}(w, v)$ and this edge exists if $sim_{POS}(w, v)$ exceeds a threshold τ and sim_{POS} measures the simlarity of words in acquiring similar POS tags.

We start with a POS tagged corpus. Each and every word will possibly have multiple tags. If T=$<t_1, \ t_2, \ t_3, \ t_4, \ ... \ , \ t_k>$ is a set of all possible POS tags, then each word $w_i$in the corpus would have a POS-vector V_i=$<n_i1, \ n_i2, \ n_i3, \ n_i4, \ ... \ , \ n_ik>$, where each of $n_ij's$ correspond to the number of times the word w_i gets the POS tag t_j. Now, taking the top 10000 words as nodes and cosine distance between the words as the edge weights (we ofcourse threshold the edges), we obtain a weighted word network to observe that, indeed the words which assume multiple tags have high degrees and high clustering coefficients.

Hence, for creating the word network, we obtain a lexicon of the language we intend to create the word network. Take top 10000 words along with their possible tags from their lexicon and create the word POS-vectors and then find the edges by evaluating and thresholding the POS similarity between each word's (node's) POS-vectors. We obtain the graph and evaluate some of the properties discussed in Section 2.2

5.2 Experiment

We pick up two languages- English and German and obtain tagged corpus of sizes $150K$ and $350K$ respectively, we obtain their pos-word networks. As expected, they are found to have high clustering coefficients (English-1.9259 , German- 1.9688) and the words which usually acquire multiple POS tags are found to exhibit high degrees and high node clustering coefficients.

Chapter 6

Conclusion

In this work, we presented a principled and systematic approach to understand the syntactic structure of Bengali and other languages and induce the natural word classes of this language. We summarize below our salient observations.

- The degree distribution of the network follows a power-law behavior reflecting a hierarchy of the words with respect to their syntactic ambiguities.

- The clustering coefficient of the network is significantly higher than that of the random graphs pointing to the presence of strong community structures that are representative of the natural word classes.

- Clustering splits the network into word classes representing different lexical categories and the cluster size distribution follows a power-law. There are a very few giant clusters consisting of many ambiguous words and a large number of medium to small size clusters consisting of mostly unambiguous words.

- The results obtained for all the different graph construction and clustering algorithms are very close to each other implying the underlying robustness of the distributional hypothesis. However, the size of the corpus has a strong effect on the quality of the emerging clusters.

- We note that morphology plays a significant role in defining the syntactic clusters of Bengali. However, it may be harmful to start with the assumption proposed in [8] that each morphological category defines a syntactic class. In particular, we do observe possessive nouns and finite, non-finite and infinitive verbs forming separate clusters, but we also observe that presence of plural markers (e.g. *der*, *rA*) or accusative or locative inflections for nouns need not essentially mark a separate syntactic category.

- We note that the NER framework based on the morpho-syntactic cohesiveness of clusters yeilds us a robust semi-supervised NER engine, useful especially in resource scarce languages.
- We have also verified the nature of ambiguous words holding high degrees and high clustering coefficients in both word networks and pos-word networks.

In conclusion, the pen and paper based linguistic analysis technique for identification of lexical categories might well be automated in a principled manner by exploiting the concept of distributional hypothesis, with which we went on to propose a robust semi-supervised NER framework. Cross-linguistic study of the topology of the word networks can reveal several universal properties as well as typological variations in the linguistic systems. Apart from providing insights into the natural word classes leading to the design of appropriate tagsets, the study of these networks can significantly increase our understanding of the evolution of syntax.

Bibliography

[1] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.

[2] C. Biemann. Chinese whispers - an efficient graph clustering algorithm and its application to natural language processing problems. In *Proceedings of HLT-NAACL'06 workshop on TextGraphs*, pages 73–80, 2006.

[3] C. Biemann. Unsupervised part-of-speech tagging employing efficient graph clustering. In *Proceedings of COLING/ACL'06 Student Research Workshop*, pages 7–12, 2006.

[4] A. Clark. Inducing syntactic categories by context distribution clustering. In Claire Cardie, Walter Daelemans, Claire Nédellec, and Erik Tjong Kim Sang, editors, *Proceedings of CoNLL/LLL'00*, pages 91–94. 2000.

[5] A. Clark. Combining distributional and morphological information for part of speech induction. In *Proceedings of EACL'03*, pages 59–66, 2003.

[6] L. da F. Costa, O. N. Oliveira Jr., G. Travieso, F. A. Rodrigues, P. R. V. Boas, L. Antiqueira, M. P. Viana, and L. E. C. da Rocha. Analyzing and modeling real-world phenomena with complex networks: A survey of applications. *arXiv:0711.3199*, 2007.

[7] S. Dandapat, S. Sarkar, and A. Basu. A hybrid model for parts-of-speech tagging and its application to Bengali. *International Journal of Information Technology*, 1(4):169–173, 2004.

[8] S. Dasgupta and V. Ng. Unsupervised part-of-speech acquisition for resource-scarce languages. In *EMNLP-CoNLL'07*, pages 218–227, 2007.

[9] I. S. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic co-clustering. In *Proceedings of KDD'03*, pages 89–98, 2003.

[10] Asif Ekbal and Sivaji Bandyopadhyay. A web-based Bengali news corpus for named entity recognition. *Language Resources and Evaluation* , 2008.

[11] R. Ferrer-i-Cancho. The structure of syntactic dependency networks: insights from recent advances in network theory. In V. Levickij and G. Altmman, editors, *Problems of quantitative linguistics*, pages 60–75. 2005.

[12] R. Ferrer-i-Cancho, A. Capocci, and G. Caldarelli. Spectral methods cluster words of the same class in a syntactic dependency network. *International Journal of Bifurcation and Chaos*, 17(7), 2007.

[13] R. Ferrer-i-Cancho, A. Mehler, O. Pustylnikov, and A. Díaz-Guilera. Correlations in the organization of large-scale syntactic dependency networks. In *Proceedings of HLT/NAACL'07 workshop on TextGraphs*, pages 65–72. Association for Computational Linguistics, 2007.

[14] R. Ferrer-i-Cancho and R. V. Sole. The small world of human language. *Proceedings of The Royal Society of London. Series B, Biological Sciences*, 268(1482):2261–2265, November 2001.

[15] R. Ferrer-i-Cancho and R. V. Sole. Patterns in syntactic dependency networks. *Physical Review E*, 69(051915), 2004.

[16] S. Finch and N. Chater. Bootstrapping syntactic categories using statistical methods. In *Background and Experiments in Machine Learning of Natural Language: Proceedings of the 1st SHOE Workshop*, pages 229–235. Katholieke Universiteit, Brabant, Holland, 1992.

[17] D. Freitag. Toward unsupervised whole-corpus tagging. In *COLING'04*, page 357, 2004.

[18] S. Gauch and R. Futrelle. Experiments in Automatic Word Class and Word Sense Identification for Information Retrieval. In *Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 425–434, 1994.

[19] S. Goldwater and T. Griffiths. A fully bayesian approach to unsupervised part-of-speech tagging. In *Proceedings of ACL'07*, pages 744–751, June 2007.

[20] A. Haghighi and D. Klein. Prototype-driven learning for sequence models. In *Proceedings of HLT-NAACL'06*, 2006.

[21] Z. S. Harris. *Mathematical Structures of Language*. Wiley, New York, 1968.

[22] G. A. Miller and W. G. Charles. Contextual Correlates of Semantic Similarity. *Language and Cognitive Processes*, 6(1):1–28, 1991.

[23] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45:167–256, 2003.

[24] J. R. Quillian. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, 1993.

[25] R. Rapp. A practical solution to the problem of automatic part-of-speech induction from text. In *Proceedings of ACL'05 (companion volume)*, pages 77 – 80, 2005.

[26] H. Schütze. Part-of-speech induction from scratch. In *Proceedings of ACL'93*, pages 251–258, 1993.

[27] H. Schütze. Distributional part-of-speech tagging. In *Proceedings of EACL'95*, pages 141–148, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[28] C. E. Shannon and W. Weaver. *The Mathematical Theory of Information*. University of Illinois Press, 1949.

[29] L. Steels. Language as a complex adaptive system. In *Proceedings of PPSN VI*, pages 17–26, 2000.

[30] Ian H. Witten and Eibe Frank. *Data Mining: Practical machine learning tools with Java implementations*. Morgan Kaufmann, 1993.