Learning Syntactic Categories Using Paradigmatic Representations of Word Context

Mehmet Ali Yatbaz **Enis Sert** **Deniz Yuret**
Artificial Intelligence Laboratory
Koç University, İstanbul, Turkey
{myatbaz,esert,dyuret}@ku.edu.tr

Abstract

We investigate paradigmatic representations of word context in the domain of unsupervised syntactic category acquisition. Paradigmatic representations of word context are based on potential substitutes of a word in contrast to syntagmatic representations based on properties of neighboring words. We compare a bigram based baseline model with several paradigmatic models and demonstrate significant gains in accuracy. Our best model based on Euclidean co-occurrence embedding combines the paradigmatic context representation with morphological and orthographic features and achieves 80% many-to-one accuracy on a 45-tag 1M word corpus.

1 Introduction

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

Syntactic categories represent groups of words that can be substituted for one another without altering the grammaticality of a sentence. Linguists identify syntactic categories based on semantic, syntactic, and morphological properties of words. There is also evidence that children use prosodic and phonological features to bootstrap syntactic category acquisition (Ambridge and Lieven, 2011). However there is as yet no satisfactory computational model that can match human performance. Thus identifying the best set of features and best learning algorithms for syntactic category acquisition 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.

In this study, we represent the paradigmatic axis directly by building *substitute vectors* for each word position in the text. The dimensions of a substitute vector represent words in the vocabulary, and the magnitudes represent the probability of occurrence in the given position. Note that the substitute vector for a word position (e.g. the second word in Fig. 1) is a function of the context only (i.e. "the ___ cried"), and does not depend on the word that does actually appear there (i.e. "man"). Thus substi-

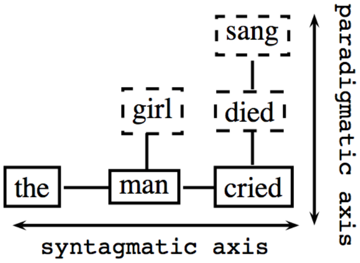

Figure 1: Syntagmatic vs. paradigmatic axes for words in a simple sentence (Chandler, 2007).

Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 940–951, Jeju Island, Korea, 12–14 July 2012.

tute vectors represent *individual word contexts*, not word types. We refer to the use of features based on substitute vectors as *paradigmatic representations of word context*.

Our preliminary experiments indicated that using context information alone without the identity or the features of the target word (e.g. using dimensionality reduction and clustering on substitute vectors) has limited success and modeling the co-occurrence of word and context types is essential for inducing syntactic categories. In the models presented in this paper, we combine paradigmatic representations of word context with features of co-occurring words within the co-occurrence data embedding (CODE) framework (Globerson et al., 2007; Maron et al., 2010). The resulting embeddings for word types are split into 45 clusters using k-means and the clusters are compared to the 45 gold tags in the 1M word Penn Treebank Wall Street Journal corpus (Marcus et al., 1999). We obtain many-to-one accuracies up to .7680 using only distributional information (the identity of the word and a representation of its context) and .8023 using morphological and orthographic features of words improving the state-of-the-art in unsupervised part-of-speech tagging performance.

The high probability substitutes reflect both semantic and syntactic properties of the context as seen in the example below (the numbers in parentheses give substitute probabilities):

> *"Pierre Vinken, 61 years old, will join the board as a nonexecutive director Nov. 29."*
>
> **the:** its (.9011), the (.0981), a (.0006), . . .
> **board:** board (.4288), company (.2584), firm (.2024), bank (.0731), . . .

Top substitutes for the word "the" consist of words that can act as determiners. Top substitutes for "board" are not only nouns, but specifically nouns compatible with the semantic context.

This example illustrates 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. However in this study we follow previous work and evaluate our results by comparing them to gold standard part-of-speech tags.

Section 2 gives a detailed review of related work. Section 3 describes the dataset and the construction of the substitute vectors. Section 4 describes co-occurrence data embedding, the learning algorithm used in our experiments. Section 5 describes our experiments and compares our results with previous work. Section 6 gives a brief error analysis and Section 7 summarizes our contributions. All the data and the code to replicate the results given in this paper is available from the authors' website at `http://goo.gl/RoqEh`.

2 Related Work

There are several good reviews of algorithms for unsupervised part-of-speech induction (Christodoulopoulos et al., 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 (Church, 1988), unsupervised disambiguation systems, which use a dictionary of possible tags for each word (Merialdo, 1994), 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.

2.1 Distributional models

Distributional models can be further categorized into three subgroups based on the learning algorithm. The first subgroup represents each word type 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 et al., 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 4.

Clustering: Clustering based methods represent context using 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 some methods in (Schütze, 1995)). They may suffer from data sparsity caused by infrequent words and 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 mediocre results. Mintz (2003) only uses the most frequent 45 frames and Biemann (2006) clusters the most frequent 10,000 words using contexts formed from the most frequent 150-200 words. Schütze (1995) and Lamar et al. (2010b) employ SVD to enhance similarity between less frequently observed words and contexts. Lamar et al. (2010a) 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 a 45-tag 1M word corpus.

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 (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 with a 1M word corpus) 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 with a 1M word corpus). Given that close to 95% of the word occurrences in human labeled data are tagged with their most frequent part of speech (Lee et al., 2010), this is probably not surprising; one-tag-per-word is a fairly good first approximation for induction.

2.2 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. 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. 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.

2.3 Paradigmatic representations

Sahlgren (2006) gives a detailed analysis of paradigmatic and syntagmatic relations in the context of word-space models used to represent word meaning. 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. We find that representing the paradigmatic axis more directly using substitute vectors rather than frequent neighbors improve part-of-speech induction.

Our paradigmatic representation is also related to the second order co-occurrences used in (Schütze, 1995). Schütze concatenates the left and right context vectors for the target word type with the left context vector of the right neighbor and the right context vector of the left neighbor. The vectors from the neighbors include potential substitutes. Our method improves on his foundation by using a 4-gram language model rather than bigram statistics, using the whole 78,498 word vocabulary rather than the most frequent 250 words. More importantly, rather than simply concatenating vectors that represent the target word with vectors that represent the context we use S-CODE to model their co-occurrence statistics.

2.4 Evaluation

We report many-to-one and V-measure scores for our experiments as suggested in (Christodoulopoulos et al., 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 6 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.

3 Substitute Vectors

In this study, we predict the part of speech of a word in a given context based on its substitute vector. The dimensions of the substitute vector represent words in the vocabulary, and the entries in the substitute vector represent the probability of those words being used in the given context. Note that the substitute vector is a function of the context only and is indifferent to the target word. This section details the choice of the data set, the vocabulary and the estimation of substitute vector probabilities.

The Wall Street Journal Section of the Penn Treebank (Marcus et al., 1999) was used as the test corpus (1,173,766 tokens, 49,206 types). 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 substitute probabilities we trained a language model using approximately 126 million tokens of Wall Street Journal data (1987-1994) extracted from CSR-III Text (Graff et al., 1995) (we excluded the test corpus). 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.

It is best to use both left and 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). The probability of a substitute word w in a given context c_w can be estimated as:

$$\begin{aligned} P(w_0 = w | c_w) &\propto P(w_{-n+1} \dots w_0 \dots w_{n-1}) && (1) \\ &= P(w_{-n+1}) P(w_{-n+2} | w_{-n+1}) \\ &\quad \dots P(w_{n-1} | w_{-n+1}^{n-2}) && (2) \\ &\approx P(w_0 | w_{-n+1}^{-1}) P(w_1 | w_{-n+2}^{0}) \\ &\quad \dots P(w_{n-1} | w_0^{n-2}) && (3) \end{aligned}$$

where w_i^j represents the sequence of words $w_i w_{i+1} \dots w_j$. In Equation 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 2 therefore they have been dropped in Equation 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 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.

For computational efficiency only the top 100 substitutes and their unnormalized probabilities were computed for each of the 1,173,766 positions in the test set[1]. The probability vectors for each position were normalized to add up to 1.0 giving us the final substitute vectors used in the rest of this study.

4 Co-occurrence Data Embedding

The general strategy we follow for unsupervised syntactic category acquisition is to combine features of the context with the identity and features of the target word. Our preliminary experiments indicated that using the context information alone (e.g. clustering substitute vectors) without the target word identity and features had limited success.[2] It is the co-occurrence of a target word with a particular type of context that best predicts the syntactic category. In this section we review the unsupervised methods we used to model co-occurrence statistics: the Co-occurrence Data Embedding (CODE) method (Globerson et al., 2007) and its spherical extension (S-CODE) introduced by (Maron et al., 2010).

Let X and Y be two categorical variables with finite cardinalities $|X|$ 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. The basic idea behind CODE and related methods is to represent (embed) each value of X and each value of Y as points in a common low dimensional 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:

$$p(x, y) = \frac{1}{Z} \bar{p}(x) \bar{p}(y) e^{-d_{x,y}^2} \qquad (4)$$

where $d_{x,y}^2$ is the squared distance between the embeddings of x and y, $\bar{p}(x)$ and $\bar{p}(y)$ are empirical probabilities, and $Z = \sum_{x,y} \bar{p}(x) \bar{p}(y) e^{-d_{x,y}^2}$ is a normalization term. If we use the notation ϕ_x for the point corresponding to x and ψ_y for the point corresponding to y then $d_{x,y}^2 = \|\phi_x - \psi_y\|^2$. The log-likelihood of a given embedding $\ell(\phi, \psi)$ can be

[1] The substitutes with unnormalized log probabilities can be downloaded from `http://goo.gl/jzKH0`. For a description of the FASTSUBS algorithm used to generate the substitutes please see `http://arxiv.org/abs/1205.5407v1`. FASTSUBS accomplishes this task in about 5 hours, a naive algorithm that looks at the whole vocabulary would take more than 6 days on a typical 2012 workstation.

[2] A 10-nearest-neighbor supervised baseline using cosine distance between substitute vectors gives .7213 accuracy. Clustering substitute vectors using various distance metrics and dimensionality reduction methods give results inferior to this upper bound.

expressed as:

$$\begin{aligned}\ell(\phi,\psi) &= \sum_{x,y} \bar{p}(x,y)\log p(x,y) \qquad (5)\\ &= \sum_{x,y} \bar{p}(x,y)(-\log Z + \log \bar{p}(x)\bar{p}(y) - d^2_{x,y})\\ &= -\log Z + const - \sum_{x,y} \bar{p}(x,y) d^2_{x,y}\end{aligned}$$

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

$$\frac{\partial}{\partial \phi_x} \sum_{x,y} -\bar{p}(x,y) d^2_{x,y} = \sum_y 2\bar{p}(x,y)(\psi_y - \phi_x) \qquad (6)$$

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

$$\frac{\partial}{\partial \phi_x}(-\log Z) = \sum_y 2p(x,y)(\phi_x - \psi_y) \qquad (7)$$

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_y are similar.

S-CODE (Maron et al., 2010) additionally restricts all ϕ_x and ψ_y 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.

5 Experiments

In this section we present experiments that evaluate substitute vectors as representations of word context within the S-CODE framework. Section 5.1 replicates the bigram based S-CODE results from (Maron et al., 2010) as a baseline. The S-CODE algorithm works with discrete inputs. The substitute vectors as described in Section 3 are high dimensional and continuous. We experimented with two approaches to use substitute vectors in a discrete setting. Section 5.2 presents an algorithm that partitions the high dimensional space of substitute vectors into small neighborhoods and uses the partition id as a discrete context representation. Section 5.3 presents an even simpler model which pairs each word with a random substitute. When the left-word – right-word pairs used in the bigram model are replaced with word – partition-id or word – substitute pairs we see significant gains in accuracy. These results support our running hypothesis that paradigmatic features, i.e. potential substitutes of a word, are better determiners of syntactic category compared to left and right neighbors. Section 5.4 explores morphologic and orthographic features as additional sources of information and its results improve the state-of-the-art in the field of unsupervised syntactic category acquisition.

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 1 summarizes all the results reported in this paper and the ones we cite from the literature.

5.1 Bigram model

In (Maron et al., 2010) adjacent word pairs (bigrams) in the corpus are fed into the S-CODE algorithm as X, Y samples. The algorithm uses stochastic gradient ascent to find the ϕ_x, ψ_y embeddings for left and right words in these bigrams on a single 25-dimensional sphere. 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 an instance weighted k-means algorithm and the resulting groups are compared to the correct part-of-speech tags. Maron et al. (2010) report many-to-one scores of .6880 (.0016) for 45 clusters and .7150 (.0060) for 50 clusters (on the full PTB45 tag-set). If only ϕ_w vectors are clustered without concatenation we found the performance drops significantly to about .62.

To make a meaningful comparison we re-ran the bigram experiments using our default settings and obtained a many-to-one score of .7314 (.0096) and the V-measure is .6558 (.0052) for 45 clusters. The following default settings were used: (i) each word

Distributional Models	MTO	VM	Models with Additional Features	MTO	VM
(Lamar et al., 2010a)	.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 and Klein, 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
Bigrams (Sec. 5.1)	.7314 (.0096)	.6558 (.0052)	Substitutes and Features (Sec. 5.4)	.8023 (.0070)	.7207 (.0041)
Partitions (Sec. 5.2)	.7554 (.0055)	.6703 (.0037)			
Substitutes (Sec. 5.3)	.7680 (.0038)	.6822 (.0029)			

Table 1: 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 et al., 2010). Distributional models use only the identity of the target word and its context. The models on the right incorporate orthographic and morphological features.

was kept with its original capitalization, (ii) the learning rate parameters were adjusted to $\varphi_0 = 50$, $\eta_0 = 0.2$ for faster convergence in log likelihood, (iii) the number of s-code iterations were increased from 12 to 50 million, (iv) k-means initialization was improved using (Arthur and Vassilvitskii, 2007), and (v) the number of k-means restarts were increased to 128 to improve clustering and reduce variance.

5.2 Random partitions

Instead of using left-word – right-word pairs as inputs to S-CODE we wanted to pair each word with a paradigmatic representation of its context to get a direct comparison of the two context representations. To obtain a discrete representation of the context, the random–partitions algorithm first designates a random subset of substitute vectors as centroids to partition the space, and then associates each context with the partition defined by the closest centroid in cosine distance. Each partition thus defined gets a unique id, and word (X) – partition-id (Y) pairs are given to S-CODE as input. The algorithm cycles through the data until we get approximately 50 million updates. The resulting ϕ_x vectors are clustered using the k-means algorithm (no vector concatenation is necessary). Using default settings (64K random partitions, 25 s-code dimensions, $Z = 0.166$) the many-to-one accuracy is .7554 (.0055) and the V-measure is .6703 (.0037).

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 2 gives results where the number of initial random partitions is varied over a large range and shows the results to be fairly stable across two orders of magnitude.

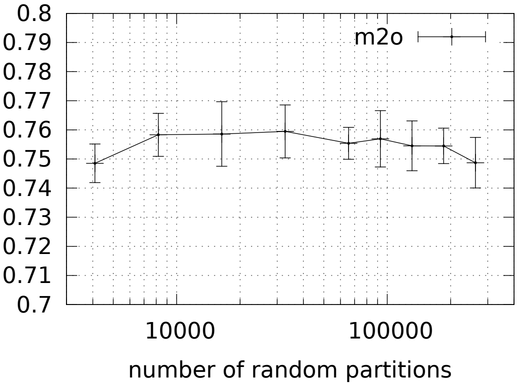

Figure 2: MTO is not sensitive to the number of partitions used to discretize the substitute vector space within our experimental range.

Figure 3 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 4 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. 6 dominates the repulsion in Eq. 7 and all points tend to converge to the same location. When $\tilde{Z}$ is too high, it prevents meaningful clusters from coalesc-

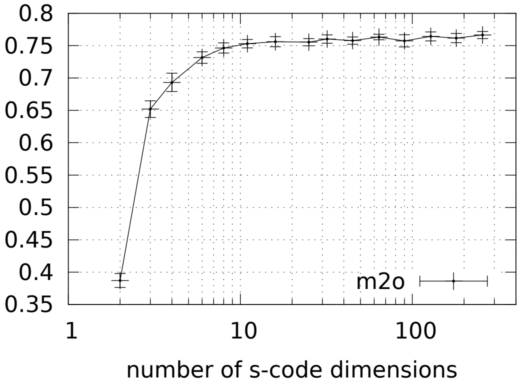

Figure 3: MTO falls sharply for less than 10 S-CODE dimensions, but more than 25 do not help.

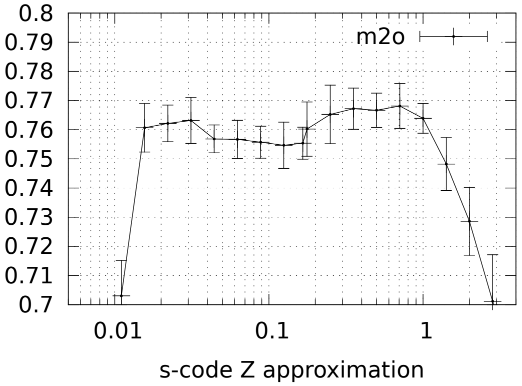

Figure 4: MTO is fairly stable as long as the $\tilde{Z}$ constant is within an order of magnitude of the real Z value.

ing.

We find the random partition algorithm to be fairly robust to different parameter settings and the resulting many-to-one score significantly better than the bigram baseline.

5.3 Random substitutes

Another way to use substitute vectors in a discrete setting is simply to sample individual substitute words from them. The random-substitutes algorithm cycles through the test data and pairs each word with a random substitute picked from the precomputed substitute vectors (see Section 3). We ran the random-substitutes algorithm to generate 14 million word (X) – random-substitute (Y) pairs (12 substitutes for each token) as input to S-CODE. Clustering the resulting ϕ_x vectors yields a many-to-one score of .7680 (.0038) and a V-measure of .6822 (.0029).

This result is close to the previous result by the random-partition algorithm, .7554 (.0055), demonstrating that two very different discrete representations of context based on paradigmatic features give consistent results. Both results are significantly above the bigram baseline, .7314 (.0096). Figure 5 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.

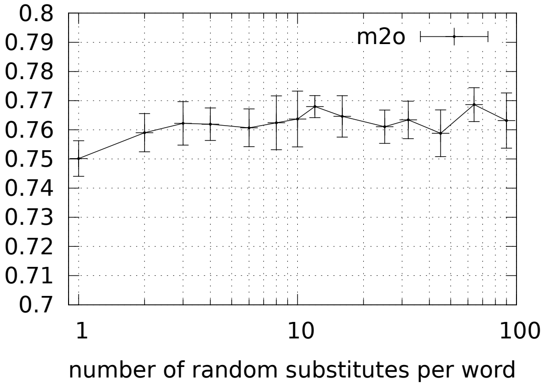

Figure 5: MTO is not sensitive to the number of random substitutes sampled per word token.

5.4 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 2 describes a number other approaches that show similar improvements. This section describes one way to integrate additional features to the random-substitute model.

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 300. The program induced 5 suffix types that are present in a total of 10,484 word types. These suffixes were input to S-CODE as morphological features whenever the associated word types were sampled.

In order to incorporate morphological and orthographic features into S-CODE we modified its input. For each word – random-substitute pair generated as in the previous section, we added word – feature pairs to the input for each morphological and orthographic feature of the word. Words on average have 0.25 features associated with them. This increased the number of pairs input to S-CODE from 14.1 million (12 substitutes per word) to 17.7 million (additional 0.25 features on average for each of the 14.1 million words).

Using similar training settings as the previous section, the addition of morphological and orthographic features increased the many-to-one score of the random-substitute model to .8023 (.0070) and V-measure to .7207 (.0041). Both these results improve the state-of-the-art in part-of-speech induction significantly as seen in Table 1.

6 Error Analysis

Figure 6 is the Hinton diagram showing the relationship between the most frequent tags and clusters from the experiment in Section 5.4. 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).

- 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 are not surprising given that the words that have been split are not generally substitutable with the other members of their Penn Treebank 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.

7 Contributions

Our main contributions can be summarized as follows:

- We introduced substitute vectors as paradigmatic representations of word context and demonstrated their use in syntactic category acquisition.

- 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 baseline bigram model.

- We extended the S-CODE framework to incorporate morphological and orthographic features and improved the state-of-the-art in unsupervised part-of-speech induction to 80% many-to-one accuracy.

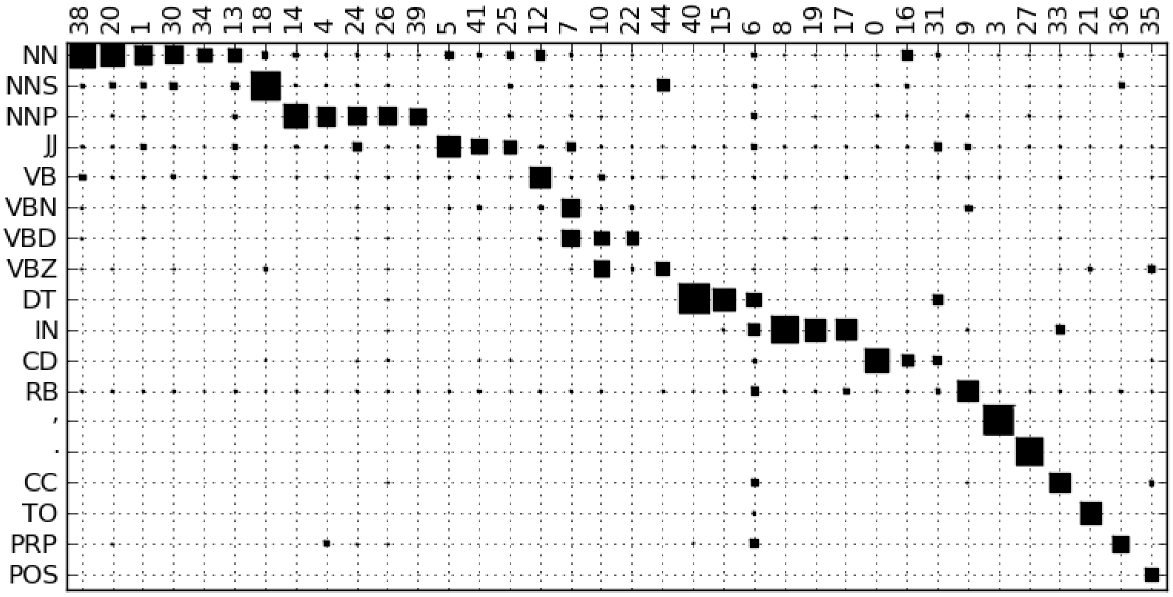

Figure 6: Hinton diagram comparing most frequent tags and clusters.

- All our code and data, including the substitute vectors for the one million word Penn Treebank Wall Street Journal dataset, is available at the authors' website at `http://goo.gl/RoqEh`.

References

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

D. Arthur 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.

Taylor Berg-Kirkpatrick 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.

Taylor Berg-Kirkpatrick, 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.

C. Biemann. 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.

Phil Blunsom 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.

Peter F. Brown, 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.

D. Chandler. 2007. *Semiotics: the basics*. The Basics Series. Routledge.

Christos Christodoulopoulos, 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.

Christos Christodoulopoulos, 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.

Kenneth Ward Church. 1988. A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of the second conference on Applied natural language processing*, ANLC '88, pages 136–143, Stroudsburg, PA, USA. Association for Computational Linguistics.

Alexander Clark. 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.

Mathias Creutz 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.

D. Freudenthal, 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.

Kuzman Ganchev, João Graça, Jennifer Gillenwater, and Ben Taskar. 2010. Posterior regularization for structured latent variable models. *J. Mach. Learn. Res.*, 99:2001–2049, August.

Jianfeng Gao 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.

Amir Globerson, Gal Chechik, Fernando Pereira, and Naftali Tishby. 2007. Euclidean embedding of cooccurrence data. *J. Mach. Learn. Res.*, 8:2265–2295, December.

Sharon Goldwater 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.

David Graff, Roni Rosenfeld, and Doug Paul. 1995. Csr-iii text. Linguistic Data Consortium, Philadelphia.

Aria Haghighi 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.

Mark Johnson. 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.

Michael Lamar, Yariv Maron, and Elie Bienenstock. 2010a. 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.

Michael Lamar, Yariv Maron, Mark Johnson, and Elie Bienenstock. 2010b. 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.

Yoong Keok Lee, 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.

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

Yariv Maron, 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.

Bernard Merialdo. 1994. Tagging english text with a probabilistic model. *Comput. Linguist.*, 20:155–171, June.

T.H. Mintz. 2003. Frequent frames as a cue for grammatical categories in child directed speech. *Cognition*, 90(1):91–117.

M. Redington, N. Crater, and S. Finch. 1998. Distributional information: A powerful cue for acquiring syntactic categories. *Cognitive Science*, 22(4):425–469.

A. Rosenberg 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.

Magnus Sahlgren. 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.

Hinrich Schütze. 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.

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