Unsupervised Part of Speech Induction Using Paradigmatic Representations of Word Context

Mehmet Ali Yatbaz *
Koç University

Enis Sert
Koç University

Deniz Yuret
Koç University

We investigate paradigmatic representations of word context in the domain of unsupervised part of speech induction. Paradigmatic representations of word context are based on potential substitutes of a word in contrast to syntagmatic representations based on its neighbors. We model the joint probability of words and their contexts (as represented by potential substitutes) using the S-CODE framework. S-CODE maps target words, their potential substitutes and other features to high dimensional Euclidean vectors. These vectors aggregate into clusters that largely match the traditional part-of-speech boundaries and give state-of-the-art results in unsupervised part-of-speech induction, including 80% many-to-one accuracy on the Penn Treebank and statistically significant improvements over best published results on 17 out of 19 corpora in 15 languages.

1. Introduction

Grammar rules apply not to individual words (e.g. dog, eat) but to part-of-speech categories (e.g. noun, verb). Thus learning part-of-speech categories (also known as lexical or syntactic categories) is one of the fundamental problems in language acquisition.

Linguists identify part-of-speech categories based on semantic, syntactic, and morphological properties of words. There is also evidence that children use prosodic and phonological features to bootstrap part-of-speech category acquisition (Ambridge and Lieven 2011). However there is as yet no satisfactory computational model that match human performance. Thus identifying the best set of features and best learning algorithms for part-of-speech induction is still an open problem.

Relationships between linguistic units can be classified into two types: syntagmatic (concerning positioning), and paradigmatic (concerning substitution). Syntagmatic relations determine which units can combine to create larger groups and paradigmatic relations determine which units can be substituted for one another. Figure 1 illustrates the paradigmatic vs syntagmatic axes for words in a simple sentence and their possible substitutes.

Part-of-speech categories represent groups of words that can be substituted for one another without altering the grammaticality of a sentence. In this paper we explore models of part-of-speech induction based on potential substitutes of words. We build *substitute word distributions*

∗ Artificial Intelligence Laboratory, Koç University, 34450 Sarıyer, İstanbul, Turkey. E-mail: myatbaz@ku.edu.tr, esert@ku.edu.tr, dyuret@ku.edu.tr

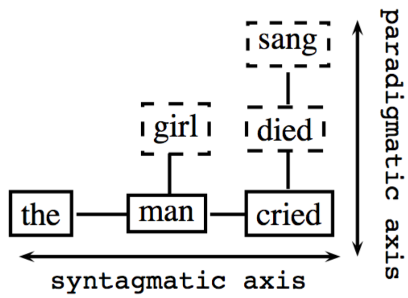

Figure 1: Syntagmatic vs. paradigmatic axes for words in a simple sentence (Chandler 2007).

for each position in the text which specify the probability of every vocabulary word in that position. Table 1 gives substitute distributions for an example sentence.

Table 1: The substitute word distributions (with probabilities in parentheses) for some of the positions in the example sentence *"Pierre Vinken, 61 years old, will join the board as a nonexecutive director Nov. 29."* based on an n-gram language model.

will:	*will* (0.9985), *would* (0.0007), *to* (0.0006), *also* (0.0001), . . .
join:	*join* (0.6528), *leave* (0.2140), *oversee* (0.0559), *head* (0.0262), *rejoin* (0.0074), . . .
the:	*its* (0.9011), *the* (0.0981), *a* (0.0006), . . .
board:	*board* (0.4288), *company* (0.2584), *firm* (0.2024), *bank* (0.0731), *strike* (0.0030), . . .

Note that the substitute word distribution for a position (e.g. the second position in Fig. 1) is a function of the context only (i.e. *"the ___ cried"*), and does not depend on the word that actually appears there (i.e. "man"). Thus substitute distributions represent *individual word contexts*, not word types. We refer to representations based on substitute distributions as *paradigmatic representations of word context*.

We expect words used in similar contexts (with similar substitute distributions) to share the same part-of-speech. Thus part-of-speech induction depends on which contexts are considered similar, and context similarity in turn is a function of the features used to represent word context. Paradigmatic representations, using features of the substitute distribution, uncover latent similarities between contexts that on the surface seem to have little in common. This makes paradigmatic representations more robust to data sparsity, compared to syntagmatic representations which use neighboring words as features. Our empirical results demonstrate that paradigmatic representations significantly outperform syntagmatic ones when compared using similar part-of-speech induction algorithms on identical datasets. Section 1.1 presents alternative representations of word context and discusses paradigmatic representations in more detail.

1.1 Representing Word Context

In this section we demonstrate the different contextual representations in the part-of-speech induction (aka. syntactic word categorization) literature and introduce the substitute words as an alternative to the current context representations. In the rest of the paper the words in the vocabulary are referred as *types* and the instances of types are referred as *tokens*.

The contextual representations can be categorized into three groups based on the way they incorporate the local context information of the target type or token: (1) syntagmatic representation, (2) Hidden Markov Models (HMM) and (3) paradigmatic representation. These representations can be further subdivided into two subgroups based on whether they group the types or the tokens.

Syntagmatic Representation. In syntagmatic representation the context is defined with the neighboring words, typically co-occurrences with a single word on the left or a single word on the right word called a "frame" (e.g., **the** *dog* **is***;* **the** *cat* **is**) (Schütze and Pedersen 1993; Redington, Crater, and Finch 1998; Mintz 2003; St Clair, Monaghan, and Christiansen 2010; Lamar et al. 2010; Maron, Lamar, and Bienenstock 2010). Turney and Pantel (2010) give a broad overview of syntagmatic approaches and their applications within the Vector Space Modeling framework. Depending on the way they incorporate co-occurences, these models can perform hard (type based) or soft (token based) clustering.

Schütze (1993) represented the context of a word type by concatenating its left and right co-occurrence vectors. These vectors were calculated for each type by using the left and the right neighbors of the type instances therefore they characterize the distribution of the left and right neighboring tokens of the type. One constraint of this representation is that it represents types rather than tokens thus it is not possible to group the instances of any type into the separate categories.

Mintz (2003) showed on a subset of child directed speech corpus (CHILDES) (MacWhinney 2000) that non-adjacent high frequent bigram frames are useful for the language learners on the syntactic categorization of the tokens. For example, the tokens that are observed at "_" in the frame "**the _ is**" are assigned to the same category. Using the top-45 frequent frames Mintz achieved an average of 98% unsupervised accuracy[1]. The main limitation of the top-45 frequent frames is that they could only analyze the 6% of the tokens on average due to the sparsity. Another drawback is that the tokens with only one common neighbors could not exchange information.

St Clair et al. (2010) extended the work of Mintz (2003) and introduced the flexible bigram frames which represent the context by using the left and the right bigrams separately. As a result tokens with a common left or right bigram can exchange information and might be grouped together. For instance, two tokens that are observed at "'_" in "**the _ is**" and "**a _ is**" can be categorized together due to the shared right bigram "**is**". Using a feed forward connectionist model they showed that the flexible frames are statistically better than the frequent frames in terms of the supervised accuracy[2]. They also showed that representing token contexts only with the left or the right bigram is statistically better than the frequent frames but worse than the flexible frames in terms of supervised accuracy. Both Mintz (2003) and St Clair (2010) did not report any results with contexts larger than bigram since as the context is enriched, the re-occurrence frequency of a frame becomes lower which causes the data sparsity (Manning and Schütze 1999).

HMM. Prototypical HMM uses a bigram structure where tokens are generated by latent categories and learns the latent category sequence that generates the given word sequence instead of clustering tokens directly (Brown et al. 1992; Blunsom and Cohn 2011; Goldwater and Griffiths 2007; Johnson 2007; Ganchev et al. 2010; Berg-Kirkpatrick and Klein 2010; Lee, Haghighi, and Barzilay 2010). The POS induction literature focused on the first and second order HMMs since the higher order HMMs have additional complicating factors[3] and require more complex training

1 Unsupervised accuracy was defined as the number of hits (when two intervening tokens that observed in the frame are from the same category) divided by number of false alarms (when two intervening tokens that observed in the frame are from different categories).
2 In order to perform meaningful comparisons they used all of the frequent frames instead of the top-45 ones.
3 The number of parameters in a prototypical HMM quadratically increases as the HMM order increases.

procedures (Johnson 2007). Depending on the design and the training procedure HMM models can group types or tokens which are detailed in Section 5.

Paradigmatic Representation. In the paradigmatic representation context is defined as the distribution of the substitute words in that context. Schütze (1995) incorporates paradigmatic information by concatenating the left co-occurrence and the right co-occurrence vectors of the right and the left tokens, respectively and grouped the tokens that have similar vectors. The vectors from the neighbors include potential substitutes. Yatbaz et al. (2012) calculate the most likely substitute words of a word in a given context and clusters the types that have similar substitutes.

Our paradigmatic representation is related to the second order co-occurrences used in (Schütze 1995). Our method improves on his foundation from three aspects: (1) it can cluster both the types and tokens (2) it uses a 4-gram language model rather than bigram statistics, (3) it includes the whole 78,498 word vocabulary rather than the most frequent 250 words. More importantly, rather than simply concatenating the vectors that represent the target word with vectors that represent the context we use a co-occurrence modeling algorithm.

Similarly, Schütze and Pedersen (1993) define the words that frequently co-occur together as the *syntagmatic associates* and words that have similar left and right neighbors as the *paradigmatic parallels*. We find that representing the paradigmatic axis more directly using substitute vectors rather than frequent neighbors improves part of speech induction.

Sahlgren (2006) gives a detailed analysis of paradigmatic and syntagmatic relations in the context of word-space models used to represent the word meanings. Sahlgren's paradigmatic model represents word types using co-occurrence counts of their frequent neighbors, in contrast to his syntagmatic model that represents word types using counts of contexts (documents, sentences) they occur in. Our substitute vectors do not represent word types at all, but *contexts of word tokens* using probabilities of likely substitutes. Sahlgren finds that in word-spaces built by frequent neighbor vectors, more nearest neighbors share the same part of speech compared to word-spaces built by context vectors.

The two examples below illustrate the advantages of paradigmatic representation in uncovering similarities where no overt similarity that can be captured by a syntagmatic representation exists. The word "director" from the first sentence and the word "chief" from the second one have no common neighbors in their 4-gram neighborhood. The paradigmatic representation captures the similarity of these words by suggesting the same top substitutes for both (the numbers in parentheses give substitute probabilities):

(1) *"Pierre Vinken, 61 years old, will join the board as a nonexecutive* **director** *Nov. 29."*
director: chairman (.8242), director (.0127), directors (.0127) . . .

(2) *". . . Joseph Corr was succeeded by Frank Lorenzo ,* **chief** *of parent Texas Air ."*
chief: chairman (.09945), president (.0031), directors (.0012) . . .

The high probability substitutes reflect both semantic and syntactic properties of the context. Top substitutes for "director" and "chief" are not only nouns, but specifically nouns compatible with the semantic context. Top substitutes for the word "the" in the first example consist of words that can act as determiners: its (.9011), the (.0981), a (.0006), ...

2. Algorithm

In this section we describe the components of our algorithm and their relationship with each other. The algorithm predicts the syntactic category of a word in a given context based on its random substitutes. In other words first we construct the co-occurrence representation of words and their substitutes with the help of a language model and then map each value in the co-occurrence data to a corresponding embedding on a n-dimensional sphere using the S-CODE algorithm (Maron, Lamar, and Bienenstock 2010). Finally, we apply k-means clustering to categorize the word embeddings by which we induced the word categories. In the next subsection we detail the representation of word contexts as co-occurrence data, in Subsection 2.2 we explain the embedding calculation and finally in Subsection 2.3 we describe the different ways of embedding clustering.

2.1 Context Representation

Word contexts are represented by random substitutes that are sampled from the corresponding substitute word distributions. Random substitutes are sampled with replacement from the substitute distributions that are calculated based on an n-gram language model. The sample space of the substitute word distributions is the vocabulary of the language model.[4] It is possible (and beneficial) to sample more than one substitutes and generate more pairs from the same substitute distribution as in Table 2. The calculation of substitute distributions and random substitute sampling are detailed in Appendix A. To capture the relation between each word and its context we construct a co-occurrence representation by pairing the words with their random substitutes. Table 2 shows random substitutes of each word and their co-occurrence representation on an example sentence. A target word might appear both as a word and a random substitute therefore to clarify this ambiguity we concatenate "W" and "C" to words and contexts (i.e., random substitutes), respectively, in the co-occurrence data.

The next section explains the S-CODE algorithm which takes the co-occurrence data as its input and calculates the embeddings of the words and their substitutes on an n-dimensional sphere. In the rest of the paper we use the term "substitutes" and "random substitutes" interchangeably.

2.2 Co-occurrence Embedding

The S-CODE algorithm maps each word and substitute value in the co-occurrence data to an embedding on an n-dimensional sphere as detailed in Appendix B. The basic idea of the mapping is that words and substitutes that are frequently observed as pairs in the co-occurrence data will have close embeddings while unobserved pairs will have embeddings that are apart from each other.

The co-occurrence data in Figure 2 consists of pairs such as (*W:director*, *C:chairman*) and (*W:chief*, *C:chairman*) therefore S-CODE forces the embeddings of *W:director* and *W:chief* to be close to the embedding of *C:chairman*. Similar to the former case the embeddings of *W:Pierre* and *W:Frank* will be close to the embedding *C:Mr.* because of the frequently observed pairs. As a result the final embeddings of *W:director* and *W:chief* will be close to each other due to the common substitute *C:chairman* and will be apart from *W:Pierre* and *W:Frank* due to the lack of common substitute as shown on Figure 2 (similarly the embeddings of *W:Pierre* and *W:Frank* will be close to each other due to *C:Mr.*).

S-CODE constructs embeddings on n-dimensional sphere for each word type and substitute. Each pair in the co-occurrence data can be represented in three different ways by using the output of S-CODE: (1) word embedding ($\mathbf{W}$) which represents the word type information, (2) substitute

4 Sampled substitutes might include the unknown word tag "<unk>" since it is in the language model vocabulary. For example substitutes of proper nouns usually include "<unk>" as a substitute.

Table 2: The table on the left shows three possible substitutes, separated with “/”, sampled for each position in the example sentence *“Pierre Vinken, 61 years old, will join the board as a nonexecutive director Nov. 29 .”* based on a 4-gram language model. The right one represents the input sentence as co-occurrences of words and their substitutes. Thus words on the left column presents the target word while words on the right column represents the context of the corresponding target word.

Word	**Random Substitutes**
Pierre	*Mr. / Pierre / Mr.*
Vinken	*<unk> / Beregovoy / Cardin*
,	*, / , / ,*
61	*48 / 52 / 41*
years	*years / years / years*
old	*old / old / old*
,	*, / , / ,*
will	*will / will / will*
join	*head / join / leave*
the	*its / its / the*
board	*board / company / firm*
as	*as / as / as*
a	*a / a / a*
nonexecutive	*nonexecutive / non-executive / nonexecutive*
director	*chairman / chairman / director*
Nov.	*April / May / of*
29	*16 / 29 / 9*
.	*. / . / .*

Word	**Context**
W:Pierre	*C:Mr.*
W:Pierre	*C:Pierre*
W:Pierre	*C:Mr.*
W:Vinken	*C:<unk>*
W:Vinken	*C:Beregovoy*
W:Vinken	*C:Cardin*
. . .	
W:join	*C:head*
W:join	*C:join*
W:join	*C:leave*
W:the	*C:its*
W:the	*C:its*
W:the	*C:the*
. . .	
W:director	*C:chairman*
W:director	*C:chairman*
W:director	*C:director*
. . .	

embedding ($\mathbf{C}$) which represents the context information, and (3) concatenation of word and substitute embeddings ($\mathbf{W} \oplus \mathbf{C}$). In the next section we apply k-means clustering to these three representation and analyze the characteristic of final clusters.

2.3 Embedding Clustering

Each target word in the original input sentence is represented with word–substitute pairs and each unique word and substitute value are represented with embeddings on an n dimensional sphere. Therefore clustering the embeddings means clustering the target words in the original input. We run instance weighted k-means algorithm to cluster the final embeddings constructed by S-CODE. The target words can be represented in three ways as suggested in the previous section and the clustering output of each representation will have different characteristics. We use instance weighted k-means to cluster each representation.

Word embeddings ($\mathbf{W}$*).* All tokens of a word type are represented with the same embedding. For instance, although we sample three substitutes per target word in Table 2, each word type has only one embedding in $\mathbf{W}$. Thus clustering target words based on this representation employ the one-tag-per-word assumption from the beginning. The one-tag-per-word assumption is suitable for the part of speech induction problem given that 93.69% of the word occurrences in the human labeled PTB data are tagged with their most frequent part of speech (Toutanova et al. 2003). However the clustering performance on the ambiguous words (words that have more than one

Word	Context
...	...
W:director	C:chairman
W:chief	C:chairman
...	...
W:Pierre	C:Mr.
W:Frank	C:Mr.
...	...

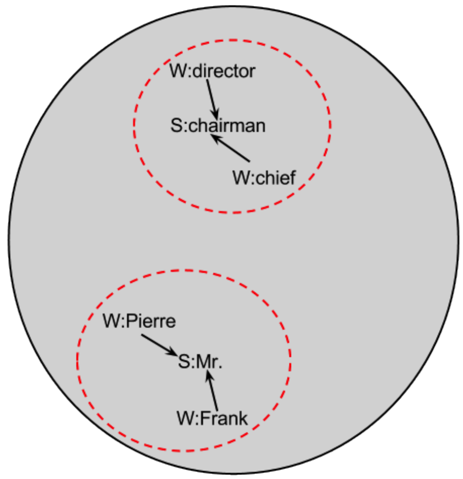

Figure 2: The figure on the right is the final embeddings of the input co-occurrence data given on the left table after S-CODE converges. Dashed circles represent the possible groupings of the embeddings on the sphere.

tag) degrades due to the one-tag-per-word assumption. For example, the word *offer* is tagged as NN(399), VB(105) and VBP(34) in its 538 WSJ instances[5]. In this scheme all instances of *offer* tagged with one tag and MTO upper bound of *offer* will be .7616 due to the most frequent tag NN(399). Our model in Section 3.2 clusters word embeddings and achieves the MTO upper bound of *offer* however this is not the case for all ambiguous words.

Context embeddings ($\mathbf{C}$*).* Each target word token is represented with embeddings of its substitutes that depends on the token context. Models based on this representation cluster word tokens and do not force the one-tag-per-word assumption. For example, the word "Pierre" in Table 2 will be represented with the embeddings of *S:Mr.*, *S:Pierre* and *S:Mr.* while another occurrence of the word "Pierre" in a different context might be represented with different substitutes. It is possible to represent the context with several substitutes, in such a case k-means might group substitute embeddings into different clusters which leads to an ambiguity on the final cluster of the target word. To solve this issue the target word is assigned to the cluster in which the majority of its substitute embeddings are present[6]. The model in Section 3.3 clusters $\mathbf{C}$ and achieves .8215 MTO on the word *offer* which is higher than the one-tag-per-word MTO upper bound.

Concatenation of word and context embeddings ($\mathbf{W} \oplus \mathbf{C}$*).* This representation concatenates the word type and its substitute embeddings such that each word instance is represented with r vectors where r is the number of random substitutes per target word. For instance, the target word "Pierre" in Table 2 will be represented with three vectors that will be the concatenation of the embedding of *W:Pierre* with the embeddings of *C:Mr.*, *C:Pierre* and *C:Mr.*, separately. Models that are based on this representation do not employ the one-tag-per-word assumption and cluster word tokens. Clusters that are constructed according this representation tend to be

5 NN, VB and VBP are three part-of-speech tags from the Penn Treebank corpus and they correspond to singular noun, verb in base form and non-3^{rd}person singular verb in present tense, respectively.
6 Ties are broken randomly.

sparser than the previous one due to the concatenation of $\mathbf{W}$. Similar to the former case we run k-means on these concatenated vectors and use majority voting to decide the target word cluster. The model in Section 3.4 clusters concatenation of embeddings and outperforms the previous models by scoring .8866 MTO accuracy on the word *offer*.

The first representation applies the one-tag-per-word assumption from the beginning and clusters word types instead of tokens. On the other hand the second one relaxes the one-tag-per-word assumption and clusters tokens by using context embeddings. The final one clusters tokens by representing each token with the concatenation of the word and context embeddings therefore it is more biased to sparse clusters than the previous one.

Section 3 compares the performance of these three representations on the POS induction problem.

3. Experiments

Section 3.1 details the test corpus and experiment parameters used in the rest of this work. Section 3.2 is presents the sensitivity analysis of model parameters. Section 3.2, 3.2 and 3.4 aim to compare performance of models on $\mathbf{W}$, $\mathbf{C}$ and $\mathbf{W} \oplus \mathbf{C}$, respectively. Section 3.5 explores morphological and orthographic features as additional sources of information for POS induction of word types and its results improve the state-of-the-art in the field of POS induction. Section 3.6 compares our paradigmatic representation of the word context to previously described syntagmatic representations of the word context for word types. Finally, Section 3.7 extends the language and corpus coverage by applying the best performing model to 19 corpora in 15 languages.

3.1 Experimental Settings

To make a meaningful comparison to the previous works the Wall Street Journal Section of the Penn Treebank (PTB) (Marcus et al. 1999) was used as the test corpus (1,173,766 tokens, 49,206 types) to be induced. The treebank uses 45 part-of-speech tags which is the set we used as the gold standard for comparison in our experiments.

To compute substitutes in a given context we trained a language model using approximately 126 million tokens of Wall Street Journal data (1987-1994) extracted from CSR-III Text (Graff, Rosenfeld, and Paul 1995) (excluding sections of the PTB). We used SRILM (Stolcke 2002) to build a 4-gram language model with Kneser-Ney discounting. Words that were observed less than 20 times in the language model training data were replaced by UNK tags, which gave us a vocabulary size of 78,498. The perplexity of the 4-gram language model on the test corpus is 96. For computational efficiency only the top 100 substitutes and their unnormalized probabilities were computed for each positions in the PTB using the FASTSUBS algorithm (Yuret 2012)[7]. The probability vectors for each position were normalized to add up to 1.0 giving us the final substitute distributions used in the rest of this study.

The experiments were run using the following default settings (unless otherwise stated): (i) each word was kept with its original capitalization, (ii) the learning rate parameters were set to $\varphi_0 = 50, \eta_0 = 0.2$ for faster convergence in log likelihood, (iii) the number of S-CODE iterations were set to 50 million, (iv) the S-CODE dimensions and Z were set to 25 and 0.166, respectively, (v) a modified k-means algorithm with smart initialization was used (Arthur and Vassilvitskii 2007), and (vi) the number of k-means restarts were set to 128 to improve clustering and reduce variance.

7 The substitutes with unnormalized log probabilities can be downloaded from `http://goo.gl/jzKH0`.

Table 3: Summary of results in terms of the MTO and VM scores. Standard errors are given in parentheses when available. Starred entries have been reported in the review paper (Christodoulopoulos, Goldwater, and Steedman 2010). Distributional models use only the identity of the target word and its context. The models on the right incorporate orthographic and morphological features.

Distributional Models	MTO	VM	Models with Additional Features	MTO	VM
Lamar et al. (2010)	.708	-	Clark (2003)*	.712	.655
Brown et al. (1992)*	.678	.630	Christodoulopoulos et al. (2011)	.728	.661
Goldwater et al. (2007)*	.632	.562	Berg-Kirkpatrick et al. (2010)	.755	-
Ganchev et al. (2010)*	.625	.548	Christodoulopoulos et al. (2010)	.761	.688
Maron et al. (2010)	.688 (.0016)	-	Blunsom and Cohn (2011)	.775	.697
Substitutes(Sparse-tokens) (Sec. 3.4)	.7030 (.0070)	.6006 (.0071)	Substitutes and Features (Sec. 3.5)	.8002 (.0070)	.7163 (.0040)
Bigrams (Sec. 3.6)	.7319 (.0088)	.6554 (.0039)			
Substitutes(Types) (Sec. 3.2)	.7667 (.0056)	.6819 (.0029)			

Each experiment was repeated 10 times with different random seeds and the results are reported with standard errors in parentheses or error bars in graphs. Table 3 summarizes all the results reported in this section and the ones we cite from the literature.

3.2 Clustering Word Embeddings (W)

The S-CODE uses stochastic gradient ascent (see Appendix B) to find the ϕ_w, ψ_c embeddings for word and random-substitute in these pairs on a single 25-dimensional sphere. The algorithm cycles through the data until we get approximately 50 million updates. The resulting ϕ_w vectors are clustered using an instance weighted k-means algorithm. Cluster-id for each ϕ_w is assigned as the predicted cluster-id for word type W. Using the default settings and sampling 64 substitutes for each token the many-to-one accuracy is .7667 (.0056)and the V-measure is .6819 (.0029).

To analyze the sensitivity of this result to our specific parameter settings we ran a number of experiments where each parameter was varied over a range of values.

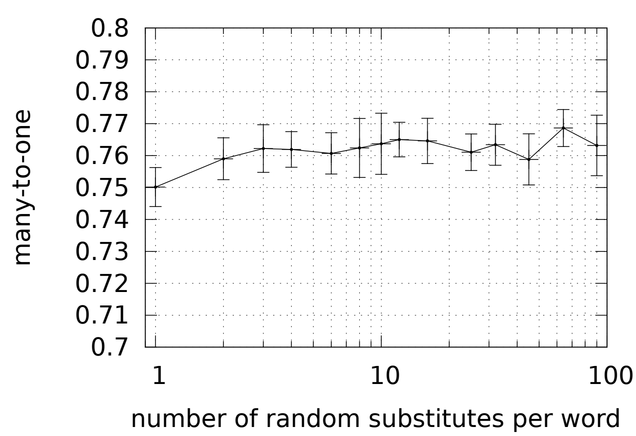

Figure 3: MTO is not sensitive to the number of random substitutes sampled per word token.

Figure 3 illustrates that the random-substitute result is fairly robust as long as the training algorithm can observe more than a few random substitutes per word.

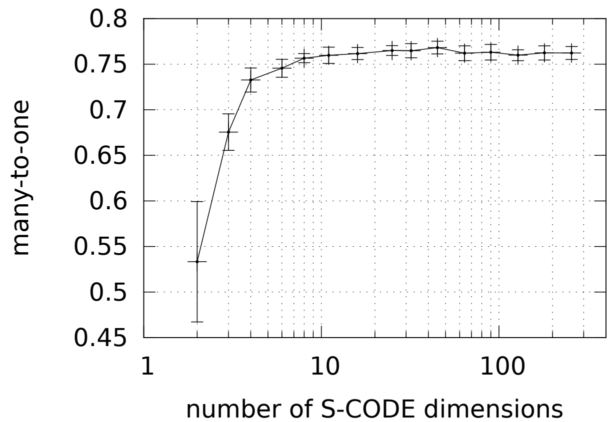

Figure 4: MTO falls sharply for less than 10 S-CODE dimensions, but more than 25 do not help.

Figure 4 shows that at least 10 embedding dimensions are necessary to get within 1% of the best result, but there is no significant gain from using more than 25 dimensions.

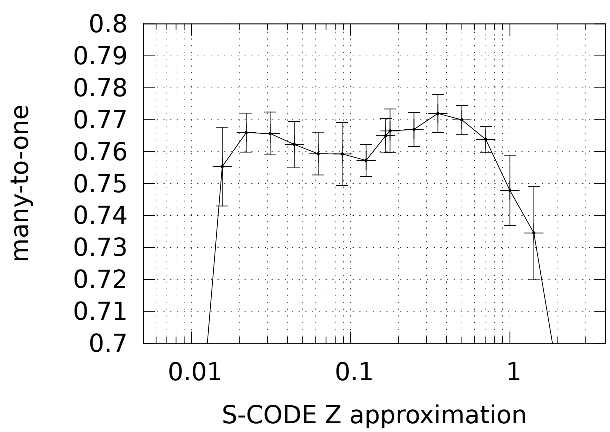

Figure 5: MTO is fairly stable as long as the $\tilde{Z}$ constant is within an order of magnitude of the real Z value.

Figure 5 shows that the constant $\tilde{Z}$ approximation can be varied within two orders of magnitude without a significant performance drop in the many-to-one score. For uniformly distributed points on a 25 dimensional sphere, the expected $Z \approx 0.146$. In the experiments where we tested we found the real Z always to be in the 0.140-0.170 range. When the constant $\tilde{Z}$ estimate is too small the attraction in Eq. B.3 dominates the repulsion in Eq. B.4 and all points tend to converge to the same location. When $\tilde{Z}$ is too high, it prevents meaningful clusters from coalescing.

We find our model with random substitute to be fairly robust to different parameter settings and the resulting many-to-one score significantly better than the state-of-the-art distributional models.

3.3 Clustering Context Embeddings (C)

In the previous section we group word types rather than word tokens by clustering the word embeddings. In this section we remove the one-tag-per-word assumption and group word tokens according to embeddings of their substitutes. We generate 64 substitutes for each word token and input them to S-CODE as word(W) – substitute ($\mathbf{C}$) pairs. The resulting embeddings of the context (i.e., substitutes) are clustered using the instance weighted k-means algorithm with 128 restarts. The process yields 64 cluster-ids (for every pair generated from word token's context) for each word token's context. The cluster-ids tokens are predicted by the majority cluster-id of the corresponding pairs. Ties for the majority are broken randomly. The many-to-one accuracy is .6366 (.0023) and the V-measure is .4865 (.0051) .

In order to demonstrate the merit in the token based POS induction, we first define the gold-tag perplexity for the word types as following:

$$GP(w) = 2^{H(p_w)} = 2^{-\sum_t p_w(t) log_2 p_w(t)} \tag{1}$$

where p_w is the gold POS tag distribution of the word type w and $H(p_w)$ is the entropy of the p_w distribution. Gold-tag perplexity (GP) is used to determine the POS ambiguity of the word types, relating how often a word type is associated with different POS tags in the test corpus. A GP of 1 for a word type w indicates w is associated with same POS tag throughout the test corpus, meaning the word type w's POS is unambiguous. As the GP increases ambiguity of the word types increases and poses a handicap for induction models that limits tag variety for the word types. To display the limitations, we split the test corpus in two subsets: word types with GP less than 1.75 and word types with GP equal or greater than 1.75. We performed MTO evaluation on our induction output and obtained the induced-tag – gold-tag mappings. Using the mappings obtained over the test corpus, we evaluated the accuracy in the subsets.

Table 4: The MTO accuracy of $\mathbf{W}$, $\mathbf{C}$ and $\mathbf{W} \oplus \mathbf{C}$ based models on two subsets that consist of words with GP smaller and larger than 1.75, respectively. The percentage of each subset in the test data is reported in the title bar. The average GP of each clustering over the whole corpus is reported in the last column. Each score is an average of 10 random starts of our algorithm and the standard error of each one is reported in parenthesis while statistically the best MTO score of each column is reported in bold.

Model	$GP < 1.75$ 89%	$GP \geq 1.75$ 11%	$GP \geq 1$ 100%	Average GP
Clustering $\mathbf{W}$ embeddings (Type based)	**.8054 (.0065)**	.4383 (.0104)	**.7667 (.0056)**	1.0 (.0)
Clustering $\mathbf{W} \oplus \mathbf{C}$ embeddings (Sparse-token based)	.7322 (.0079)	**.4671 (.0174)**	.7030 (.0070)	1.3406 (.0057)
Clustering $\mathbf{C}$ embeddings (Token based)	.6620 (.0051)	.4309 (.0093)	.6366 (.0023)	1.5318 (.0076)

The performance of our algorithm on $\mathbf{C}$ embeddings is summarized in Table 4. Due to the one-tag-per-word nature of POS induction, the type based model outperforms the token based one on the unambiguous words. The token based model achieves statistically comparable results with the type based model on the ambiguous words. Type based model can not handle words with ambiguity while the token based model can. In order to take advantage of both models we apply our algorithm on concatenation of $\mathbf{W}$ and $\mathbf{C}$ embeddings in the next section.

3.4 Clustering Concatenation of Word and Context Embeddings ($\mathbf{W} \oplus \mathbf{C}$)

Two models presented in earlier sections perform POS induction either by assuming (Section 3.2) or discarding (Section 3.3) the one-tag-per-word assumption. In this section we define a sparse-token based model which clusters the concatenation of $\mathbf{W}$ and $\mathbf{C}$ embeddings. This model not only tends to put instances of a word type into the same cluster but also performs token based clustering by incorporating the word type and context information together.

Similar to the previous models, we generate $\mathbf{W}$ – $\mathbf{C}$ pairs as the input to S-CODE. For each observed $\mathbf{W}$ – $\mathbf{C}$ pair in the S-CODE input, corresponding 25-dimensional ϕ_w and ψ_c embeddings are concatenated to create a 50-dimensional representation. We used the same experimental setting of the previous section and predict the token clusters according to the majority cluster-id of the corresponding pairs. The many-to-one accuracy of this model is .7030 (.0070) and the V-measure is .6006 (.0071) .

Table 4 presents the performance of the $\mathbf{W} \oplus \mathbf{C}$ based model over the subsets and it achieves statistically better MTO than both of the $\mathbf{W}$ and $\mathbf{C}$ based models on ambiguous words. Due to the bias towards to the sparse clustering, sparse-token based model statistically improves the MTO accuracy on unambiguous words compared to the $\mathbf{C}$ based model but it still can not achieve the performance of the $\mathbf{W}$ based model. The $\mathbf{W} \oplus \mathbf{C}$ based model constructs token based clusters that tend to assign instances of a word type into the same cluster which leads to a smaller average GP than the $\mathbf{C}$ based model as shown in Table 4.

3.5 Morphological and Orthographic Features

Clark (2003) demonstrates that using morphological and orthographic features significantly improves part of speech induction with an HMM based model. Section 5 describes a number of other approaches that show similar improvements. We integrate additional features together with substitutes by using the model described in Appendix C.

The orthographic features we used are similar to the ones in (Berg-Kirkpatrick et al. 2010) with small modifications:

- Initial-Capital: this feature is generated for capitalized words with the exception of sentence initial words.
- Number: this feature is generated when the token starts with a digit.
- Contains-Hyphen: this feature is generated for lowercase words with an internal hyphen.
- Initial-Apostrophe: this feature is generated for tokens that start with an apostrophe.

We generated morphological features using the unsupervised algorithm Morfessor (Creutz and Lagus 2005). Morfessor was trained on the WSJ section of the Penn Treebank using default settings, and a perplexity threshold of 1. In our model, a word type consists of two parts: a stem and a suffix part. The suffix part is used as the morphological feature thus each word type has only one morphological feature[8]. The program induced 5575 suffix types that are present in a total of 19223 word types. Table 5 presents the co-occurrence tuples of the example sentence after incorporating the orthographic and morphological features.

8 We extracted the stem part by concatenating the splits until including the first "STM" labeled split and the suffix part by concatenating rest of the splits.

Table 5: The words of input sentence *"Pierre Vinken, 61 years old, will join the board as a nonexecutive director Nov. 29 ."* is represented with their substitutes and features. Words on the left column presents the target word, words on the second column represents the context and tokens on the rest of the columns are the features of the correponding target word. Features without values are unobserved therefore set to null.

Word	**Context**	**Morphology**	**Initial Capital**	**Number**	**Contains Hypen**	**Initial Apostrophe**
W:Pierre	*C:Mr.*		*F:IC*			
W:Vinken	*C:<unk>*		*F:IC*			
W:,	*C:,*					
W:61	*C:48*			*F:N*		
W:years	*C:years*	*F:s*				
W:old	*C:old*					
W:join	*C:head*					
W:the	*C:its*					
W:board	*C:company*					
W:as	*C:as*					
W:a	*C:a*					
W:nonexecutive	*C:non-executive*					
W:director	*C:chairman*	*F:or*				
W:Nov.	*C:May*		*F:IC*			
W:29	*C:9*			*F:N*		
W.	*C:.*					

Using the training settings of the previous section, the addition of morphological and orthographic features increased the many-to-one score of the random-substitute model to .8002 (.0070) and V-measure to .7163 (.0040). Both these results improve the state-of-the-art in part of speech induction significantly as seen in Table 3.

3.6 Paradigmatic vs Syntagmatic Representations of Word Context

To get a direct comparison of the paradigmatic and syntagmatic context representations we feed 4 different co-occurrences defined in Section 1.1 into the S-CODE algorithm. The first model accepts word (X) - right bigram (Y) pairs as the input, the second model accepts word (X) - left bigram(Y) pairs as the input, the third model accepts word (X) - concatenation of the left and right bigrams (Y) pairs (Mintz 2003) as the input and the final model accepts words (X) - left bigram(Y_1) and right bigram (Y_2) tuples (St Clair, Monaghan, and Christiansen 2010) as the input to the S-CODE. At the end we cluster the word types (X) with k-means algorithm and report the results on Table 6. To replicate the work of Maron et al. (2010) we feed word (X) - right bigram (Y) pairs as the input. At the end each word w in the vocabulary ends up with two points on the sphere, a ϕ_w point representing the behavior of w as the left word of a bigram and a ψ_w point representing it as the right word. The two vectors for w are concatenated to create a 50-dimensional representation at the end. These 50-dimensional vectors are clustered using the k-means algorithm. Maron et al. (2010) report many-to-one scores of .6880 (.0016) for 45 clusters and .7150 (.0060) for 50 clusters (on the PTB). Using our default settings the bigram model achieves .7319 (.0088) MTO and .6554 (.0039) VM accuracies. Table 6 summarizes all the results and shows that the paradigmatic representation accuracies are significantly higher than the syntagmatic representation MTO and VM accuracies.

Table 6: Summary of results in terms of the MTO and VM scores of the S-CODE algorithm when the paradigmatic or syntagmatic representations are feed as an input. Standard errors are given in parentheses when available. Results of the statistically best performing system are written in bold. We do not report the original results of Maron et al. (2010) since our replication achieves higher accuracies.

Input	MTO	VM
X (word) - Y (right bigram)	.6625 (.0115)	.5809 (.0066)
X (word) - Y (left bigram)	.6604 (.0054)	.5983 (.0028)
X (word) - Y (left and right bigram concatenation)	.7268 (.0091)	.6416 (.0052)
X (word) - Y_1, Y_2 (left and right bigrams)	.7173 (.0061)	.6381 (.0032)
Maron et al. (2010)(replication)	.7319 (.0088)	.6554 (.0039)
X (word) - Y (random substitutes)	**.7667 (.0056)**	**.6819 (.0029)**

3.7 Multilingual Experiments

We performed experiments with a range of languages and three different feature configurations to establish both the robustness of our model across languages and to observe the effects of different features. Following Christodoulopoulos et al. (2011), in addition to the PTB we extend our experiments to 8 languages from MULTEXT-East (Bulgarian, Czech, English, Estonian, Hungarian, Romanian, Slovene and Serbian) (Erjavec 2004) and 10 languages from the CoNLL-X shared task (Bulgarian, Czech, Danish, Dutch, German, Portuguese, Slovene, Spanish, Swedish and Turkish) (Buchholz and Marsi 2006). For all experiments, we use the best performing model of Section 3.2 (i.e. clustering the word embeddings) with default settings. To perform meaningful comparisons with the previous work we train and evaluate our models on the training section of MULTEXT-East[9] and CONLL-X languages (Lee, Haghighi, and Barzilay 2010).

We train a 4-gram language model with the corresponding training corpora of each language as described in Section 3.1. To sample substitutes we calculate the probabilities of the top 100 substitutes for each position by using the corresponding language model. Morphological features of each language are extracted by the method described in Section 3.5. The details of the language model training and feature extraction are detailed in Appendix D.

For each language we report results of three models: (1) word embeddings (W), (2) word embeddings with orthographic features ($W + O$) and (3) word embeddings with both orthographic and morphological features ($W + O + M$). Similar to the settings used in Section 3.2, we use the 25 dimensional sphere with 64 substitutes for all languages. For each language the number of induced clusters is set to the number of tags in the gold-set as presented in Table 7.

As a baseline model we chose the syntagmatic bigram version of S-CODE described in Section 3.6 which is a very strong baseline compared to the ones used in (Christodoulopoulos, Goldwater, and Steedman 2011). Table 7 summarizes the MTO and VM scores of our models together with the syntagmatic bigram baseline and the best published accuracies on each language corpus.

W significantly outperforms the syntagmatic bigram baseline in both MTO and VM scores on 14 languages. $W + O + M$ has the state-of-the-art MTO and VM accuracy on the PTB. $W + O$ and $W + O + M$ achieve the highest MTO scores on all languages of MULTEXT-East corpora while scoring the highest VM accuracies on English and Romanian. On the CoNLL-

9 Languages of MULTEXT-East corpora do not tag the punctuation marks, thus we add an extra tag for punctuation to the tag-set of these languages.

Table 7: The MTO and VM scores on 19 corpora in 15 languages together with the number of types and tags in the gold–set which equals to number of induced clusters in all languages. Best published results are from ‡(Blunsom and Cohn 2011), ⋆(Christodoulopoulos, Goldwater, and Steedman 2011) and †(Clark 2003). Bold results represent the best MTO and VM accuracies of the corresponding language with at least 90% confidence level. MULTEXT-East corpora do not tag the punctuation marks, thus we add an extra tag for punctuation and represent it with "+1".

	Language	Types	Tags	Best Published	Syntagmatic Bigram	W	W+O	W+O+M
WSJ	English	49,190	45	.775 / .697‡	.7314 / .6558	.7667 / .6819	.7820 / .7020	**.8002 / .7163**
MULTEXT-East	Bulgarian	16,352	12+1	.665 / .556⋆	.6732 / .4119	.6927 / .5341	.6964 / .5469	**.7027** / .5513
	Czech	19,115	12+1	.642 / **.539**⋆	.6269 / .4586	.7025 / .5020	.7022 / .5047	**.7045** / .5096
	English	9,773	12+1	.733 / .633⋆	.7690 /. 6131	.8239 / .6631	.8246 / .6696	**.8329 / .6769**
	Estonian	17,845	11+1	.644 / **.533**⋆	.6089 / .4119	.6612 / .4469	**.6704** / .4658	.6445 / .4452
	Hungarian	20,321	12+1	.682 / **.548**⋆	.6181 / .4514	.6900 / .4972	.6963 / .5173	**.7254** / .5402
	Romanian	15,189	14+1	.611 / .523⋆	.6565 / .5202	.6412 / .5004	**.6607** / .5262	.6432 / .5127
	Slovene	17,871	12+1	.679 / **.567**⋆	.6772 / .5044	.6914 / .4951	**.6966** / .4998	.6823 / .4938
	Serbian	18,095	12+1	.641 / **.510**†	.6267 / .4510	.6311 / .4536	.6317 / .4557	.6370 / .4648
CoNLL-X Shared Task	Bulgarian	32,439	54	.704 / **.596**†	.6972 / .5532	.7328 / .5781	**.7348** / .5844	.7321 / .5835
	Czech	130,208	12	.701‡ / .484⋆	.6944 / .5036	.6739 / .4838	**.7176 / .5336**	.7039 / .5118
	Danish	18,356	25	.761‡ / .591⋆	.6757 / .5290	.7236 / .5583	.7538 / .5962	.7417 / .5919
	Dutch	28,393	13	.711‡ / .547⋆	.6703 / .5205	.6957 / .5331	**.7401 / .5986**	.7210 / .5919
	German	72,326	54	.744⋆ / .630†	.7525 / .6285	.7669 / .6306	**.7799 / .6575**	.7557 / .6395
	Portuguese	28,931	22	.785‡ / .639⋆	.7031 / .5617	.7439 / .5798	.7901 / .6316	.7861 / .6353
	Slovene	7,128	29	.642⋆ / **.539**†	.6384 / .4976	.6513 / .4957	**.6545** / .5093	.6543 / .5031
	Spanish	16,458	47	**.788**‡ / .632⋆	.7086 / .5844	.7479 / .6086	.7712 / .6346	.7588 / .6287
	Swedish	20,057	41	.682 / **.589**†	.6721 / .5558	.6962 / .5674	**.6962** / .5721	.6675 / .5628
	Turkish	17,563	30	.628 / .408⋆	.6069 / .3551	.6239 / .3823	.6372 / .4098	**.6487 / .4206**

X languages our models perform better than the best published MTO or VM accuracies on 10 languages.

4. Discussion

In this section we perform further analysis on the clustering output of our best model and indicate the possible reasons of comparably low VM scores. To illustrate how words are distributed in the induced clusters, we compare the output of our model with gold-tags of the PTB. We also discuss the effect of coarse gold-tag sets on our model performance.

Figure 6 is the Hinton diagram of the PTB showing the relationship between the most frequent tags and clusters from the experiment in Section 3.5. In general the errors seem to be the lack of completeness (multiple large entries in a row), rather than lack of homogeneity (multiple large entries in a column). The algorithm tends to split large word classes into several clusters. Some examples are:

- Titles like Mr., Mrs., and Dr. are split from the rest of the proper nouns in cluster (39).
- Auxiliary verbs (10) and the verb "say" (22) have been split from the general verb clusters (12) and (7).

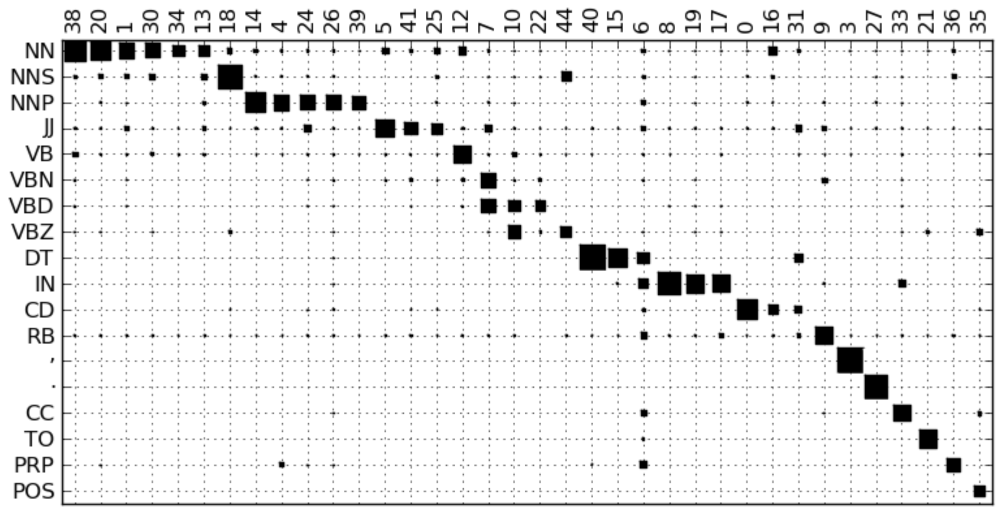

Figure 6: Hinton diagram comparing most frequent tags and clusters. Area of each square is proportional to the joint probability of the given tag and cluster.

- Determiners “the” (40), “a” (15), and capitalized “The”, “A” (6) have been split into their own clusters.
- Prepositions “of” (19), and “by”, “at” (17) have been split from the general preposition cluster (8).

Nevertheless there are some homogeneity errors as well:

- The adjective cluster (5) also has some noun members probably due to the difficulty of separating noun-noun compounds from adjective modification.
- Cluster (6) contains capitalized words that span a number of categories.

Most closed-class items are cleanly separated into their own clusters as seen in the lower right hand corner of the diagram.

The completeness errors become more noticeable on languages with coarse tag-sets thus our models perform worse than the best published models on 6 of MULTEXT-East languages in terms of VM scores while achieving the state-of-the-art MTO scores on the same languages as shown on Table 7. On CONLL-X languages the effect of completeness errors is less noticeable since all languages except Czech and Dutch have fine grained tag-sets.

The completeness errors are not surprising given that the words that have been split are not generally substitutable with the other members of their gold-tag set category. Thus it can be argued that metrics that emphasize homogeneity such as MTO are more appropriate in this context than metrics that average homogeneity and completeness such as VM as long as the number of clusters is controlled.

There are two concerns inherent in all distributional methods: (i) words that are generally substitutable like “the” and “its” are placed in separate categories (DT and PRP$) by the gold standard, (ii) words that are generally not substitutable like “do” and “put” are placed in the same category (VB). Freudenthal et al. (2005) point out that categories with unsubstitutable words fail the standard linguistic definition of a syntactic category and children do not seem to make errors of substituting such words in utterances (e.g. *"What do you want?"* vs. **"What put you want?"*). Whether gold standard part of speech tags or distributional categories are better suited to applications like parsing or machine translation can be best decided using extrinsic evaluation. In this study we evaluate our results by comparing them to gold standard part of speech tags and leave the extrinsic evaluation of the induced tags for future work.

5. Related Work

There are several good reviews of algorithms for unsupervised part of speech induction (Christodoulopoulos, Goldwater, and Steedman 2010; Gao and Johnson 2008) and models of syntactic category acquisition (Ambridge and Lieven 2011).

This work is to be distinguished from supervised part of speech disambiguation systems, which use labeled training data (Toutanova et al. 2003), unsupervised disambiguation systems, which use a dictionary of possible tags for each word (Yatbaz and Yuret 2010), or prototype driven systems which use a small set of prototypes for each class (Haghighi and Klein 2006). The problem of induction is important for studying under-resourced languages that lack labeled corpora and high quality dictionaries. It is also essential in modeling child language acquisition because every child manages to induce syntactic categories without access to labeled sentences, labeled prototypes, or dictionary constraints.

Models of unsupervised part of speech induction fall into two broad groups based on the information they utilize. Distributional models only use word types and their context statistics. Word-feature models incorporate additional morphological and orthographic features.

5.1 Distributional models

Distributional models can be further categorized into three subgroups based on the learning algorithm. The first subgroup represents each word type/token with its context vector and clusters these vectors accordingly (Schütze 1995). Work in modeling child syntactic category acquisition has generally followed this clustering approach (Redington, Crater, and Finch 1998; Mintz 2003). The second subgroup consists of probabilistic models based on the Hidden Markov Model (HMM) framework (Brown et al. 1992). A third group of algorithms constructs a low dimensional representation of the data that represents the empirical co-occurrence statistics of word types (Globerson et al. 2007), which is covered in more detail in Section 6.

Clustering. Clustering based methods represent the context using the neighboring words, typically a single word on the left and a single word on the right called a "frame" (e.g., **the** *dog* **is**; **the** *cat* **is**). They cluster word types rather than word tokens based on the frames they occupy thus employing one-tag-per-word assumption from the beginning (with the exception of (Mintz 2003; St Clair, Monaghan, and Christiansen 2010) and some methods in (Schütze 1995)). They may suffer from the data sparsity caused by the infrequent words and the infrequent contexts. The solutions suggested either restrict the set of words and set of contexts to be clustered to the most frequently observed, or use dimensionality reduction. Redington et al. (1998) define context similarity based on the number of common frames bypassing the data sparsity problem but achieve lower scores than the best performing systems. Mintz (2003) only uses the most frequent 45 frames to cluster tokens and achieves 98% unsupervised accuracy on the tokens observed in the most frequent 45 frames. Similar to Mintz's work, St Clair et al. (2010) show that systems that model the left and right frames of tokens seperately perform better than the frequent frames both interms of token clustering accuracy and the token coverage. Biemann

(2006) contructs a graph based view of the most frequent 10,000 words using contexts formed from the most frequent 150-200 words and clusters the tokens. Schütze (1995) and Lamar et al. (2010) employ SVD to enhance similarity between less frequently observed word types and contexts. Lamar et al. (2010) represent each context by the currently assigned left and right tag (which eliminates data sparsity) and cluster word types using a soft k-means style iterative algorithm. They report the best clustering result to date of .708 many-to-one accuracy on the PTB.

HMMs. The prototypical bitag HMM model maximizes the likelihood of the corpus $w_1 \dots w_n$ expressed as $P(w_1|c_1) \prod_{i=2}^{n} P(w_i|c_i)P(c_i|c_{i-1})$ where w_i are the word tokens and c_i are their (hidden) tags. One problem with such a model is its tendency to distribute probabilities equally and the resulting inability to model highly skewed word-tag distributions observed in hand-labeled data (Johnson 2007). To favor sparse word-tag distributions one can enforce a strict one-tag-per-word solution (type clustering) (Brown et al. 1992; Clark 2003), use sparse priors in a Bayesian setting (Goldwater and Griffiths 2007; Johnson 2007), or use posterior regularization (Ganchev et al. 2010). Each of these techniques provide significant improvements over the standard HMM model: for example Gao and Johnson (2008) show that sparse priors can gain from 4% (.62 to .66 on the PTB) in cross-validated many-to-one accuracy. However Christodoulopoulos et al. (2010) show that the older one-tag-per-word models such as (Brown et al. 1992) outperform the more sophisticated sparse prior and posterior regularization methods both in speed and accuracy (the Brown model gets .68 many-to-one accuracy on the PTB). Given that 93.69% of the word occurrences in human labeled data are tagged with their most frequent part of speech (Toutanova et al. 2003), this is probably not surprising; one-tag-per-word is a fairly good first approximation for induction.

5.2 Word-feature models

One problem with the algorithms in the previous section is the poverty of their input features. Of the syntactic, semantic, and morphological information linguists claim underlie syntactic categories, context vectors or bitag HMMs only represent limited syntactic information in their input. Experiments incorporating morphological and orthographic features into HMM based models demonstrate significant improvements. (Clark 2003; Berg-Kirkpatrick and Klein 2010; Blunsom and Cohn 2011) incorporate similar orthographic features and report improvements of 3, 7, and 10% respectively over the baseline Brown model.

(Clark 2003; Blunsom and Cohn 2011) cluster types by incorporating similar orthographic features and report improvements of 3 and 10% respectively over the baseline Brown model. Berg-Kirkpatrick et al. incorporate orthographic features into EM algorithm where they replace the multinomial components with miniature logistic regressions and cluster tokens while improving the Brown model by 7%.

Christodoulopoulos et al. (2010) use prototype based features as described in (Haghighi and Klein 2006) with automatically induced prototypes and report an 8% improvement over the baseline Brown model by clustering tokens. Abend et al. (2010) train a prototype-driven model with morphological features by first clustering the high frequent types as the landmarks and then assigning the remaining types to these landmark clusters. Christodoulopoulos et al. (2011) define a type-based Bayesian multinomial mixture model in which each word instance is generated from the corresponding word type mixture component and word contexts are represented as features. They achieve a .728 MTO score by extending their model with additional morphological and alignment features gathered from parallel corpora. To our knowledge, nobody has yet tried to incorporate phonological or prosodic features in a computational model for syntactic category acquisition.

5.3 Paradigmatic representations

Yatbaz et al. (2012) explore the paradigmatic representation of the word contexts by modeling the co-occurrence of words and their substitutes within the CODE framework. Their experiments on the PTB types shows that paradigmatic representation improves the state-of-the-art MTO and V-measure (VM) accuracies of both distributional models and models with additional word features. This paper builds on that preliminary work by (1) exploring induction of part-of-speechs at token level (in addition to type level), (2) improving the model for using additional features, and (3) experimenting with additional languages.

5.4 Evaluation

We report many-to-one and V-measure scores for our experiments as suggested in (Christodoulopoulos, Goldwater, and Steedman 2010). The many-to-one (MTO) evaluation maps each cluster to its most frequent gold tag and reports the percentage of correctly tagged instances. The MTO score naturally gets higher with increasing number of clusters but it is an intuitive metric when comparing results with the same number of clusters. The V-measure (VM) (Rosenberg and Hirschberg 2007) is an information theoretic metric that reports the harmonic mean of homogeneity (each cluster should contain only instances of a single class) and completeness (all instances of a class should be members of the same cluster). In Section 4 we argue that homogeneity is perhaps more important in part of speech induction and suggest MTO with a fixed number of clusters as a more intuitive metric.

6. Contributions

Our main contributions can be summarized as follows:

- We introduced substitute vectors as paradigmatic representations of word context and demonstrated their use in unsupervised part of speech induction on 19 corpora in 15 languages.
- We demonstrated that using paradigmatic representations of word context and modeling co-occurrences of word and context types with the S-CODE learning framework give superior results when compared to a syntagmatic bigram model.
- We extended the S-CODE framework to incorporate morphological and orthographic features and improved the state-of-the-art many-to-one accuracy in unsupervised part of speech induction on 17 out of 19 corpora.
- All our code and data, including the substitute vectors for the PTB, MULTEXT-East and CoNLL-X shared task corpora are available at the authors' website at `xxx.xxx.xxx`.

Appendix A: Computation of Substitute Distributions

In this study, we predict the syntactic category of a word in a given context based on its substitute distribution. The sample space of the substitute distribution is the vocabulary of the language model including the unknown word tag "<unk>". Note that the substitute distribution is a function of the context only and is indifferent to the target word.

It is best to use both the left and the right context when estimating the probabilities for potential lexical substitutes. For example, in *"He lived in San Francisco suburbs."*, the token

San would be difficult to guess from the left context but it is almost certain looking at the right context. We define c_w as the $2n-1$ word window centered around the target word position: $w_{-n+1} \dots w_0 \dots w_{n-1}$ ($n=4$ is the n-gram order we have used). The probability of a substitute word w in a given context c_w can be estimated as:

$$P(w_0 = w|c_w) \propto P(w_{-n+1} \dots w_0 \dots w_{n-1}) \tag{A.1}$$

$$= P(w_{-n+1})P(w_{-n+2}|w_{-n+1}) \dots P(w_{n-1}|w_{-n+1}^{n-2}) \tag{A.2}$$

$$\approx P(w_0|w_{-n+1}^{-1})P(w_1|w_{-n+2}^{0}) \dots P(w_{n-1}|w_0^{n-2}) \tag{A.3}$$

where w_i^j represents the sequence of words $w_i w_{i+1} \dots w_j$. In Equation A.1, $P(w|c_w)$ is proportional to $P(w_{-n+1} \dots w_0 \dots w_{n-1})$ because the words of the context are fixed. Terms without w_0 are identical for each substitute in Equation A.2 therefore they have been dropped in Equation A.3. Finally, because of the Markov property of n-gram language model, only the closest $n-1$ words are used in the experiments.

Near the sentence boundaries the appropriate terms were truncated in Equation A.3. Specifically, at the beginning of the sentence shorter n-gram contexts were used and at the end of the sentence terms beyond the end-of-sentence token were dropped.

To obtain a discrete representation of the context, the random-substitutes algorithm pairs each word token with a substitute sampled from the pre-computed substitute distribution generated from the word token's context and then word (W) – random-substitute (S) pairs are fed to the S-CODE algotihm as input.

Appendix B: The CODE Model

In this section we review the unsupervised method that we use to model co-occurrence statistics: the Co-occurrence Data Embedding (CODE) (Globerson et al. 2007) method and its spherical extension (S-CODE) introduced by (Maron, Lamar, and Bienenstock 2010).

Let W and C be two categorical variables with finite cardinalities $|W|$ and $|C|$. We observe a set of pairs $\{w_i, c_i\}_{i=1}^n$ drawn IID from the joint distribution of W and C. The basic idea behind CODE and related methods is to represent (embed) each value of W and each value of C as points in a common Euclidean space $\mathbf{R}^d$ such that values that frequently co-occur lie close to each other. There are several ways to formalize the relationship between the distances and co-occurrence statistics, in this paper we use the following:

$$p(w,c) = \frac{1}{Z}\bar{p}(w)\bar{p}(c)e^{-d_{w,c}^2} \tag{B.1}$$

where $d_{w,c}^2$ is the squared distance between the embeddings of w and c, $\bar{p}(w)$ and $\bar{p}(c)$ are empirical probabilities, and $Z = \sum_{w,c} \bar{p}(w)\bar{p}(c)e^{-d_{w,c}^2}$ is a normalization term. If we use the notation ϕ_w for the point corresponding to w and ψ_c for the point corresponding to c then

$d^2_{w,c} = \|\phi_w - \psi_c\|^2$. The log-likelihood of a given embedding $\ell(\phi, \psi)$ can be expressed as:

$$\begin{aligned}\ell(\phi, \psi) &= \sum_{w,c} \bar{p}(w,c) \log p(w,c) \\ &= \sum_{w,c} \bar{p}(w,c)(-\log Z + \log \bar{p}(w)\bar{p}(c) - d^2_{w,c}) \\ &= -\log Z + const - \sum_{w,c} \bar{p}(w,c) d^2_{w,c}\end{aligned} \tag{B.2}$$

The likelihood is not convex in ϕ and ψ. We use gradient ascent to find an approximate solution for a set of ϕ_w, ψ_c that maximize the likelihood. The gradient of the $d^2_{w,c}$ term pulls neighbors closer in proportion to the empirical joint probability:

$$\frac{\partial}{\partial \phi_w} \sum_{w,c} -\bar{p}(w,c) d^2_{w,c} = \sum_{y} 2\bar{p}(w,c)(\psi_c - \phi_w) \tag{B.3}$$

The gradient of the Z term pushes neighbors apart in proportion to the estimated joint probability:

$$\frac{\partial}{\partial \phi_x} (-\log Z) = \sum_{y} 2p(w,c)(\phi_w - \psi_c) \tag{B.4}$$

Thus the net effect is to pull pairs together if their estimated probability is less than the empirical probability and to push them apart otherwise. The gradients with respect to ψ_c are similar. S-CODE (Maron, Lamar, and Bienenstock 2010) additionally restricts all ϕ_w and ψ_c to lie on the unit sphere. With this restriction, Z stays around a fixed value during gradient ascent. This allows S-CODE to substitute an approximate constant $\tilde{Z}$ in gradient calculations for the real Z for computational efficiency. In our experiments, we used S-CODE with its sampling based stochastic gradient ascent algorithm and smoothly decreasing learning rate.

Appendix C: S-CODE with More than Two Variables

In order to accommodate multiple feature types the S-CODE model in the previous section needs to be extended to handle more than two variables. Globerson et. al (2007) suggest the following likelihood function:

$$\ell(\phi, \psi^{(1)}, \ldots, \psi^{(K)}) = \bar{p}(w,c) \log p(w,c) + \sum_{i}^{K} \sum_{w,f^{(i)}} \bar{p}(w, f^{(i)}) \log p(w, f^{(i)}) \tag{C.1}$$

where $\bar{p}(w,c)$ is the empirical joint distribution of context C with W, $F^{(1)}, \ldots, F^{(K)}$ are extra K different variables whose empirical joint distributions with W, $\bar{p}(w, f^{(1)}) \ldots \bar{p}(w, f^{(K)})$, are known. Eq. C.1 then represents a set of CODE models $p(w, f^{(k)})$ where each $F^{(k)}$ has an embedding $\psi^{(k)}_f$ but all models share the same ϕ_w embedding.

We adopt this likelihood function, let W represent a word, C represent a context (i.e., random substitute), and $F^{(1)}, \ldots, F^{(K)}$ stand for morphological and orthographic features of the word thus each co-occurrence is a (K+1)-tuple, $(W, C, F^{(1)}, \ldots F^{(K)})$. With this setup, the training procedure needs to change little: instead of sampling a word (w) – context (c), the word (w) – context (c) – features ($f_1, \ldots, f_K$) tuple is sampled and input to the gradient ascent algorithm. The gradient search algorithm updates the embeddings according to $p(w, c)$ and $p(w, f^{(i)})$ where $i = 1 \ldots k$ and no updates are performed between c and $f^{(i)}$s since they do not have any co-occurrence statistics and w is the only shared variable.

Tuples might have null values due to unobserved features. For example in the case of POS induction, the word "**car**" has no morphological or orthographic features therefore all the elements of the tuple have null value except the word type (w) and the context (c). We do not perform any pull or push updates on embeddings during the gradient search if the corresponding $f^{(k)}$ is null[10].

Appendix D: Language Statistics

This section explains the language model training and feature extraction of each language that we apply our model in Section 3.7.

Statictical Language Modeling. For all languages except Serbian, English and Turkish, we train the language models by using the corresponding Wikipedia dump files[11]. Serbian shares a common basis with Croatian and Bosnian therefore we trained 3 different language models using Wikipedia dump files of Serbian together with these two languages and measured the perplexities on the MULTEXT-East Serbian corpus. We chose the Croatian language model since it achieved the lowest perplexity score and unknown word ratio on the MULTEXT-East Serbian corpus. To train the statistical language model of English, we use Wall Street Journal data (1987-1994) extracted from CSR-III Text (Graff, Rosenfeld, and Paul 1995) (excluding sections of the PTB) and for the Turkish language modeling we use the web corpus collected from Turkish news and blog sites (Sak, Güngör, and Saraçlar 2008).

In order to reduce the unknown word ratio of resource poor languages and to standardize the process we set the vocabulary threshold to 2 for all languages except English. English has a relatively low unknown word ratio therefore we set the threshold to 20 instead of 2. Table 1 summarizes the language model related statistics and scores that vary across the languages in terms of quality and quantity.

Feature extraction. Morphological features of each language are extracted using the training sections of the corresponding MULTEXT-East and CoNLL-X corpora. We don't use the language model corpora to extract morphological features. Number of morphological feature of each language is presented in Table 1. We use the same set of orthographic features described in Section 3.5 except we add an "Only-Punctuation" feature to the languages of MULTEXT-East corpora. The "Only-Punctuation" feature is generated when a token only consists of punctuation characters.

10 In the POS induction problem w and c represents the word type and context therefore they are always observed.

11 Latest Wikipedia dump files are freely available at `http://dumps.wikimedia.org/` and the text in the dump files can be extracted using WP2TXT (`http://wp2txt.rubyforge.org/`)

Table 1: Last two columns present the number of induced suffix parts and word types with these suffix parts after the morphological feature extraction.

		Language Model		Test set				
	Language	Source	Word Count	Word Count	Perplexity (ppl)	Unknown Word	Suffix Parts	Word Types with Suffix parts
WSJ	English	News	126,170,376	1,173,766	79.926	0.012	5575	19223
MULTEXT-East	Bulgarian	Wikipedia	32,511,616	101,173	655.202	.0565	609	4209
	Czech	Wikipedia	59,698,049	100,368	1,069.67	.0299	2787	12848
	English	News	126,170,376	118,424	265.246	.0288	1251	4783
	Estonian	Wikipedia	14,513,571	94,898	871.765	.0654	4448	13638
	Hungarian	Wikipedia	66,069,788	98,426	742.676	.0449	5423	15995
	Romanian	Wikipedia	35680870	118,328	666.855	.1074	2064	9445
	Slovene	Wikipedia	18,969,846	112,278	658.711	.0389	2093	11834
	Serbian	Wikipedia	17,129,679	108,809	804.962	.0580	2722	12476
CoNLL-X Shared Task	Bulgarian	Wikipedia	32,511,616	190,217	538.972	.0430	926	8225
	Czech	Wikipedia	59,698,049	1,249,408	1,233.95	.0250	12443	85673
	Danish	Wikipedia	35,863,945	94,386	351.24	.0393	3708	10897
	Dutch	Wikipedia	159,978,524	195,069	390.818	.0476	5250	13407
	German	Wikipedia	437,777,863	699,610	680.036	.0487	15219	45414
	Portuguese	Wikipedia	150,099,154	206,678	378.656	.0861	5033	15721
	Slovene	Wikipedia	18,969,846	28,750	663.053	.0414	1257	4781
	Spanish	Wikipedia	332,311,650	89,334	274.418	.0424	2648	9316
	Swedish	Wikipedia	32,004,538	191,467	1,233.95	.0250	2897	12725
	Turkish	Web	491,195,991	47,605	868.829	.0508	5651	14227

References

Abend, Omri, Roi Reichart, and Ari Rappoport. 2010. Improved unsupervised pos induction through prototype discovery. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1298–1307, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ambridge, B. and E.V.M. Lieven, 2011. *Child Language Acquisition: Contrasting Theoretical Approaches*, chapter 6.1. Cambridge University Press.

Arthur, D. and S. Vassilvitskii. 2007. 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.

Berg-Kirkpatrick, Taylor, Alexandre Bouchard-Côté, John DeNero, and Dan Klein. 2010. 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. Association for Computational Linguistics.

Berg-Kirkpatrick, Taylor and Dan Klein. 2010. Phylogenetic grammar induction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1288–1297, Uppsala, Sweden, July. Association for Computational Linguistics.

Biemann, C. 2006. 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.

Blunsom, Phil and Trevor Cohn. 2011. 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. Association

for Computational Linguistics.
Brown, Peter F., Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. 1992. Class-based n-gram models of natural language. *Comput. Linguist.*, 18:467–479, December.
Buchholz, Sabine and Erwin Marsi. 2006. Conll-x shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, CoNLL-X ’06, pages 149–164, Stroudsburg, PA, USA. Association for Computational Linguistics.
Chandler, D. 2007. *Semiotics: the basics*. The Basics Series. Routledge.
Christodoulopoulos, Christos, Sharon Goldwater, and Mark Steedman. 2010. 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. Association for Computational Linguistics.
Christodoulopoulos, Christos, Sharon Goldwater, and Mark Steedman. 2011. 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. Association for Computational Linguistics.
Clark, Alexander. 2003. 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. Association for Computational Linguistics.
Creutz, Mathias and Krista Lagus. 2005. 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.
Erjavec, Tomaž. 2004. MULTEXT-east version 3: Multilingual morphosyntactic specifications, lexicons and corpora. In *Fourth International Conference on Language Resources and Evaluation, LREC’04*, pages 1535–1538. ELRA.
Freudenthal, D., J.M. Pine, and F. Gobet. 2005. On the resolution of ambiguities in the extraction of syntactic categories through chunking. *Cognitive Systems Research*, 6(1):17–25.
Ganchev, Kuzman, João Graça, Jennifer Gillenwater, and Ben Taskar. 2010. Posterior regularization for structured latent variable models. *J. Mach. Learn. Res.*, 99:2001–2049, August.
Gao, Jianfeng and Mark Johnson. 2008. A comparison of bayesian estimators for unsupervised hidden markov model pos taggers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP ’08, pages 344–352, Stroudsburg, PA, USA. Association for Computational Linguistics.
Globerson, Amir, Gal Chechik, Fernando Pereira, and Naftali Tishby. 2007. Euclidean embedding of co-occurrence data. *J. Mach. Learn. Res.*, 8:2265–2295, December.
Goldwater, Sharon and Tom Griffiths. 2007. 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. Association for Computational Linguistics.
Graff, David, Roni Rosenfeld, and Doug Paul. 1995. Csr-iii text. Linguistic Data Consortium, Philadelphia.
Haghighi, Aria and Dan Klein. 2006. 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*, HLT-NAACL ’06, pages 320–327, Stroudsburg, PA, USA. Association for Computational Linguistics.
Johnson, Mark. 2007. Why doesn’t EM find good HMM POS-taggers? In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 296–305, Prague, Czech Republic, June. Association for Computational Linguistics.
Lamar, Michael, Yariv Maron, and Elie Bienenstock. 2010. 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. Association for Computational Linguistics.
Lamar, Michael, Yariv Maron, Mark Johnson, and Elie Bienenstock. 2010. Svd and clustering for unsupervised pos tagging. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 215–219, Uppsala, Sweden, July. Association for Computational Linguistics.
Lee, Yoong Keok, Aria Haghighi, and Regina Barzilay. 2010. Simple type-level unsupervised pos tagging. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP ’10, pages 853–861, Stroudsburg, PA, USA. Association for Computational Linguistics.
MacWhinney, B. 2000. *The CHILDES Project: Tools for Analyzing Talk, Volume II: The Database*, volume 2. Lawrence Erlbaum.

Manning, Christopher D. and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts.
Marcus, Mitchell P., Beatrice Santorini, Mary Ann Marcinkiewicz, and Ann Taylor. 1999. Treebank-3. Linguistic Data Consortium, Philadelphia.
Maron, Yariv, Michael Lamar, and Elie Bienenstock. 2010. 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.
Mintz, T.H. 2003. Frequent frames as a cue for grammatical categories in child directed speech. *Cognition*, 90(1):91–117.
Redington, M., N. Crater, and S. Finch. 1998. Distributional information: A powerful cue for acquiring syntactic categories. *Cognitive Science*, 22(4):425–469.
Rosenberg, A. and J. Hirschberg. 2007. V-measure: A conditional entropy-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.
Sahlgren, Magnus. 2006. *The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces*. Ph.D. thesis, Stockholm University.
Sak, H., T. Güngör, and M. Saraçlar. 2008. Turkish language resources: Morphological parser, morphological disambiguator and web corpus. *Advances in natural language processing*, pages 417–427.
Schütze, H. and J. Pedersen. 1993. A Vector Model for syntagmatic and paradigmatic relatedness. In *Proceedings of the 9th Annual Conference of the University of Waterloo Centre for the New OED and Text Research*, Oxford, England.
Schütze, Hinrich. 1995. Distributional part-of-speech tagging. In *Proceedings of the seventh conference on European chapter of the Association for Computational Linguistics*, EACL ’95, pages 141–148, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
St Clair, Michelle C, Padraic Monaghan, and Morten H Christiansen. 2010. Learning grammatical categories from distributional cues: flexible frames for language acquisition. *Cognition*, 116(3):341–60.
Stolcke, Andreas. 2002. Srilm-an extensible language modeling toolkit. In *Proceedings International Conference on Spoken Language Processing*, pages 257–286, November.
Toutanova, Kristina, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL ’03, pages 173–180, Stroudsburg, PA, USA. Association for Computational Linguistics.
Turney, Peter D. and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *J. Artif. Intell. Res. (JAIR)*, 37:141–188.
Yatbaz, Mehmet Ali, Enis Sert, and Deniz Yuret. 2012. 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, Jeju Island, Korea, July. Association for Computational Linguistics.
Yatbaz, Mehmet Ali and Deniz Yuret. 2010. Unsupervised part of speech tagging using unambiguous substitutes from a statistical language model. In *Coling 2010: Posters*, pages 1391–1398, Beijing, China, August. Coling 2010 Organizing Committee.
Yuret, Deniz. 2012. 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.