Word Context and Token Representations from Paradigmatic Relations and Their Application to Part-of-Speech Induction

by

Enis Rıfat Sert

A Thesis Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in

Computational Sciences and Engineering

Koç University

September, 2013

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Enis Rıfat Sert

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

_____

Assoc. Prof. Deniz Yuret

_____

Asst. Prof. Emine Yılmaz

_____

Asst. Prof. Gülşen Cebiroğlu Eryiğit

Date: _____

In memory of my beloved brother

Engin Deniz Sert
1989 – 2011

Abstract

Representation of words as dense real vectors in the Euclidean space provides an intuitive definition of relatedness in terms of the distance or the angle between one another. Regions occupied by these word representations reveal syntactic and semantic traits of the words. On top of that, word representations can be incorporated in other natural language processing algorithms as features.

In this thesis, we generate word representations in an unsupervised manner by utilizing paradigmatic relations which are concerned with substitutability of words. We employ an Euclidean embedding algorithm (S-CODE) to generate word context and word token representations from the substitute word distributions, in addition to word type representations. Word context and word token representations are capable of handling syntactic category ambiguities of word types because they are not restricted to a single representation for each word type.

We apply the word type, word context and word token representations to the part-of-speech induction problem by clustering the representations with k-means algorithm and obtain type and token based part-of-speech induction for Wall Street Journal section of Penn Treebank with 45 gold-standard tags. To the best of our knowledge, these part-of-speech induction results are the state-of-the-art for both type based and token based part-of-speech induction with Many-To-One mapping accuracies of 0.8025 and 0.8039, respectively. We also introduce a measure of ambiguity, Gold-standard-tag Perplexity, which we use to show that our token based part-of-speech induction is indeed successful at inducing part-of-speech categories of ambiguous word types.

Özetçe

Kelimelerin Öklit uzayında gerçek yoğun vektörler tarafından temsili kelimeler arasındaki ilgililiğin uzaklık ve açı cinsinden tamınlanmasına olanak sağlamaktadır. Kelime temsilleri tarafından işgal edilen bölgeler kelimelerin sözdizimsel ve anlamsal özelliklerini yansıtmaktadırlar. Bunlara ek olarak, kelime temsilleri doğal dil işleme algoritmalarına öznitelik olarak eklenebilmektedirler.

Bu tez içinde, kelime temsillerini denetimsiz olarak, örneksel ilişkilerini yani kelimelerin değiştirilebilirliğini kullanarak üretiyoruz. S-CODE isimli Öklitsel gömme algoritmasını çalıştırarak kelime türü temsillerine ek olarak, kelime bağlamı ve kelime andacı temsilleri elde ediyoruz. Kelime bağlamı ve kelime andacı temsilleri her kelime türü için sadece bir temsille kısıtlanmadıkları için çok sözdizimsel kategorili kelimerle başa çıkma yeteneğine sahiptirler.

Kelime türü, kelime bağlamı ve kelime andacı temsilerini k-means algoritmasını kullanarak kümeleyip sözcük türü tümevarımı (part-of-speech induction) problemine uyguluyoruz. Penn Treebank bütüncesinin 45 sözcük türü etiketli Wall Street Journal kısımı için tür ve andaç temelli sözcük türü tümevarımları elde ediyoruz. Sözcük türü tümevarımlarımız ile tür temelliler icin 0.8025 ve andaç temelliler icin 0.8039 Çoktan-Bire eşleme kesinlikleri elde ediyoruz. Bildiğimiz kadarıyla tekniklerimiz bu sonuçlarla alandaki en gelişmiş teknikler olmuşlardır. Bununla beraber, çok anlamlılığı ölçmek için 'Altın Standart Etiket Tereddütü' ölçüsünü takdim ederek andaç temelli sözcük türü tümevarımlarımızın çok sözdizimsel kategorili kelimelerde başarılı olduğunu gösteriyoruz.

Acknowledgements

I am deeply indebted to Assoc. Prof. Deniz Yuret who took a chance on me and agreed to be my advisor. His wisdom, guidance and foresight made this work possible. His unconventional attitude (compared to what I observed in the Graduate School of Koç University) towards his research team made the work much more enjoyable. The conversations we had were always a pleasure for me. For all these, I thank him.

I thank Asst. Prof. Emine Yılmaz and Asst. Prof. Gülşen Cebiroğlu Eryiğit for their valuable comments and for taking their time to take a part in my thesis committee.

I thank to my colleagues and friends in Koç University Artificial Intelligence Laboratory and Graduate School of Engineering. Mehmet Ali Yatbaz's efforts were fundamental in realization of this work. The conversations I had with him, Emre Ünal, Aydın Han, Volkan Cirik and Hüsnü Şensoy were fun and enlightening. Barış Çağlar was a great supportive friend in both undergraduate and graduate years. Tolga Bağcı, Yusuf Sahillioğlu, Eray Varlik, Mehmet Akif Yalçınkaya, Arda Aytekin, Talha Akyol, Mustafa Haboğlu and Berkay Yarpuzlu were some of the people who made the graduate school experience fun.

I also thank İdil Arşık who was there for me through thin and thick for the last two years. She witnessed tough periods of my life and she cared about me and supported me for which I am grateful and feel very lucky.

Finally, I thank to my parents İnal and İsmail Zahir Sert for their everlasting love, patience and support, despite the indescribable pain they are in. No matter what happens, I know they will be there for me.

Contents

List of Tables

List of Figures

Glossary

context is the ordered sequence of word types that surrounds a target word token.

part-of-speech induction is the process of tagging the word tokens in a corpus with identifiers to cluster the word tokens of the same syntactic category in an unsupervised manner.

representation (or word representation) is a dense real vector in the d dimensional Euclidean space that is associated with a word.

target word token is the word token of interest in a sentence or a document.

token based part-of-speech induction is a part-of-speech induction process that is not obliged to assign the same category to word tokens of the same word type.

type based part-of-speech induction is a part-of-speech induction process that assigns the same category to word tokens of the same word type. Ambiguous word types that are observed with different syntactic categories depending on their context can not be tagged with different identifiers.

word context representation (or context representation) is the representation of the context of a word token. If contexts of two different word

tokens are the same (same sequence of word types), context representations result in the same representation. Context representations are independent of the target word the contexts surround.

word token (or token) is an instance (observation) of a word type.

word token representation (or token representation) is the representation of a word type with its context. Word token representations has traits of both word types and word context representations.

word type (or type) is a unique word (sequence of characters including numbers and punctuation) in a vocabulary generated from a corpus.

word type representation (or type representation) is the representation of a word type. Each word token of the word type shares the same representation.

Please note that this glossary describes the terms according to their usage in thesis and the descriptions may not be completely compatible with other works in the literature.

Chapter 1

Introduction

This thesis investigates representations of words as dense real vectors in the Euclidean space that are generated in an unsupervised manner and their application to the part-of-speech (also called lexical or syntactic) category induction problem.

Learning a mapping from words to vectors in the Euclidean space provides an intuitive definition of relatedness in terms of the distance (e.g. Euclidean distance) or the angle (e.g. cosine similarity) between the word representations. These similarities reveal semantic and syntactic relations between words (Huang et al., 2012; Mikolov et al., 2013; Schütze, 1998). These relations are useful in constituent parsing (Socher et al., 2013), part-of-speech induction (Lamar et al., 2010b; Maron et al., 2010; Schütze, 1995), sentiment analysis (Maas et al., 2011), word sense disambiguation (Schütze, 1998).

In addition, it is possible to incorporate the word representations as if they are additional features in order to improve the performance of supervised natural language processing algorithms for tasks like chunking, named entity recognition, part-of-speech tagging, semantic role labeling and sentiment classification (Collobert et al., 2011; Dhillon et al., 2011, 2012; Turian et al., 2010).

Learning part-of-speech categories of words, on the other hand, is one of the fundamental problems in natural language processing. Since the grammar rules do not apply to the individual words but to their syntactic categories,

any word in a well formed sentence can be exchanged with an arbitrary word and the sentence would still be well formed as long as the replacement word is in the same part-of-speech category as the replaced word (but most likely the semantics of the sentence would be different). Being able to group words into part-of-speech categories has its benefits in applications such as parsing and machine translation that aim to learn or utilize grammar.

Young children are able to form original syntactically correct sentences without ever knowing the formal definition of nouns or verbs by being exposed to natural language from adults, other children or TV over time (Brown and Berko, 1960), which suggests that they form a categorization of word types. However, there is still no computational model for acquisition of the part-of-speech categories from unlabeled text (i.e. part-of-speech induction) that is comparable in performance with humans which is one of the reasons we are interested in the problem. Another reason for the interest is that languages with poor reserve of labeled data benefit from the unsupervised methods as only raw text data is required instead of costly human labor. Part-of-Speech categories to be induced would prove to be useful in natural language processing applications in these languages.

In this thesis, we build on the work of Yatbaz et al. (2012) who generate word type representations from large amounts of unlabeled text by representing the paradigmatic relations of word tokens in their contexts as substitute word distributions and utilizing an Euclidean embedding algorithm named S-CODE. Paradigmatic relations are concerned with the substitutability of the word token with another word type. Another type of relation between the word tokens and their contexts are syntagmatic relations which are concerned with the positioning of the word token within the context. Figure 1.1 is an illustration of these relationships as axes on a simple sentence.

By applying the word type representations to the part-of-speech induction problem and outperforming previous work, Yatbaz et al. (2012) show that paradigmatic relations are powerful at capturing the syntactic similarities. The presented word type representations are successful at capturing the syntactic properties of unambiguous word types as shown by their performance at the part-of-speech induction task. However, word type representations

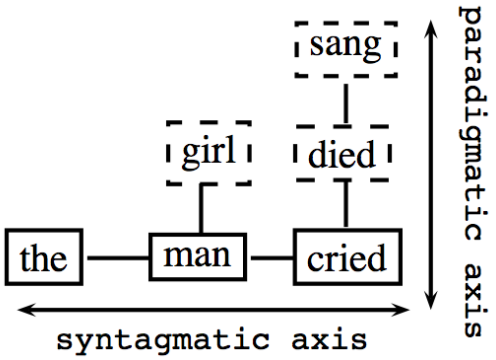


Figure 1.1: Syntagmatic and paradigmatic axes on a the sentence "the man cried" (Chandler, 2007).

are fundamentally limited in their ability to reflect the semantic and syntactic properties of ambiguous word types as they do not take the individual contexts the word types are observed into consideration. For a concrete example, observe the following sentences containing the word type 'work': "I work at a public library." and "I am at work right now.". From the context they are observed in, it is clear that both instances of 'work' are assigned with different syntactic and semantic duties in the sentences and a single representation simply can not capture both of these observations.

In this thesis, in addition to the word type representations, we generate representations for word contexts and word tokens from paradigmatic relations captured as substitute word distributions. Word context representations are vectors associated with the ordered sequence of words surrounding the target word type of interest. Two different word tokens with identical surrounding words would have the same context representations, since the actual target word types do not affect the word context representations. Word token representations are vectors that capture the properties of both target word's type representation and the target word's context representation.

To show that our word context and word token representations capture syntactic similarities and handle ambiguities of word types, we once again decided to apply them to the task of part-of-speech induction. While the part-of-speech induction task has a high upper bound for type based induction, non-trivial portions of corpora still consist of ambiguous word types

(e.g. most frequent tags for word types in Wall Street Journal section of Penn Tree Bank (Marcus et al., 1999) result in 93.69% Many-To-One accuracy but 14.94% of the tokens in the corpus consist of considerably ambiguous types (see Section 4.4)). Induction of the part-of-speech categories are based on clustering of the word type, word context and word token representations. Clustering of the word type representations result in type based part-of-speech induction. Clustering of the word context and word token representations result in token based part-of-speech induction.

The structure of the thesis is as follows:

Chapter 2 details the related work on the word representations and part-of-speech induction. We examine the word representations under three categories based on how they are obtained. The literature review on the part-of-speech induction are focussed on HMMs and their training. We also examine few methods that are not based on HMMs as well.

Chapter 3 describes the pipeline of operations that begin with the representation of paradigmatic relations with substitute distributions and end with the evaluation of part-of-speech induction.

Chapter 4 reports our experimental settings, results and compares them to part-of-speech induction systems in the literature. We also introduce morphological and orthographic features and compare our system to other feature incorporating part-of-speech induction systems. In what follows, we compare our word representations with the word representations in the literature on the part-of-speech induction task. We end the chapter with an analysis of parameters and word token representations.

Chapter 5 concludes the thesis with a summary of our work and offers future topics to be investigated based on the word representations we present.

Chapter 2

Related Work

In this chapter, we survey the previous studies on word representations and on part-of-speech induction systems in two separate sections. We start the discussion of word representations with systems that only produce word type representations and end the section with systems that are able to produce word context or word token representations as well. We complete the chapter with a short review of the part-of-speech induction literature.

2.1 Word Representations

There are two popular means of obtaining word representations from unlabeled text. The first of them is utilization of frequency matrices. A frequency (or co-occurrence) matrix F is a matrix whose elements f_{ij} are the frequency (or number times) of observing events i and j together. If the rows of such matrix are associated with words and the columns are associated with another event (e.g. the contexts), then employing rows as word representations is one of the many ways of obtaining representations from the matrix. Turney and Pantel (2010) present a comprehensive review of frequency matrix based methods used to derive word representations for investigating semantic relations. The second popular means of obtaining word representations is based on neural networks. Neural network based language models, whose performance challenge the established n-gram and discounting based lan-

guage models, predict the word following a sequence of words and generate word representations as a side effect of the learning process. There are also methods that do not fall into these two categories.

We begin with the literature on word type representations. We follow up with the literature on word context and word token representations.

2.1.1 Word Type Representations

Word type representations stand for the case of association of all tokens of a word type with a single representation. These representations, obviously, are unable to handle word types with multiple meanings or taking on more than one syntactic duty in sentences depending on the word's context (i.e. ambiguous word types).

In Latent Semantic Analysis (LSA) (Deerwester et al., 1990), a frequency matrix of word types and contexts (word, document or any other context) is formed where each element f_{wc} is the number of occurrences of word type w in the context c. Then, dimensionality of this matrix is reduced using singular value decomposition (SVD) and the resulting rows are used as word type representations.

Kanerva et al. (2000) and Sahlgren (2005) obtain word type representations very similarly to LSA, by reducing the dimensionality of a frequency matrix of word types and contexts. But instead of using computationally costly SVD, they use Random Indexing. In Random Indexing, first, a random low dimensional representation for each context is generated. Then, for each word type, weighted (by their observation frequency with the word type) sum of these representations are determined to be the word type representation.

Collobert and Weston (2008) present a convolutional neural network architecture that not only learn a language model and generate word type representations from unlabeled data, but also jointly train predictors for part-of-speech tags, named entities, semantic roles and chunking using labeled data.

Mnih and Hinton (2007) introduce the log-bilinear language model that

can be interpreted as a feed-forward neural network with one linear hidden layer and a softmax output layer. The model learns and utilizes linear combination of word type representations of the words preceding the word position to be predicted. Mnih and Hinton (2009) modify this model to have a hierarchical structure in order to reduce computational cost and name it hierarchical log-bilinear language model.

Mikolov et al. (2010) use recurrent neural networks to train a language model and obtain word type representations. Mikolov et al. (2013) show that these representations capture syntactic and semantic regularities in terms of offsets in the vector space, such as $x_{apple} - x_{apples} \approx x_{car} - x_{cars} \approx x_{family} - x_{families}$ where x_w is the representation of word type w.

Maron et al. (2010) introduce the S-CODE framework, an extension of the CODE (Globerson et al., 2007) framework, that obtains word type representations from co-occurrence data generated from syntagmatic relations of words. Yatbaz et al. (2012) extend this work by generating word type representations from paradigmatic relations, which we extend in thesis to obtain word context and word token representations.

Lamar et al. (2010b) employ the approach of Schütze (1995) (see Section 2.1.2) in a two step manner to obtain word type representations. In the first step, authors replicate the method of Schütze (1995) to obtain word type representations. In the second step, word type representations from the previous step are clustered. Two new co-occurrence matrices for left and right cluster neighbors of each word type are formed and first step is reapplied to these new matrices to generate word type representations.

Maas et al. (2011) develop a probabilistic model inspired by Latent Dirichlet Allocation to learn word type representations from unlabeled data that reflect semantic relations. Additionally, authors incorporate labeled sentiment data to capture sentiment relations on top of the semantic relations.

Luong et al. (2013) train their neural network language model on morpheme level instead of word level and generate representations for the morphemes. Word type representations are generated on the fly from the morphemes that construct them which in turn estimates rare, morphologically complex or unseen word types more sensibly.

2.1.2 Word Context and Word Token Representations

Word context representations are associated with the contexts of the words instead of the words themselves. These representations can handle ambiguous word types, since each observation of a word type has a unique context which has a unique representation.

Word token representations are associated with the context of a word and the word itself at the same time. In a sense, they are hybrids of word type representations and word context representations. Word token representations are able to handle ambiguous word types as well, since they take the context into consideration.

It is possible to have duplicate word context representations for two different word types with the same context, since word context representations are independent of the target word itself. Word token representations do not have this issue, because the extra information from the word type differentiates word types with the same context from each other.

Schütze (1995) forms two separate co-occurrence matrices for left and right neighbors of each word type. Combinations of these matrices such as concatenation of left and right matrices are used to compose new matrices. By applying SVD to these new matrices, in order to reduce dimensionality and sparseness, word type, word context and word token representations are generated.

Schütze (1998) performs word sense discrimination by forming a word context representation for each occurrence of an ambiguous word type and clustering them. These word context representations are obtained by the weighted average of the representations of the word types in the context window. Each word type representation is derived from a co-occurrence matrix of neighboring words.

Reisinger and Mooney (2010) conceive word context representations from the word types in the context windows of word tokens. Word context representations corresponding to each word type are clustered and cluster centroids are used to represent word types in a level that is finer than type level but coarser than token level (i.e. k representations for word type w where

$k \ll n$ and n is the number of tokens with type w) and are called prototypes of the word types.

Huang et al. (2012) train a neural language model by using both local (word sequence) and global (document) context via a joint training objective and generate word type representations. By taking the weighted average of the representations of word types in the fixed size context windows of word tokens, authors form word context representations for those tokens. Just as Reisinger and Mooney (2010), authors cluster these word context representations for each word type to obtain word type prototypes that account for homonymy and polysemy.

Dhillon et al. (2011) introduce Low Rank Multi-View Learning model that learns word token representations from left and right co-occurrence matrices which have a row for each word token in the training corpus. The model uses left and right matrices to generate a matrix that projects high dimensional co-occurrence matrices to lower dimensional matrices using Canonical Correlation Analysis (CCA).

2.2 Part-of-Speech Induction

We begin this section by focusing on the Hidden Markov Models because a great deal of previous work on part-of-speech induction are based on HMMs. A bigram HMM assumes that there is a sequence of tags $T = t_1, \ldots, t_n$ that are invisible during the observation and they generate the word sequence $W = w_1, \ldots, w_n$ we observe in the corpus and the likelihood of the corpus is expressed as $P(W, T) = P(w_1|t_1)P(t_1)\prod_{i=2}^{n} P(w_i|t_i)P(t_i|t_{i-1})$ where n is the number of word tokens in the corpus. A trigram HMM is same as the bigram model with the exception of each hidden tag depends on two hidden tags before it instead of just one. Models try to discover the tag assignment sequence that maximize the likelihood of the system. Christodoulopoulos et al. (2010) offer a wide ranging review of the systems that tackle part-of-speech induction problem with HMMs and evaluation measures for the induced tags.

Brown clustering (Brown et al., 1992) is an approximate greedy hierar-

chical clustering algorithm that starts from a trivial tag sequence (e.g. each word type is in its own cluster) and merge clusters that improve the likelihood stated above and result in a type based part-of-speech induction.

Clark (2003) employs an approach similar to Brown clustering. Instead of hierarchical clustering, the author uses a search algorithm that iteratively changes cluster of the word types to the cluster that provides the maximum improvement in the likelihood of the model. The model is also augmented to incorporate morphology and frequency of the word types. The system results in type based part-of-speech induction.

Biemann (2006) splits the word types into two overlapping subsets according to frequencies, high to medium frequency word types and medium to low frequency word types, and applies Chinese Whispers graph clustering algorithm to partition both of the subsets. One more clustering is done on the overlapping of two subsets to form the final clustering. Using this final clustering as the lexicon a trigram Viterbi HMM with a morphological component used to obtain token based part-of-speech induction.

Goldwater and Griffiths (2007) build on the standard trigram HMM with Dirichlet priors, whose parameters can be fixed or inferred, over the parameters. The system uses a Gibbs sampler to carry out the token based part-of-speech induction. Johnson (2007) adopts a similar approach by using a standard bigram HMM with Dirichlet priors and uses EM, Gibbs sampling and Variational Bayes estimator for token based part-of-speech induction. Graça et al. (2009) employ the bigram HMM, but encourages sparsity by constraining the posterior distributions using posterior regularization framework of Graça et al. (2007). Blunsom and Cohn (2011) use hierarchical Pitman-Yor process priors for a trigram HMM with Gibbs sampling enforcing one tag per word type restriction for training it.

Berg-Kirkpatrick et al. (2010) propose a logistic regression method for state-state and state-emission distributions for a standard HMM. Logistic regression incorporates features to the model and the model is trained with EM algorithm with a gradient based maximization step.

Haghighi and Klein (2006) claim that supervised prototypes are canonical examples of the annotation labels and demonstrate their use in unsuper-

vised learning. Christodoulopoulos et al. (2010) develop further this semi-supervised prototype-driven method. Instead of procuring the prototypes from the gold standard labels of a corpus as in Haghighi and Klein (2006), the prototypes are extracted from the labels generated with part-of-speech induction methods such as Brown clustering.

Das and Petrov (2011) obtain a graph based projection of part-of-speech tags from a labeled corpus in language L_0 to the parallel unlabeled corpus in language L_1. These projected labels are used as features for the model in Berg-Kirkpatrick et al. (2010) to perform part-of-speech induction on the parallel unlabeled corpus in language L_1.

Another way to induce part-of-speech categories involves generation of word representations and clustering to form the categories from them. Some of the word representations in the literature (Lamar et al., 2010b; Maron et al., 2010; Schütze, 1995) are readily applied to the part-of-speech induction problem we do not mention them in this section again. We also apply clustering algorithms, for inducing part-of-speech categories, to the word representations in the literature that we were able to obtain (see Section 4.5).

Christodoulopoulos et al. (2011) present a multinomial mixture model with Dirichlet priors over the mixing weights and use collapsed Gibbs sampling for inference. The system also incorporates features and restricts itself to type based part-of-speech induction.

Lamar et al. (2010a) forms two separate co-occurrence matrices from left and right neighbors of each word type and initial left and right latent descriptors for word types are generated from the rows of the left and right co-occurrence matrices by projection on unit-sphere (i.e. rows are scaled by inverse of their l^2 norms). Using these left and right latent descriptors a labeling is obtained by clustering. These labels are used to obtain new co-occurrence matrices from left and right neighbors' labels of each word type and new latent descriptors are obtained those matrices. Labeling followed by generation of new latent descriptors continue iteratively with weakening learning parameters and the final labeling results in type based part-of-speech induction.

Chapter 3

Procedure

In this chapter, we present the pipeline of operations we carry out to obtain word type, word context and word token representations, both type based and token based part-of-speech induction and the measures we use to gauge their performance.

We begin by describing how we compute the probability distributions of the substitutable word types for word tokens — aptly named the substitute word distribution — which represents the paradigmatic relations of the word tokens, in Section 3.1. Section 3.2 presents the methods we utilize to generate co-occurrence data with the word types and the contexts they are observed in from the substitute word distributions. Section 3.3 details the Euclidean embedding algorithm S-CODE, which inputs the co-occurrence data from the previous section and outputs embeddings in the Euclidean space of the co-occurrence data variables. Section 3.4 reveals how we obtain word type, word context and word token representations from the resulting embeddings. Section 3.5 demonstrates how we perform type based and token based part-of-speech induction. Finally, Section 3.6 explains the measures we employ to assess the performance of the induction.

3.1 Paradigmatic Relations with Substitute Word Distributions

In this thesis, we use the approach of Yuret and Yatbaz (2010) and Yatbaz et al. (2012) to capture the paradigmatic (substitutability based) relations of word tokens with categorical distributions — substitute word distributions — whose outcomes are the word types in our vocabulary. The probability of each word type (outcome) is the probability of observing the word type in place of the target word token (in other words in the context of the word token).

Context of a target word in a sentence is defined as the sequence of words in the window of size $2n-1$ centered at the position of the target word token. The context excludes the target word token. For example, in the sentence "*There is no asbestos in our products now.*", the context of the word token '*asbestos*', for $n = 4$, is '*There is no — in our products*' (where — specifies the position of the target word token).

Assuming the position of the target word token is 0, the context spans from positions $-n+1$ to $n-1$ and the probability of observing each type w in our vocabulary in the context of the target word token is computed using the following equation:

$$
\begin{aligned}
P(w_0 = w|c_{w_0}) &\propto P(w_{-n+1} \dots w_0 \dots w_{n-1}) && (3.1)\\
&= P(w_{-n+1}) \dots P(w_0|w_{-n+1}^{-1}) \dots P(w_{n-1}|w_{-n+1}^{n-2}) && (3.2)\\
&\approx P(w_{-n+1}) \dots P(w_0|w_{-n+1}^{-1}) \dots P(w_{n-1}|w_0^{n-2}) && (3.3)\\
&\propto P(w_0|w_{-n+1}^{-1}) \dots P(w_{n-1}|w_0^{n-2}) && (3.4)
\end{aligned}
$$

Where w_i^j is the word sequence from w_i to w_j (for $i < j$) and c_{w_0} is the context of the target word token at position 0 of length $2n-1$, meaning w_{-n+1}^{-1} to the left and w_1^{n-1} to the right of w_0.

In the Equation 3.1, the right-hand side is proportional to the left-hand side because $P(c_{w_0})$ is independent of any given word w for w_0. With application of the chain rule, Equation 3.2 is obtained from the Equation 3.1.

There	*There* (0.7557), *This* (0.0757), *It* (0.0706)
is	*is* (0.2639), *are* (0.2144), *was* (0.1950)
no	*no* (0.9063), *an* (0.0338), *some* (0.0191)
asbestos	*question* (0.4901), *doubt* (0.1935), *one* (0.0576)
in	*in* (0.5882), *for* (0.0396), *that* (0.0320)
our	*fertilizer* (0.1675), *other* (0.1632), *commercial* (0.0856)
products	*right* (0.2096), *own* (0.1117), *history* (0.0458)
now	*out* (0.1713), *are* (0.0955), *business* (0.0592)
.	. (0.8973), *?* (0.0803), *!* (0.0224)

Figure 3.1: Most probable substitute word types for each context in the sentence "*There is no asbestos in our products now.*". The values in the parentheses are the substitution probability of the word type to the left.

With n^{th}-order Markov assumption, only the closest $n-1$ words in each term of the Equation 3.2 are needed and result is the Equation 3.3. The Equation 3.4 is proportional to the Equation 3.3 because any term that does not depend on w_0 is fixed since the context of the word at position 0 is fixed. Near the boundaries of the sentence, specifically the first and last $n-1$ words, appropriate terms of the Equation 3.4 are truncated or dropped (e.g. if 0 is the first word of a sentence, $P(w_0|w_{-n+1}^{-1})$ becomes $P(w_0)$).

The probabilities required to compute the Equation 3.4 can be obtained from an n-gram language model. Figure 3.1 displays examples of substitute word distributions that capture paradigmatic relations, with three most probable substitute word types per context and their respective probabilities in parentheses.

3.2 Discretization of Substitute Word Distributions

The Euclidean embedding algorithm we employ in this thesis requires its input as co-occurring observations from two categorical variables. We let the word types and the contexts they are observed in to be the categorical variables. We associate each word token with the pair of word type (of the word token) and context (of the word token). A simple method for

representing the word tokens in this manner is to prepare a two column data set where the first column is filled with the word types and second column is filled with respective substitute word distributions as specified in Section 3.1. However, the substitute word distributions we generate are categorical distributions and it is not obvious how we can transform them to a discrete setting.

This section aims to describe our approaches to address discretization of substitute word distributions generated from the contexts. We describe the substitute sampling method of Yatbaz et al. (2012) and present a new discretization approach, nearest neighbors. Both of the techniques are intuitive, preserve the characteristics of the distributions in some way and result in satisfactory performance.

3.2.1 Substitute Sampling

One way to discretize the substitute word distributions is to choose k most probable word types in the distributions. This approach, however, fails to capture the characteristics of the distributions of the word types. For example, consider two different substitute word distributions with exactly same k most probable types, but one of them is a skewed distribution while the other one is a flat distribution. Both of them generate the exact same co-occurrence data, but clearly it is not possible to distinguish the source distribution from the co-occurrence data.

Yatbaz et al. (2012) fix this approach by sampling k word types from the substitute word distributions (with replacement) instead of selecting k most probable word types. This change helps transferring the skewness or the flatness of the source distribution to the co-occurrence data. Figure 3.2 demonstrates the sampling process on the substitute word distributions present in the Figure 3.1.

The k value should be chosen with reason. If it is too small it may affect the system negatively since samples may not accurately represent the distribution they are drawn from (especially valid for flat distributions). A very large k value, on the other hand, won't affect the system negatively or

There	*There*
There	*There*
There	*This*
is	*is*
is	*are*
is	*is*
no	*no*
no	*no*
no	*no*
asbestos	*point*
asbestos	*question*
asbestos	*doubt*
in	*using*
in	*in*
in	*in*
our	*fertilizer*
our	*commercial*
our	*other*
products	*view*
products	*country*
products	*right*
now	*out*
now	*business*
now	*out*
.	.
.	*?*
.	.

Figure 3.2: Sampling thrice from the substitute word distributions in Figure 3.1. Notice that Figure 3.1 only has 3 most probable substitute distributions for typographic purposes while we sampled from the whole distribution, hence the unseen substitute words.

positively (see Section 4.2), but it will increase the computational burden.

3.2.2 Nearest Neighbors

The other approach to discretize a substitute word distribution, that we propose in this thesis, is to implicitly express it in terms of the similar substitute word distributions for the contexts we have in our test corpus. We assign unique identifiers to each context. Then, for each context, we find their nearest neighbors based on the similarity of their substitute word distributions. Examples of nearest neighbor data and the forming of the two column co-occurrence data are presented in Figure 3.3 and 3.4, respectively.

Most intuitive similarity, in this setting, is a distribution similarity function, such as Jensen-Shannon divergence. However, if we consider substitute distributions as vectors in Euclidean space lying on the standard $n-1$ simplex where n is the size of our vocabulary, similarity functions such as cosine similarity or Euclidean distance are viable options as well.

A question that can come to the mind of the reader may be why using the top k nearest substitute word distributions is a good way capture the source distribution while using the most probable k word types is not. Sampling from a distribution is well defined while sampling from a nearest neighbor list is not. We tried to use ad-hoc methods to transform similarities to probability distributions (e.g. take inverse of the similarities and normalize to obtain a distribution, use a normal distribution parameterized on the similarities and a bandwidth value, etc.), but results were not as strong as using top k nearest neighbors.

Once again k value should be chosen with care, but for different reasons than sampling discretization. If it is too small it can affect the system negatively since nearest neighbors may not sufficiently represent the distribution they are close to (see Section 4.2). A very large k value will also affect the system negatively. In the limit, k will be equal to the number of word tokens in the test set, meaning that every context will be a nearest neighbor of every other context. This clearly reduces the descriptive power of the nearest neighbors because many of the neighbors are not actually similar to each

There	215 (0), 395558 (0), 147918 (5.286e-05)
is	216 (0), 865116 (0), 66458 (0)
no	217 (0), 900787 (1.626e-01), 1130405 (1.967e-01)
asbestos	218 (0), 76648 (4.369e-01), 1134041 (4.649e-01)
in	219 (0), 416560 (3.377e-01), 98543 (3.391e-01)
our	220 (0), 1074457 (6.240e-01), 230752 (6.739e-01)
products	221 (0), 791857 (6.871e-01), 296915 (6.871e-01)
now	222 (0), 888828 (6.656e-01), 495685 (6.949e-01)
.	223 (0), 53448 (3.298e-04), 27774 (3.888e-03)

Figure 3.3: Most similar substitute word distribution IDs in the test corpus for each context in the sentence "*There is no asbestos in our products now.*", according to Jensen-Shannon divergence. The values in the parentheses are the similarity score (smaller is better) of the ID to the left. We assumed 'There' has the context ID of 215 and rest of the word tokens have appropriate context IDs in increasing order. Notice that each context is the nearest neighbor of itself with score of 0.

other at all.

3.3 Spherical Co-Occurrence Data Embedding

In this thesis, we make use of the Spherical Co-Occurrence Data Embedding (S-CODE) (Maron et al., 2010), which is an extension to the Symmetric Interaction Model of the Co-occurrence Data Embedding (CODE) (Globerson et al., 2007), to map co-occurrence data we generate from the word types and substitute word distributions in Section 3.2 to d dimensional Euclidean space to cluster later in the process. This section is a short review of the CODE and S-CODE frameworks. For illustrative purposes, Figure 3.5 depicts an artificial co-occurrence data and its embedding on a 2 dimensional unit sphere using the S-CODE.

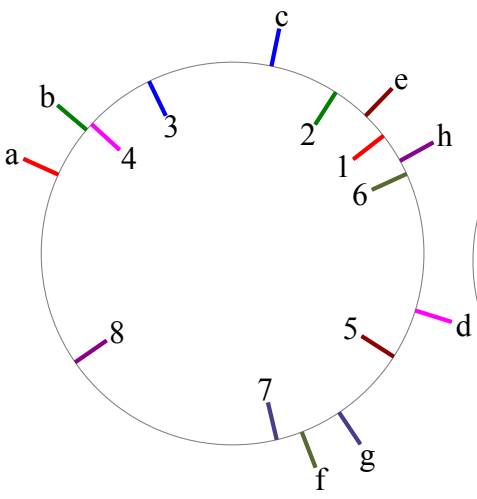
Let X and Y be two categorical variables with finite cardinality $|X|$ and $|Y|$. We observe a set of pairs $\{x_i, y_i\}_{i=1}^{n}$ drawn IID from the joint distribution of X and Y. These pairs are summarized by the empirical distributions $\bar{p}(x, y)$, $\bar{p}(x)$ and $\bar{p}(y)$. The idea is to find such embeddings $\phi(x)$ and $\psi(y)$, for each unique variable x and y, that reflect the statistical relation-

There	215
There	395558
There	147918
is	216
is	865116
is	66458
no	217
no	900787
no	1130405
asbestos	218
asbestos	76648
asbestos	1134041
in	219
in	416560
in	98543
our	220
our	1074457
our	230752
products	221
products	791857
products	296915
now	222
now	888828
now	495685
.	223
.	53448
.	27774

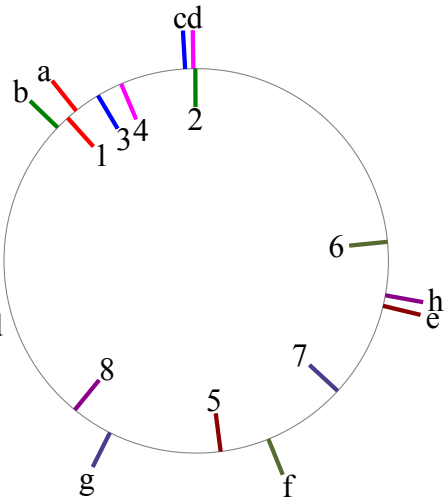
Figure 3.4: Selection of 3 nearest neighbors from Figure 3.3 for each word token.

X	Y
a	1
a	4
b	1
b	2
c	2
c	3
d	3
d	4
e	5
e	6
f	5
f	7
g	5
g	8
h	6
h	7

(a)



(b)



(c)

Figure 3.5: (3.5a) An artificial co-occurrence data set. X variable generates letters from a to h, while Y variable generates digits from 1 to 8. Notice that a–d are observed with 1–4 while e–h are observed with 5–8. (3.5b) Initially all embeddings are distributed randomly on the unit circle. (3.5c) Once the system converges embeddings of x and y pairs observed together are much closer to each other than pairs that are not observed together, demonstrating the attraction and repulsion forces.

ship between the variables x and y in terms of square of Euclidean distance $d^2_{x,y} = \|\phi(x) - \psi(y)\|^2$, meaning that pairs occur together frequently embedded close to each other in d dimensional space.

Globerson et al. (2007) describes a number of models to capture the relationship between the joint distributions and distances. In this thesis, we use the model extended by Maron et al. (2010):

$$p(x,y) = \frac{1}{Z}\bar{p}(x)\bar{p}(y)e^{-d^2_{x,y}} \tag{3.5}$$

where $Z = \sum_{x,y}\bar{p}(x)\bar{p}(y)e^{-d^2_{x,y}}$ is the normalization term. We can express the log-likelihood of the joint distribution over all embeddings ϕ and ψ as the following:

$$\begin{aligned}
\ell(\phi,\psi) &= \sum_{x,y}\bar{p}(x,y)\log p(x,y) \\
&= \sum_{x,y}\bar{p}(x,y)(-\log Z + \log\bar{p}(x)\bar{p}(y) - d^2_{x,y}) \\
&= -\log Z + const - \sum_{x,y}\bar{p}(x,y)d^2_{x,y} \qquad (3.6)
\end{aligned}$$

The gradient of the log-likelihood depends on the sum of embeddings $\phi(x)$ and $\psi(y)$, for $x \in X$ and $y \in Y$, and to maximize the log-likelihood, (Maron et al., 2010) use a gradient-ascent approach. The gradient is as follows:

$$\frac{\partial\ell(\phi,\psi)}{\partial\phi(x)} = \sum_{y} 2\bar{p}(x,y)[\psi(y) - \phi(x)] + \frac{1}{Z}\sum_{y}\bar{p}(x)\bar{p}(y)[\phi(x) - \psi(y)]e^{-d^2_{x,y}} \tag{3.7}$$

$$\frac{\partial\ell(\phi,\psi)}{\partial\psi(y)} = \sum_{x} 2\bar{p}(x,y)[\phi(x) - \psi(y)] + \frac{1}{Z}\sum_{x}\bar{p}(x)\bar{p}(y)[\psi(y) - \phi(x)]e^{-d^2_{x,y}} \tag{3.8}$$

The first sum, the gradient of the part with $d^2_{x,y}$ in (3.6), in (3.7) [(3.8)] acts attraction force between the $\phi(x)$ ($\psi(y)$) and all the embeddings ψ (ϕ) in proportion to respective joint empirical probabilities $\bar{p}(x,y)$.

The second sum, the gradient of $-\log Z$ in (3.6), in (3.7) [(3.8)] acts a repulsion force between the $\phi(x)$ ($\psi(y)$) and all the embeddings ψ (ϕ) in

proportion to respective marginal empirical probabilities $\bar{p}(x)$ and $\bar{p}(y)$.

Additionally (Maron et al., 2010) restricts all embeddings ϕ and ψ to lie on the d dimensional unit sphere, hence the name S-CODE. This restriction, justified by a coarse approximation in which all ϕ and ψ distributed uniformly and independently on the sphere, enables Z to be approximated by a constant value. This saves us from re-computation of Z after every so many steps which has the computational complexity $O(LRd)$, where L and R are the number of unique observations at the left and the right column of the co-occurrence data, respectively, and d is number of dimensions of the Euclidean embedding space.

For the experiments in the thesis, we use S-CODE with sampling based stochastic gradient ascent, smoothly decreasing learning rates φ_0 and η_0, a constant approximation of Z and randomly initialized ϕ and ψ vectors.

3.4 Word Representations

Once S-CODE converges, we have ϕ embeddings (vectors) for the word types and ψ embeddings for the discretizations (substitute word types or context IDs) of the substitute word distributions of the word tokens. As it will be explained in the next section (3.4.1), Yatbaz et al. (2012) present a straightforward way to obtain word type representations from the ϕ vectors. On the other hand, word context and word token representations are not as obvious as word type representations to acquire from the ϕ and ψ vectors. We propose two approaches for word context representations and two approaches for word token representations in Section 3.4.2 and Section 3.4.3, respectively. Table 3.1 is a short summary of all the representations we describe in this section.

3.4.1 Word Type Representations

Both discretization methods described in Section 3.2 generate co-occurrence data that has the word types in the test corpus in their left columns. Leveraging this observation, Yatbaz et al. (2012) utilize ϕ vectors produced by

S-CODE as word type representations. These representations are not able provide multiple representations for ambiguous word types, since there is only one vector for each word type in our test corpus and it is not possible to obtain different representations for the tokens of the same word type. In the remainder of this thesis, we refer the word type representations obtained this way as the X ***vectors***.

3.4.2 Word Context Representations

First of the two approaches we propose to obtain word context representations is applicable only to the case of nearest neighbor discretization discussed in Section 3.2.2. The reason for this method not being applicable to the sampling discretization is that sampling discretization does not generate items at the right column that uniquely identifies the word contexts.

As pointed out in the previous section each word type is associated with a ϕ vector. Symmetrically, this means each context ID is associated with a ψ vector. As each word context in the test corpus is associated with a context ID, we can just use the ψ vectors as a word context representations. We refer the word token representations obtained this way as the Y ***vectors***.

The second approach we propose is applicable to both of the discretization methods we previously discussed. It can also be utilized for any discretization method that may be conceived in future.

We observe that each substitute word distribution for a word context in the test corpus is discretized k times in the co-occurrence data which means there are k ψ vectors (which are not distinct, if the substitute word distribution's discretizations are not distinct) corresponding to each word context. For each word context, we sum k ψ vectors of the word context and scale the resulting vector to a unit vector which we determine as the representation of that word context. We refer to the word context representations obtained this way as the $\overline{Y}$ ***vectors***.

Name	Representation Type	Summary
X vectors	Word type	ϕ embeddings of S-CODE
Y vectors	Word context	ψ embeddings of S-CODE
$\overline{Y}$ vectors	Word context	Sum of ψ embeddings of S-CODE
XY vectors	Word token	Concatenation of X and Y vectors
$\overline{XY}$ vectors	Word token	Concatenation of X and $\overline{Y}$ vectors

Table 3.1: Summary of word representations.

3.4.3 Word Token Representations

In this thesis, we assume that word token representations capture characteristics of both the type of the word token and the context of the word token. Since in the previous Sections 3.4.1 and 3.4.2 we obtained representations for word types and word contexts, respectively, we combine these representations to obtain the word token representations capturing the characteristics of the word types and word contexts.

For each word token we simply concatenate word type representation for the type of the token and word context representation for the context of the token to form a representation for the word token. Note that by concatenation we mean to create a new vector in $\mathbb{R}^{n+m}$ from two vectors in $\mathbb{R}^n$ and $\mathbb{R}^m$. We refer to the concatenation of X vectors and Y vectors the as XY ***vectors*** and the concatenation of X vectors and $\overline{Y}$ vectors as the $\overline{XY}$ ***vectors***.

3.5 Part-of-Speech Induction

The simplest way to induce part-of-speech categories with the representations we generated so far, is to cluster the representations with some clustering algorithm such as k-means clustering. Clustering of the word type representations (X vectors) results in type based part-of-speech categories. We refer the part-of-speech induction with X vectors as the X clusters. Clustering of word context representations (Y or $\overline{Y}$ vectors) and word token representations (XY or $\overline{XY}$ vectors) result in token based part-of-speech categories. We refer the part-of-speech induction with Y and XY vectors as the Y and

Name	Induction Type	Summary
X clustering	Type based	Clustering of X vectors
Y clustering	Token based	Clustering of Y vectors
$\overline{Y}$ clustering	Token based	Clustering of $\overline{Y}$ vectors
XY clustering	Token based	Clustering of XY vectors
$\overline{XY}$ clustering	Token based	Clustering of $\overline{XY}$ vectors
Y_V clustering	Token based	Cluster ψ embeddings first, then vote for each context
XY_V clustering	Token based	Cluster concatenation of ϕ and ψ embeddings first, then vote for each context

Table 3.2: Summary of part-of-speech induction methods.

XY clusters, respectively (induction with $\overline{Y}$ and $\overline{XY}$ vectors are referred to similarly).

We propose two additional methods to induce part-of-speech categories that are not involved with the word representations. Each word context in the test corpus is associated with k ψ vectors, as explained with the $\overline{Y}$ vectors in Section 3.4.2. Instead of aggregating the ψ vectors, we simply cluster them and end up with k cluster identities for each word context. We, then, determine the cluster identity of the each word context by determining the most common identity (i.e. voting) in the k cluster identities of the word context. If there are ties we break them randomly. Once each word context has a cluster identity we end up with a token based part-of-speech induction. We refer this part-of-speech induction as the Y_V clustering. Similar to obtaining XY vectors from the X and Y vectors, we can also extend Y_V induction by concatenating appropriate ϕ and ψ vectors, clustering them and determining most common cluster identity for each word context. We refer this part-of-speech induction as the XY_V clustering. Table 3.2 is a short summary of all the induction methods we describe.

3.6 Evaluation Measures

In the part-of-speech induction problem, each word token in the test corpus is associate with a gold-standard part-of-speech tag and an induced cluster. We try to evaluate whether induced clusters have similar structures as the gold-standard labels. We use two of the popular methods in the part-of-speech induction literature to conduct the evaluation. These are Many-To-One mapping accuracy and V-Measure and they are detailed in the following sections.

3.6.1 Many-To-One Mapping Accuracy

Many-To-One (MTO) mapping accuracy (Christodoulopoulos et al., 2010) (also known as *cluster purity*) first finds a mapping from clusters to gold standard classes. Each cluster is mapped to the most common class within that cluster. As the name suggests it is possible for more than one cluster to be mapped to the same class. Once the clusters are mapped to the classes, the cluster sequence is transformed to class sequence simply by replacement and accuracy to the gold standard class sequence is computed in straightforward manner.

MTO accuracy yields higher scores as the number of clusters increases. The extreme case of this behavior can be observed when a unique cluster is assigned to each instance in the data set. MTO accuracy rewards this intuitively bad clustering with a perfect score, since each cluster is mapped to the right class. In this thesis, we kept our number of clusters same as the number of classes to prevent confusing increases in the MTO accuracy score.

3.6.2 V-Measure

V-Measure (VM) (Rosenberg and Hirschberg, 2007) is an entropy based cluster evaluation measure that is defined as the weighted harmonic mean of two criteria called homogeneity and completeness. Homogeneity is defined as:

$$h = 1 - \frac{H(C|K)}{H(C)} \tag{3.9}$$

Where $C = \{c_i | i = 1, \ldots, n\}$ is the set of gold standard classes, $K = \{k_j | j = 1, \ldots, m\}$ is the set of clusters, $H(C)$ is the entropy of the classes and $H(C|K)$ is the conditional entropy of the classes conditioned on the clusters. Let N be the number of instances in the data set and A_{nm} be the matrix whose element a_{ij} equals the number of instances that belong to the class c_i and in the cluster k_j such that $\sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij} = N$. Then probability distributions needed to compute $H(C)$ and $H(C|K)$ are defined as following:

$$P(c_i) = \frac{\sum_{j=1}^{m} a_{ij}}{N} \tag{3.10}$$

$$P(k_j) = \frac{\sum_{i=1}^{n} a_{ij}}{N} \tag{3.11}$$

$$P(c_i, k_j) = \frac{a_{ij}}{N} \tag{3.12}$$

Homogeneity is related to pureness of the clusters. A cluster gets purer as the classes it contains lessen in numbers and majority of the cluster members belong to a single class. In other words, homogeneity wants to have clusters that have few classes.

Completeness is also defined similarly:

$$c = 1 - \frac{H(K|C)}{H(K)} \tag{3.13}$$

Completeness can be interpreted as the distribution of the classes to the clusters. As members of a class appears in more clusters it gets more distributed and completeness suffers. Completeness wants to keep members of a class together in the same cluster.

Finally VM is defined in terms of homogeneity and completeness:

$$VM = \frac{(1 + \beta)hc}{(\beta h) + c} \tag{3.14}$$

Where β is weighting factor and usually set to 1, as in this thesis, making the VM measure harmonic mean of homogeneity and completeness. By combining these criteria, VM does not suffer the problem of large cluster numbers that troubles MTO accuracy.

Chapter 4

Experiments

The aim of this chapter is to detail the data used, explain some algorithm and parameter choices made and compare our system's performance in the part-of-speech induction problem to the other vector representations and part-of-speech induction systems mentioned in Chapter 2. We also analyze system's sensitivity to parameter changes. Additionally, we introduce morphological and orthographic features to our system which affect our overall performance.

The structure of the chapter is as follows: Section 4.1 states the corpora and algorithms used to conduct the experiments in this thesis. Section 4.2 begins by reporting the results of our experiments for the procedure in Chapter 3. It goes on to detail the incorporation of morphological and orthographic features to the system and reports the results for additional experiments with these features. Section 4.3 demonstrates the S-CODE algorithm's sensitivity to the parameters. Section 4.4 presents a measure of ambiguity of word types and analyze the ability of the word context and word token representations to reflect ambiguity in the test corpus. Finally, Section 4.5 compares part-of-speech induction performance of other word representations we were able to obtain to our word representations.

4.1 Experimental Settings

The test corpus we used is the Wall Street Journal (WSJ) Section of Penn Treebank (Marcus et al., 1999) (PTB). It has 1,173,766 tokens and 49,206 types, which are tagged with a total of 45 different part-of-speech tags. We used these as the gold-standard tags to evaluate our induced tags.

The training corpus we used to generate the language model for computing the paradigmatic relations of the word tokens in the test corpus as substitute word distributions is the Wall Street Journal data (1987-1994) extracted from CSR-III Text (Graff et al., 1995) (excluding the sections of the PTB) which has about 126 millions of tokens. We used SRILM (Stolcke, 2002) to build a 4-gram language model with Kneser-Ney discounting. Types that are observed less than 20 times in the training corpus were replace by '<unk>', which limits the vocabulary size to 78,498. The perplexity of the 4-gram language model on the test corpus is 96.

Both test corpus and training corpus tokens kept with original capitalization.

For computational efficiency, instead of computing probability of substitutability for word types in each context as proposed in Section 3.1, we used the FASTSUBS algorithm (Yuret, 2012) to compute the unnormalized probabilities of the 100 most probable substitute words in each context and later normalized them to obtain well-formed distributions. While the normalization is not necessary to sample from them since there are algorithms to sample from unnormalized distributions (Section 3.2.1), it is necessary to compute distances between the distributions (Section 3.2.2).

Sampling from the distributions is straightforward as explained in Section 3.2.1. We generated co-occurrence data of 37,560,512 pairs by sampling 32 substitute words for each word token. In the case of nearest neighbor discretization, we generated co-occurrence data of 18,780,256 pairs by computing 16 nearest neighbors of each context. To compute nearest neighbors of each distribution we used Jensen-Shannon divergence (also known as total divergence to the average (Dagan et al., 1997)) which is a symmetrized and smoothed version of Kullback-Leibler divergence.

The learning rate parameters φ_0 and η_0 of the S-CODE algorithm were set to 50 and 0.2, respectively. The approximation $\tilde{Z}$ to the normalization term Z was chosen to be 0.166. The number of dimensions of the embedding space was set to 25. The S-CODE iterations over the input continues until the log-likelihood difference between consecutive iterations are smaller 0.001, which means ratio of likelihoods is approximately 1.002 or less.

To obtain the word type and word token representations as proposed in the Sections 3.4.1 and 3.4.3, respectively, we used a computationally efficient k-means algorithm (Elkan, 2003) that is modified to use smart initialization (Arthur and Vassilvitskii, 2007), handle duplicate vectors efficiently, optimize the root mean square (rms) of cluster members and randomly restart a number of times to find an initialization configuration that yields a lower rms. The number of restarts we used to cluster $\phi(x)$ vectors of sampling and nearest neighbors and cluster $\psi(y)$ of sampling methods are 128. For the rest we used 8 restarts, because the number of vectors to cluster are much larger.

In order to account for the fluctuations inherent in the algorithms using random number generators, each experiment is repeated 10 times with different random number generator seeds. We report the results in terms of average scores with their respective standard deviations indicated in parentheses next to them.

4.2 Results

In this section we report the performance of our system for the task of part-of-speech induction in terms of MTO accuracy and VM scores for the methods detailed in Section 3.5 under two subsections. First, we report our results of the experiments that follow the exact procedure detailed in Chapter 3 and compare them to the other part-of-speech induction systems that do not incorporate morphological and orthographic features. Following that, we introduce morphological and orthographic features to S-CODE, re-run the experiments and report the results with comparisons to the part-of-speech induction systems with feature components.

4.2.1 Results for Co-occurrence of Word Tokens and Their Contexts

Table 4.1 lists all the average MTO accuracy and VM scores for our experiments and models in the literature which do not incorporate features (such as morphological and orthographic features). Our X and XY based clusters for both sampling and nearest neighbors discretizations outperform the models in the literature. Our Y based clusters are on par with the lower end of the literature.

We observe that type based part-of-speech induction slightly outperform token based ambiguous part-of-speech inductions. Sampling X clusters results in average 0.7665 (0.0079) MTO accuracy and 0.6817 (0.0043) VM scores. Similarly, nearest neighbors X clusters result in average 0.7637 (0.0076) MTO accuracy and 0.6791 (0.0061) VM scores. No Y based cluster manages to exceed the average MTO accuracy score of 0.66 and VM score of 0.51. XY based clusters perform somewhere between X and Y based clusters and are closer to the performance of X.

4.2.2 Results for Co-occurrence of Word Tokens, Their Contexts and Their Morphological and Orthographic Properties

Clark (2003) demonstrates that using morphological features, which have information about the morphemes that form a word type, and orthographic features, which have information about the characters (including the numerical and punctuation characters) that form a word type, significantly improves part-of-speech induction with an HMM based model. To obtain word representations that improve part-of-speech induction, we extract and use these types of features as co-occurrence data that is the input of S-CODE algorithm, similar to Yatbaz et al. (2012).

We use morphological features generated with the unsupervised morphology induction software Morfessor (Creutz and Lagus, 2005). Morfessor was trained on the test corpus using the default settings, except for the perplex-

Model	MTO	VM
Brown et al. (1992)*	0.6776	0.6299
Goldwater and Griffiths (2007)*	0.6646	0.5821
Johnson (2007)*	0.5024	0.4919
Graça et al. (2009)*	0.6247	0.5479
Maron et al. (2010)	0.688 (0.0016)	-
Lamar et al. (2010a)	0.708	-
Lamar et al. (2010b)	0.660	-
Substitute Sampling $X^\dagger$	**0.7665 (0.0079)**	**0.6817 (0.0043)**
Substitute Sampling $\overline{Y}$	0.6448 (0.0039)	0.4999 (0.0021)
Substitute Sampling $\overline{XY}$	0.7346 (0.0102)	0.6468 (0.0081)
Substitute Sampling Y_V	0.6346 (0.0032)	0.4855 (0.0027)
Substitute Sampling XY_V	0.7009 (0.0109)	0.6004 (0.0113)
Nearest Neighbors X	**0.7637 (0.0076)**	**0.6791 (0.0061)**
Nearest Neighbors Y	0.6551 (0.0080)	0.5142 (0.0030)
Nearest Neighbors XY	0.7496 (0.0099)	0.6627 (0.0070)
Nearest Neighbors $\overline{Y}$	0.6441 (0.0052)	0.5042 (0.0029)
Nearest Neighbors $\overline{XY}$	0.7395 (0.0065)	0.6476 (0.0051)
Nearest Neighbors Y_V	0.6419 (0.0067)	0.4991 (0.0025)
Nearest Neighbors XY_V	0.7428 (0.0068)	0.6483 (0.0047)

Table 4.1: Summary of the results in Section 4.2.1 compared to the previous works on WSJ section of PTB in terms of MTO mapping accuracy and VM scores. Entries with $\star$ are reported in Christodoulopoulos et al. (2010). Entry with † is replication of an experiment in Yatbaz et al. (2012).

ity threshold which was chosen to be 300. The software induced 5 unique suffixes and they are observed in 10,484 word types out of the 49,026 in the test corpus. The suffixes are added to the system as co-occurrence data by being paired with word types. For every occurrence of a word type w at left column of the co-occurrence data that has suffixes s_i for $i = 1, \ldots, k$, we add one w s_i pair for each i to the co-occurrence data. We don't utilize the ψ embeddings of the suffixes formed by S-CODE for obtaining the word token representations or inducing the part-of-speech categories as described in the Section 3.4 and Section 3.5, respectively.

The orthographic features we use are similar to the ones in Berg-Kirkpatrick et al. (2010) with minor differences:

Contains Hyphen This feature is active for the lowercase word types with internal hyphen.

Initial Apostrophe This feature is active for the word types that start with an apostrophe.

Initial Capital This feature is active for the word types with their initial letter is a capital letter, unless the word type is observed as the first token of the sentence it belongs.

Number This feature is active for the word types that start with a digit.

The orthographic features are introduced to the system in the same way as the morphological features. For every occurrence of a word type w at left column of the co-occurrence data that has the orthographic features o_i for $i = 1, \ldots, k$, we add one w o_i pair for each i to the co-occurrence data. As with the morphological features, we don't utilize the ψ embeddings of the orthographic features. Figure 4.1 demonstrates the incorporation of the morphological and orthographic features to the system for the sentence "By 1997, almost all remaining uses of cancer-causing asbestos will be outlawed.".

Addition of the morphological and orthographic features to the co-occurrence data, without modifying the original experimental settings, improve every experiment reported in the Section 4.2 between 0.01 and 0.06 in terms of both

By	April
1997	mid-1991
1997	/N/
,	,
almost	almost
all	all
remaining	remaining
remaining	/SUF:ing/
uses	uses
uses	/SUF:s/
of	of
cancer-causing	the
cancer-causing	/CH/
asbestos	pesticides
will	should
be	be
outlawed	lost
outlawed	/SUF:ed/
.	.

Figure 4.1: The co-occurrence data for the sentence "By 1997, almost all remaining uses of cancer-causing asbestos will be outlawed.". The contexts are discretized with sampling (Section 3.2.1). The morphological features added to the right column in the form /SUF:s/ where s is a morpheme. The orthographic features added to the right column in the form /o/ where o is an identifier for the features. The features /N/ and /CH/, in this example, stands for 'Number' and 'Contains Hyphen' features, respectively.

average MTO and average VM score. We observe that the least improved are the word context (Y) based induction, while the most improved induction are the word token (XY) based induction and the word type (X) induction are somewhere in between. Feature introduced sampling and nearest neighbor methods with X clusters result in 0.8009 (0.0077) and 0.8025 (0.0048) MTO accuracy, respectively, and improve the state-of-the-art in the word type part-of-speech induction. Feature introduced nearest neighbor method with XY clusters result in 0.8039 (0.0036) MTO accuracy and improve the state-of-the-art in the token based part-of-speech induction.

Table 4.2 lists all the average MTO accuracy and VM scores for our experiments with features and other feature incorporating models in the literature. Once again our X and XY based clusters for both sampling and nearest neighbors discretizations outperform the models in the literature. Our Y based clusters, on the other hand, are not comparable to the feature incorporating systems of the literature because they only are influenced by the contexts not the target words.

4.3 Analysis of Parameters

In this section we analyze the performance of our algorithm with respect to parameter choices. The parameters we investigate are the number of discretizations, the number of embedding dimensions and the constant approximation of Z. We keep all other experimental settings fixed and only modify the parameter of interest. For the sake of clarity and preventing appearance of redundant graphs over and over again, we perform all the parameter experiments only on the X clustering of sampling discretization without features, with the exception of number of nearest neighbor experiments. Other configurations result in similar behavior and are not reported.

Figure 4.2 and Figure 4.3 plot the number of discretizations versus the evaluation measures for sampling and nearest neighbors discretizations for X clustering, respectively. In Figure 4.2, we observe that sampling discretization of this test corpus is quite robust to the number of samples, with small performance loss in low sample scenarios. In Figure 4.3, we see that very

Model	MTO	VM
Clark (2003)*	0.7119	0.6555
Berg-Kirkpatrick et al. (2010)	0.755	-
Christodoulopoulos et al. (2010)	0.761	0.688
Christodoulopoulos et al. (2011)	0.728	0.661
Blunsom and Cohn (2011)	0.775	0.697
Substitute Sampling + Features $X^\dagger$	**0.8009 (0.0077)**	**0.7217 (0.0041)**
Substitute Sampling + Features $\overline{Y}$	0.6586 (0.0047)	0.5123 (0.0023)
Substitute Sampling + Features $\overline{XY}$	0.7872 (0.0068)	0.6945 (0.0052)
Substitute Sampling + Features Y_V	0.6505 (0.0035)	0.4984 (0.0023)
Substitute Sampling + Features XY_V	0.7533 (0.0083)	0.6551 (0.0082)
Nearest Neighbors + Features X	**0.8025 (0.0048)**	**0.7209 (0.0046)**
Nearest Neighbors + Features Y	0.6840 (0.0043)	0.5323 (0.0027)
Nearest Neighbors + Features XY	**0.8039 (0.0036)**	0.7113 (0.0025)
Nearest Neighbors + Features $\overline{Y}$	0.6688 (0.0060)	0.5245 (0.0046)
Nearest Neighbors + Features $\overline{XY}$	0.7906 (0.0077)	0.6963 (0.0056)
Nearest Neighbors + Features Y_V	0.6677 (0.0033)	0.5169 (0.0025)
Nearest Neighbors + Features XY_V	0.7923 (0.0087)	0.6996 (0.0071)

Table 4.2: Summary of the results in Section 4.2.2 compared to the previous works incorporating features on WSJ section of PTB in terms of MTO mapping accuracy and VM scores. Entry with $\star$ is reported in Christodoulopoulos et al. (2010). Entry with † is replication of an experiment in Yatbaz et al. (2012).

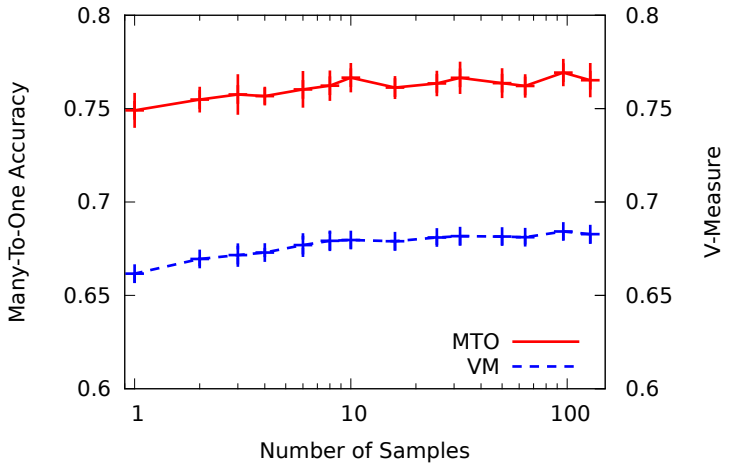


Figure 4.2: Number of samples from substitute word distributions vs. MTO and VM scores. The system is fairly robust to the change in number of samples.

small number of nearest neighbors perform poorly. As the number of nearest neighbors increase, performance quickly picks up and reaches to stability.

Figure 4.4 shows that once the number of embedding dimensions is around 10, performance of the system, for both discretizations, fluctuates within 0.01 MTO accuracy and there is no apparent gain of using more than 25 dimensions.

Figure 4.5 shows that the constant approximation of Z, $\tilde{Z}$, can vary almost an order of magnitude in both directions of the value we choose without significant loss in the MTO accuracy. Maron et al. (2010) shows that uniformly distributed vectors on a 25 dimensional sphere result in the expected $Z \approx 0.146$. In the experiments, the real Z value is always observed between 0.14 and 0.17. When the approximation of Z is too small, the attraction forces in Equations 3.7 and 3.8 dominate the system and vectors tend to converge to a single point. On the other hand, if $\tilde{Z}$ is too large attraction forces become very weak and vectors are not able form into clusters.

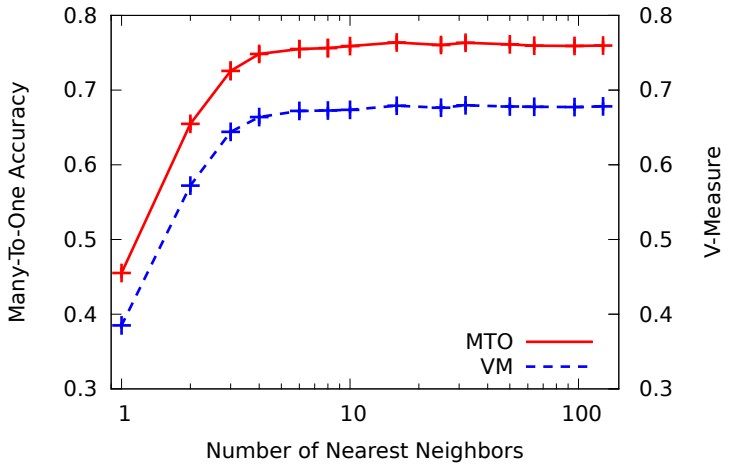


Figure 4.3: Number of nearest substitute word distribution neighbors vs. MTO and VM scores. Small number of neighbors affects the system negatively.

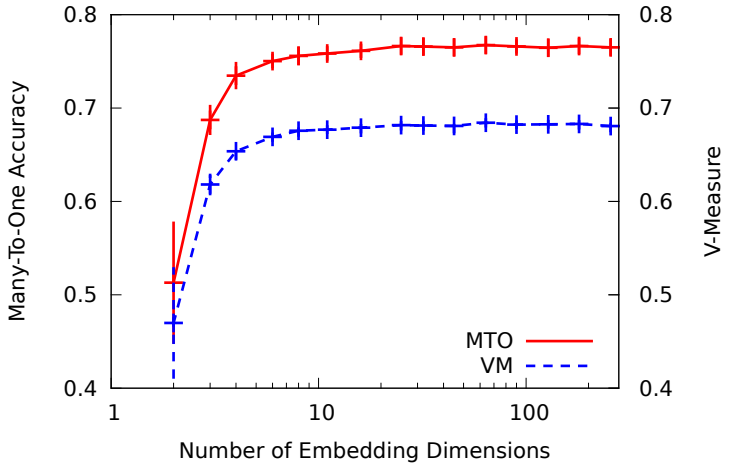


Figure 4.4: Number of embedding dimensions vs. MTO and VM scores. Small number of embedding dimensions affects the system negatively.

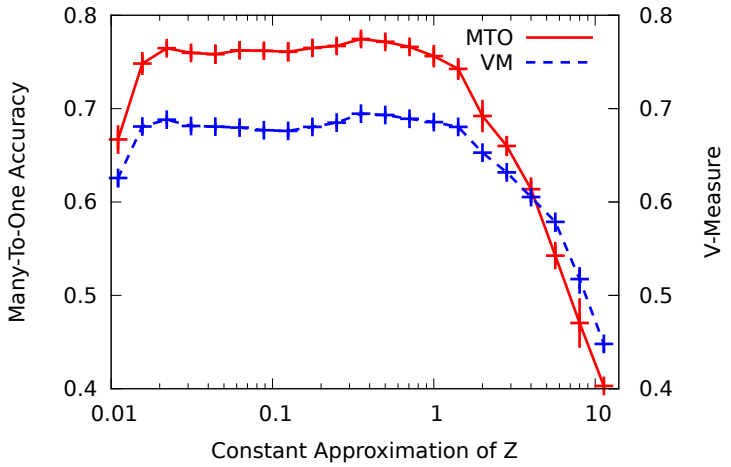


Figure 4.5: Constant approximation of Z vs. MTO and VM scores. Approximations varying almost one order of magnitude around our approximation of 0.166 do not significantly alter the performance.

4.4 Analysis of Word Token Representations

As performance of the word context representations for both sampling and nearest neighbor discretization is inferior in the part-of-speech induction task compared to the word type representations, it is a sensible concern whether the strong performance of the word token representations is a result of being too similar to the word type representations, meaning that word token representations are not able to deal with the ambiguity in the word types as we claim. To appropriately address this concern, we first define a measure called Gold-standard-tag Perplexity (GP) of a word type w, which we use to determine how ambiguous the word type in the test corpus is, as the following:

$$GP(w) = 2^{H(P_w)} = 2^{-\sum_{t \in T} P_w(t) log_2 P_w(t)} \tag{4.1}$$

where T is gold-standard part-of-speech tag set that word type w is observed with, P_w is the probability distribution of tags $t \in T$ with the word type w and $H(P_w)$ is the entropy of the probability distribution P_w. The gold-standard-tag perplexity relates how often a word type is associated with different part-of-speech categories in the test corpus. A word type w_i that is

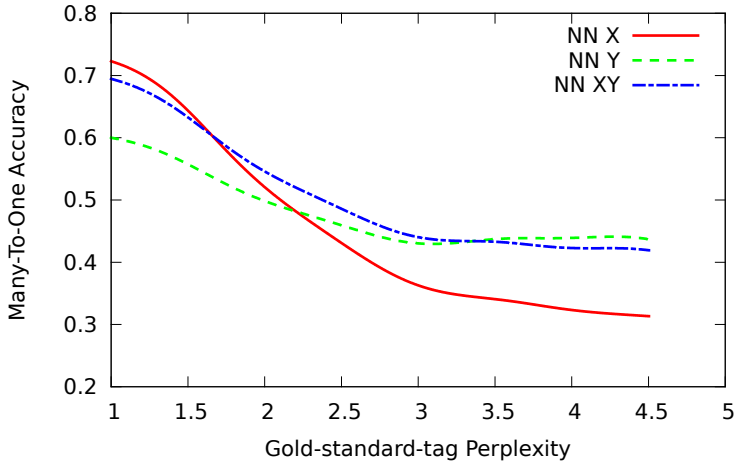


Figure 4.6: Regression lines for X, Y and XY clusters of the nearest neighbor (NN) discretization on the MTO accuracy vs. GP plot for the training corpus.

only observed with the part-of-speech category t_i throughout the test corpus is clearly unambiguous in terms its syntactic category and will have a GP of 1. On the other hand, a word type w_j that co-occurs with the tag t_k in the half of the observations and with the tag t_l in the other half is ambiguous and will have a GP of 2. As the ambiguity of a word type increases GP measure of the word type increases as well.

In order to demonstrate that word context and word token representations' part-of-speech induction outperforms word type representations' part-of-speech induction for the ambiguous word types, we plot the gold-standard-tag perplexity versus the smoothed MTO accuracy. To compose the plot, first, we find the mapping from induced clusters to gold-standard tags, just as we do for the MTO accuracy. Then, we compute the GP and the MTO accuracy for each word type using the mapping. Finally, we utilize the Nadaraya-Watson kernel regression estimate with normal kernel of bandwidth 1.0 to obtain smooth regression lines for each induction method. We create the plots only for X, Y and XY clusters of the nearest neighbor discretization, but other methods also follow similar trends. Figure 4.6 presents the said plot and Figure 4.7 presents the plot for the nearest neighbor discretization with addition of morphological and orthographic features.

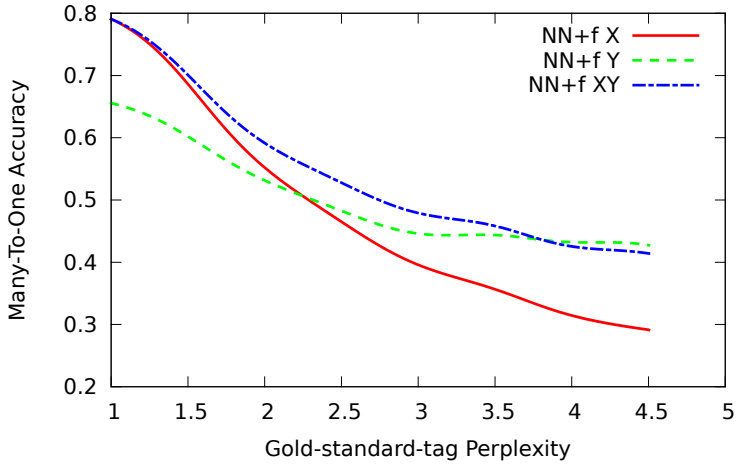


Figure 4.7: Regression lines for X, Y and XY clusters of the nearest neighbor (NN) discretization with features on the MTO accuracy vs. GP plot for the training corpus.

The plot in the Figure 4.6 shows that as the ambiguity of the word types increase, using word context and word token representations yields better performance, however, they compromise the performance on the unambiguous types. Since only 14.94% of the tokens in our test corpus consists of word types with GP greater than 1.5 (a word type observed with one T_0 tag for every six observation with T_1 tag has a GP of 1.507) and 45.71% consists of word types with GP exactly 1, improvement on the ambiguous word types does not recover the loss in the unambiguous word types. The second plot in the Figure 4.7 shows that addition of morphological and orthographic features makes the regression lines steeper, meaning that the MTO accuracy jumps for the unambiguous types and it slightly dips for the ambiguous types for all word representations. This makes sense, because if a feature is active for a word type, it is active regardless of the context the word type is in, with the exception of the 'Initial Capital' feature.

4.5 Comparison to Other Word Representations

In this section, we compare the performance of our word representations to the other word representations in the literature. Representations we were able to obtain are conceived by Collobert and Weston (2008), Mnih and Hinton (2009), Mikolov et al. (2010) and Huang et al. (2012) and they are outcomes of neural network language models. All of the representations are word type representations. Huang et al. (2012) are also able to generate word context representations by taking the weighted sum of the word type representations in the context window of target words[1]. For each set of representations, we determine the intersection of the word types in the studies with the word types of our test corpus[2] and apply the k-means algorithm of the previous sections to the word type representations in the intersection. We set the number of clusters same as the number gold-standard tags left in the intersection. After the clustering we compute MTO mapping accuracy and VM scores. We used nearest neighbor based word type (X) and word token (XY) representations of our work for comparisons, with and without morphological and orthographic features. We repeat each experiment 10 times to account for fluctuations in the k-means algorithm and report the standard deviation of scores in parentheses.

Table 4.3 presents all the results for other word representations with statistics concerning the intersection of corpora. Without exception, our word representations outperform every other word representation on the part-of-speech induction task. 1600 dimensional word representations of Mikolov et al. (2010) comes closest to our featureless representations and but does not manage to outperform them. It shows that our word representations are more suitable for inducing syntactic categories of words.

[1]However, we could not use word context representations of Huang et al. (2012) because their word types do not completely match the word types in our test corpus, preventing us from generating the word context representations.

[2]Intersections only contain the word types that are present in both the corpus used by the study and WSJ corpus. If the word types are only available in uppercase or lowercase, we conduct a case insensitive match for the word types.

Previous Work & Stats	Model	MTO	VM
Collobert et al. (2008)⋆ Word Types 37,639 Word Tokens 1,148,468 POS Tags 43	25	0.7332 (0.0076)	0.6594 (0.0040)
	50	0.7275 (0.0070)	0.6645 (0.0064)
	100	0.7180 (0.0090)	0.6652 (0.0058)
	200	0.6840 (0.0215)	0.6493 (0.0095)
	NN X	0.7650 (0.0098)	0.6832 (0.0052)
	NN XY	0.7540 (0.0096)	0.6702 (0.0063)
	NN+F X	**0.7996 (0.0076)**	**0.7259 (0.0047)**
	NN+F XY	**0.7972 (0.0054)**	0.7115 (0.0055)
Mnih et al. (2009)⋆ Word Types 37,943 Word Tokens 1,130,733 POS Tags 42	50	0.6851 (0.0183)	0.6259 (0.0093)
	100	0.6875 (0.0236)	0.6343 (0.0134)
	NN X	0.7676 (0.0070)	0.6811 (0.0045)
	NN XY	0.7592 (0.0091)	0.6663 (0.0047)
	NN+F X	**0.8024 (0.0102)**	**0.7205 (0.0072)**
	NN+F XY	**0.8023 (0.0101)**	0.7084 (0.0058)
Mikolov et al. (2010)† Word Types 31,273 Word Tokens 924,159 POS Tags 36	80	0.6475 (0.0146)	0.5823 (0.0108)
	640	0.5899 (0.0260)	0.5741 (0.0130)
	1600	0.7306 (0.0088)	0.6284 (0.0092)
	NN X	0.7386 (0.0077)	0.6249 (0.0036)
	NN XY	0.7334 (0.0091)	0.6172 (0.0032)
	NN+F X	0.7788 (0.0075)	**0.6744 (0.0054)**
	NN+F XY	**0.7829 (0.0108)**	0.6649 (0.0053)
Huang et al. (2012)† Word Types 34,618 Word Tokens 1,115,190 POS Tags 43	50	0.6948 (0.0073)	0.6479 (0.0060)
	NN X	0.7631 (0.0094)	0.6790 (0.0041)
	NN XY	0.7518 (0.0092)	0.6637 (0.0045)
	NN+F X	0.7970 (0.0071)	**0.7185 (0.0050)**
	NN+F XY	**0.8032 (0.0086)**	0.7089 (0.0046)

Table 4.3: Statistics of the subsets of WSJ section of PTB corpus that has the word types in the works reported are specified in the first column under the work's citation. Numbers in the second column specifies the number of dimensions of the referred word representations. NN X (and XY) and NN+F X (and XY) in the second column are part-of-speech inductions for nearest neighbor discretization and nearest neighbor discretization with features, respectively, for the subsets of WSJ. Representations for entries with $\star$ are generated by Turian et al. (2010). Entries with † contain only uppercase or lowercase words type representations. For that reason, we conduct a case insensitive match to determine the subsets.

Chapter 5

Conclusion

In this thesis, we presented new methods to form representations for word context and word tokens, in addition to the word types. While we generate these representations we utilized paradigmatic relations instead syntagmatic relations of words. We were also able to incorporate morphological and orthographic features to our generation process.

We applied these word representations with and without features to the part-of-speech induction problem. Our MTO mapping accuracy and VM scores proved to be the state-of-the-art results for both type based and token based part-of-speech induction.

We also compared our word representations to other word representations from the previous studies and show that indeed our representations outperform them in the task of part-of-speech induction.

Our work here shows that the word type and word token representations generated with nearest neighbors discretization and S-CODE framework are good descriptors for syntactic categories of word types and word tokens. We believe exploration of the semantic relations captured by the word representations would be a potentially rewarding future work, considering their success in the induction of syntactic categories.

Another future work would be the incorporation of our word representations to the supervised natural language processing algorithms as features. Studies like Turian et al. (2010) already demonstrated the value of inclusion

of word type representations to the supervised learning algorithms. We believe these kind of algorithms would also benefit from our word type, word context and word token representations.

As the final future work, we would like to determine the performance of our part-of-speech induction with some extrinsic method. Measuring the performance with MTO accuracy and VM scores result in simple performance figures we can compare, but both of them compute these figures by looking at the gold-standard tags. We think that trying to increase our score by imitating the gold-standard may not be the best approach in learning syntactic categories. There may be much finer grained grouping of the syntactic roles, however we can not evaluate them by comparing to the gold-standard without bias. For that reason, we believe that a task such as end-to-end machine translation, parsing or textual entailment may be more suitable to measure performance of part-of-speech induction.

Bibliography

D. Arthur and S. Vassilvitskii. k-means++: The Advantages of Careful Seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

Taylor Berg-Kirkpatrick, Alexandre Bouchard-Côté, John DeNero, and Dan Klein. Painless Unsupervised Learning with Features. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 582–590, Los Angeles, California, June 2010. Association for Computational Linguistics.

Chris Biemann. Unsupervised Part-of-Speech Tagging Employing Efficient Graph Clustering. In *Proceedings of the 21st International Conference on computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 7–12. Association for Computational Linguistics, 2006.

Phil Blunsom and Trevor Cohn. A Hierarchical Pitman-Yor Process HMM for Unsupervised Part of Speech Induction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 865–874, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. Class-based n-gram models of natural language. *Comput. Linguist.*, 18:467–479, December 1992. ISSN 0891-2017.

Roger Brown and Jean Berko. Word Association And The Acquisition Of Grammar. *Child Development*, 31(1):1–14, 1960.

D. Chandler. *Semiotics: The Basics*. The Basics Series. Routledge, 2007. ISBN 9780415363761.

Christos Christodoulopoulos, Sharon Goldwater, and Mark Steedman. Two Decades of Unsupervised POS Induction: How far have we come? In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 575–584, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

Christos Christodoulopoulos, Sharon Goldwater, and Mark Steedman. A Bayesian Mixture Model for PoS Induction Using Multiple Features. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 638–647, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics.

Alexander Clark. Combining Distributional and Morphological Information for Part of Speech Induction. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics - Volume 1*, EACL '03, pages 59–66, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. ISBN 1-333-56789-0.

R. Collobert and J. Weston. A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. In *International Conference on Machine Learning, ICML*, 2008.

R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research*, 12:2493–2537, 2011.

Mathias Creutz and Krista Lagus. Inducing the morphological lexicon of a natural language from unannotated text. In *Proceedings of AKRR'05, International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning*, pages 106–113, Espoo, Finland, June 2005.

Ido Dagan, Lillian Lee, and Fernando Pereira. Similarity-Based Methods for Word Sense Disambiguation. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, EACL '97, pages 56–63, Stroudsburg, PA, USA, 1997. Association for Computational Linguistics. doi: 10.3115/979617.979625.

Dipanjan Das and Slav Petrov. Unsupervised Part-of-Speech Tagging with Bilingual Graph-Based Projections. In *ACL*, pages 600–609, 2011.

Scott C. Deerwester, Susan T Dumais, and Richard A. Harshman. Indexing by Latent Semantic Analysis. 1990.

Paramveer Dhillon, Dean P Foster, and Lyle H Ungar. Multi-View Learning of Word Embeddings via CCA. In *Advances in Neural Information Processing Systems*, pages 199–207, 2011.

Paramveer S. Dhillon, Jordan Rodu, Dean P. Foster, and Lyle H. Ungar. Two Step CCA: A new spectral method for estimating vector models of words. In *Proceedings of the 29th International Conference on Machine learning*, ICML'12, 2012.

C. Elkan. Using the Triangle Inequality to Accelerate k-Means, 2003.

Amir Globerson, Gal Chechik, Fernando Pereira, and Naftali Tishby. Euclidean Embedding of Co-occurrence Data. *J. Mach. Learn. Res.*, 8:2265–2295, December 2007. ISSN 1532-4435.

Sharon Goldwater and Tom Griffiths. A Fully Bayesian Approach to Unsupervised Part-of-Speech Tagging. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 744–751, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

Joao V. Graça, Kuzman Ganchev, and Ben Taskar. Expectation Maximization and Posterior Constraints. 2007.

Joao V. Graça, Kuzman Ganchev, Ben Taskar, and Fernando Pereira. Posterior vs. Parameter Sparsity in Latent Variable Models. In *In Proceeding of NIPS*, pages 664–672, 2009.

David Graff, Roni Rosenfeld, and Doug Paul. CSR-III Text. Linguistic Data Consortium, Philadelphia, 1995.

Aria Haghighi and Dan Klein. Prototype-Driven Learning for Sequence Models. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 320–327. Association for Computational Linguistics, 2006.

Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. Improving Word Representations via Global Context and Multiple Word Prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, pages 873–882, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

Mark Johnson. Why doesn't EM find good HMM POS-taggers? In *EMNLP-CoNLL*, pages 296–305, 2007.

Pentti Kanerva, Jan Kristoferson, and Anders Holst. Random Indexing of Text Samples for Latent Semantic Analysis. In *In Proceedings of the 22nd Annual Conference of the Cognitive Science Society*, pages 103–6. Erlbaum, 2000.

Michael Lamar, Yariv Maron, and Elie Bienenstock. Latent-descriptor Clustering for Unsupervised POS Induction. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 799–809, Stroudsburg, PA, USA, 2010a. Association for Computational Linguistics.

Michael Lamar, Yariv Maron, Mark Johnson, and Elie Bienenstock. SVD and Clustering for Unsupervised POS Tagging. In *Proceedings of the ACL 2010*

Conference Short Papers, pages 215–219. Association for Computational Linguistics, 2010b.

Minh-Thang Luong, Richard Socher, and Christopher D. Manning. Better Word Representations with Recursive Neural Networks for Morphology. In *CoNLL*, Sofia, Bulgaria, 2013.

Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning Word Vectors for Sentiment Analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 142–150. Association for Computational Linguistics, 2011.

Mitchell P. Marcus, Beatrice Santorini, Mary Ann Marcinkiewicz, and Ann Taylor. Treebank-3. Linguistic Data Consortium, Philadelphia, 1999.

Yariv Maron, Michael Lamar, and Elie Bienenstock. Sphere Embedding: An Application to Part-of-Speech Induction. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1567–1575, 2010.

Tomas Mikolov, Martin Karafiát, Lukáš Burget, Jan Cernocky, and Sanjeev Khudanpur. Recurrent Neural Network Based Language Model. *Proceedings of Interspeech*, 2010.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic Regularities in Continuous Space Word Representations. *Proceedings of NAACL-HLT*, pages 746–751, 2013.

Andriy Mnih and Geoffrey Hinton. Three New Graphical Models for Statistical Language Modelling. In *Proceedings of the 24th International Conference on Machine learning*, pages 641–648. ACM, 2007.

Andriy Mnih and Geoffrey E Hinton. A Scalable Hierarchical Distributed Language Model. In *Advances in Neural Information Processing Systems*, pages 1081–1088, 2009.

Joseph Reisinger and Raymond J Mooney. Multi-Prototype Vector-Space Models of Word Meaning. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 109–117. Association for Computational Linguistics, 2010.

A. Rosenberg and J. Hirschberg. V-Measure: A conditional ntropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 410–420, 2007.

Magnus Sahlgren. An Introduction to Random Indexing. In *Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, TKE*, volume 5, 2005.

Hinrich Schütze. Distributional Part-of-Speech Tagging. In *Proceedings of the seventh conference on European chapter of the Association for Computational Linguistics*, pages 141–148. Morgan Kaufmann Publishers Inc., 1995.

Hinrich Schütze. Automatic Word Sense Discrimination. *Comput. Linguist.*, 24(1):97–123, March 1998. ISSN 0891-2017.

Richard Socher, John Bauer, Christopher D. Manning, and Andrew Y. Ng. Parsing With Compositional Vector Grammars. In *ACL*. 2013.

Andreas Stolcke. SRILM – An extensible language modeling toolkit. In *Proceedings International Conference on Spoken Language Processing*, pages 257–286, November 2002.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394. Association for Computational Linguistics, 2010.

Peter D Turney and Patrick Pantel. From Frequency to Meaning: Vector Space Models of Semantics. *Journal of artificial intelligence research*, 37 (1):141–188, 2010.

Mehmet Ali Yatbaz, Enis Sert, and Deniz Yuret. Learning Syntactic Categories Using Paradigmatic Representations of Word Context. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 940–951. Association for Computational Linguistics, 2012.

Deniz Yuret. FASTSUBS: An Efficient and Exact Procedure for Finding the Most Likely Lexical Substitutes Based on an N-Gram Language Model. *Signal Processing Letters, IEEE*, 19(11):725–728, Nov 2012. doi: 10.1109/LSP.2012.2215587.

Deniz Yuret and Mehmet Ali Yatbaz. The Noisy Channel Model for Unsupervised Word Sense Disambiguation. *Computational Linguistics*, 36(1): 111–127, 2010.