Draft Paper v.2

Unsupervised Part of Speech Induction Using Paradigmatic Representations of Word Context

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

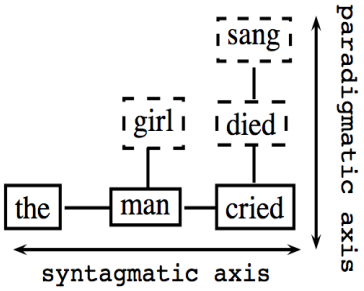

Figure 1
Syntagmatic vs. paradigmatic axes for words in a simple sentence (Chandler 2007).

© 2012 Association for Computational Linguistics

appear there (i.e. "man"). Thus substitute 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*.

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 and left the extrinsic evaluation as a future work.

Next example demonstrates the advantage of the paradigmatic representation over the syntagmatic representation for the words occur in different contexts.

> *"Blacks and Hispanics currently make up 38 % of the city 's population and hold only 25 % of the seats on the* **council** *."*
> **council:** board (.6591), company (.0795), firm (.0542), bank (.0154), . . .

The word "council" and the word "board" from the previous example have completely different contexts except the word "the". The paradigmatic representation captures the similarity of these words by suggesting the same top substitutes for both words while the sytagmatic representation fails to capture similarity due to the distinction of contexts.

Our preliminary experiments on a subsection of 1M word Penn Treebank Wall Street Journal corpus (Marcus et al. 1999) (PTB) indicate 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 order to do so, 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, Lamar, and Bienenstock 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 PTB. 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 on the PTB. We extend the experiments to 19 corpora in 15 languages and achieve state-of-the-art MTO scores on 17 of them.

The next section gives a detailed review of related work. Section 3 describes the construction of the substitute vectors and applies various similarity metrics, dimensionality reduction and clustering methods on substitute vectors. Section 4 describes co-occurrence data embedding, the

learning algorithm used in our experiments. Section **??** describes possible usage scenerios of substitute vectors with S-Code on the PTB and determines the best setup. Section 5 applies the best setup to different languages 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 `xxx.xxx.xxx`.

2. 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.

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, 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 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. (2010) employ SVD to enhance similarity between less frequently observed words 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 $w_1 \ldots w_n$ 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 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.

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. Abend et al. (2010) train a prototype-driven model with morphological features by first clustering the high frequency words as the landmarks and then assigning the remaining words to the 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.

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 improves 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, 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 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 Theory and Application

Substitute theory is a special case of Vector Space Models (VSM) (Turney and Pantel 2010) in which meaning of a word is represented by high dimensional substitute vectors. Substitute vectors capture the word–context relation by constructing probability vectors therefore context of each word is represented by its possible substitutes instead of the neighboring words. One problem of VSM is that vectors do not keep the information of identity, ortographic or morphological features of words and there is no standard way of incoorparating these extra features into VSM. Thus instead of focussing on how to incoorporate extra features to the vectors, we dedicate this section to determine the best usage of substitute vectors within the VSM framework by comparing similarity metrics together with dimension reduction and clustering methods on UPOS without using the word identities or any features other than the substitute vectors.

In the following section, we describe the substitute vector theory and computation of substitute vectors using a statistical language model. Section 3.2 gives a detailed comparison of similarity metrics in high dimensional substitute vector space. Section 3.3 analyzes application of possible dimensionality reduction algorithms to our problem. Section 3.4 presents comparison of various clustering techniques and finally Section **??** applies the findings of the previous sections to the PTB.

In section 4 and 4.6[??] we discuss the co-occurance modeling as an alternative to VSM.

3.1 Computation of Substitute Vectors

In this study, we predict the syntactic category 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.

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:

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

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

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

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. Rest of this section details the choice of the data set, the vocabulary and the estimation of substitute probabilities.

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, Rosenfeld, and Paul 1995) (excluding WSJ Section 00). We used SRILM (Stolcke 2002) to build a 4-gram language model with Kneser-Ney discounting. Words that were observed less than 500 times in the LM training data were replaced by UNK tags, which gave us a vocabulary size of 12,672. The first 24,020 tokens of the Penn Treebank Wall Street Journal Section 00 (PTB24K) was used as the test corpus to be induced. Corpus size kept small in order to efficiently compute full distance matrices. Substitution probabilities for 12,672 vocabulary words were computed at each of the 24,020 positions. The perplexity of the 4-gram language model on the test corpus was 55.4 which is quite low due to using a small vocabulary and in-domain data. The treebank uses 45 part-of-speech tags which is the set we used as the gold standard for comparison in our experiments.

3.2 Distance Metrics

We represent each context with a sparse high dimensional probability vector called the substitute vector as described in the previous section. In this section we compare various distance metrics in this high dimensional space with the goal of discovering one that will judge vectors that belong to the same syntactic category similar and vectors that belong to different syntactic categories distant. The distance metrics we have considered are listed in Table 1.

To judge the merit of each distance metric we obtained supervised baseline scores using leave-one-out cross validation and the weighted k-nearest-neighbor algorithm[1] on the gold tags of the PTB24K. The results are listed in Table 2 sorted by score.

prefix indicate a metric applied to the log of the probability vectors. Distance metrics on log probability vectors performed poorly compared to their regular counterparts indicating that differences in low probability words are relatively unimportant and high probability substitutes

1 Neighbors were weighted using 1/distance, $k = 30$ was chosen empirically.

Table 1
Similarity metrics. JS is the Jensen-Shannon divergence and KL2 is a symmetric implementation of Kullback-Leibler divergence. Bold lower case letters represent vectors.

Cosine($\mathbf{p}, \mathbf{q}$)	=	$< \mathbf{p}, \mathbf{q} > /(\|\mathbf{p}\|_2 \|\mathbf{q}\|_2)$
Euclid($\mathbf{p}, \mathbf{q}$)	=	$\|\mathbf{p} - \mathbf{q}\|_2$
Manhattan($\mathbf{p}, \mathbf{q}$)	=	$\|\mathbf{p} - \mathbf{q}\|_1$
Maximum($\mathbf{p}, \mathbf{q}$)	=	$\|\mathbf{p} - \mathbf{q}\|_\infty$
KL2($\mathbf{p}, \mathbf{q}$)	=	$\sum_i p_i ln(p_i/q_i) + q_i ln(q_i/p_i)$
JS($\mathbf{p}, \mathbf{q}$)	=	$\sum_i p_i ln(p_i/m_i) + q_i ln(q_i/m_i)$ where $m_i = 0.5(p_i + q_i)$

Table 2
Supervised baseline scores with different distance metrics. Log-metric indicates that metric applied to the log of the probability vectors.

Metric	Accuracy(%)
KL2	0.6889
Manhattan	0.6865
Jensen	0.6801
Cosine	0.6706
Maximum	0.6663
Euclid	0.6255
lg2-Maximum	0.5361
lg2-Cosine	0.4847
lg2-Euclid	0.4038
lg2-Manhattan	0.3729

determine syntactic category. The surprisingly good result achieved by the simple Maximum metric (which identifies the dimension with the largest difference between two vectors) also support this conclusion. The maximum score of .73% can be taken as a rough upper bound for an unsupervised learner using this space on the PTB24K corpus because .27% of the instances are assigned to the wrong part of speech by the majority of their closest neighbors. We will discuss alternative ways to push this upper bound higher by forcing one-tag-per-word or including word type information together with other features in the rest of this paper.

3.3 Dimensionality Reduction

Using high dimensional vectors is problematic with many learning algorithms because of the computational cost and the curse of dimensionality. In this section we investigate if there is a low dimensional representation of the substitute vectors which still preserve the neighborhood information necessary to learn syntactic categories. We first briefly describe then report experimental results on principal components analysis (PCA), Isomap (Tenenbaum, Silva, and Langford 2000), locally linear embedding (LLE) (Roweis and Saul 2000), and Laplacian eigenmaps (Belkin and Niyogi 2003).

Each dimensionality reduction algorithm tries to preserve certain aspects of the original vectors. PCA is a linear method that minimizes reconstruction error. Isomap tries to preserve distances as measured along a low dimensional submanifold assuming the input vectors were

sampled from the neighborhood of such a manifold. LLE most faithfully preserves the local linear structure of nearby input vectors. Laplacian eigenmaps most faithfully preserve proximity relations, mapping nearby inputs to nearby outputs.

We wanted to see how accuracy (based on the k-nearest-neighbor supervised baseline as in the previous section) changes based on the number of dimensions for each dimensionality reduction algorithm. For computational efficiency, we built 10 substitute vector sets for 24k chunks of contiguous segments extracted from the Wall Street Journal Section of the Penn Treebak (Marcus et al. 1999). We applied each algorithm to each chunk and obtained average accuracy and standard deviation for a given number of dimensions.

For algorithms that require a distance matrix rather than raw input vectors we used the Jensen-Shannon divergence judged best by the experiments of the previous section. For graph based methods we built neighborhood graphs using 100 nearest neighbors. The low dimensional output vectors were compared using the cosine distance metric for the supervised k-nearest-neighbor algorithm. Figure 2 plots supervised baseline accuracy vs. number of dimensions for each algorithm.

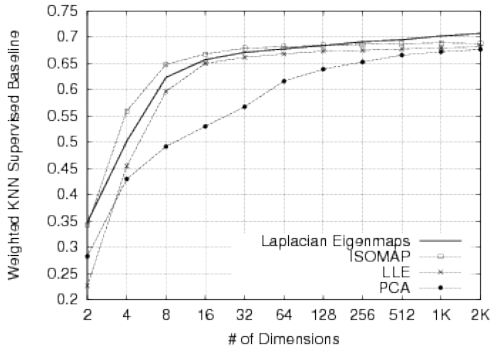

Figure 2
Supervised knn baselines for the four dimensionality reduction algorithms.

The graph based algorithms (Isomap, LLE, and Laplacian eigenmaps) all outperform PCA. They stay within 5% of their peak accuracy with as few as 16 dimensions. In fact Laplacian eigenmaps outperform the baseline with the original 12,672 dimensional vectors (68.95%) when allowed to retain more than about 250 dimensions. Spectral clustering uses the same transformation as the Laplacian eigenmaps algorithm and we compare its performance to other clustering algorithms in the next section.

3.4 Clustering

We compared three clustering algorithms applied to the original substitute vectors using many-to-one accuracy on the PTB24K. Hierarchical agglomerative clustering with complete linkage (HAC) starts with each instance in its own cluster and iteratively combines the two closest groups (measured by their most distant points) at each step (Manning, Raghavan, and Schütze 2008). K-medoids minimizes sum of pairwise distances between each datapoint to the exemplar at the center of its cluster (Kaufman and Rousseeuw 2005). Spectral clustering[2] uses the eigenvalues of the graph Laplacian $L = D^{-1/2}WD^{-1/2}$ to reduce the number of dimensions (similar to Laplacian eigenmaps) and uses simple k-means clustering on the resulting representation (Ng,

2 We used the implementation in (Chen et al. 2011) with a symmetric sparse affinity matrix of 550 nearest neighbors.

Jordan, and Weiss 2002). All three algorithms accept the distance matrix based on the KL2 distance (see Section 3.2) as input.

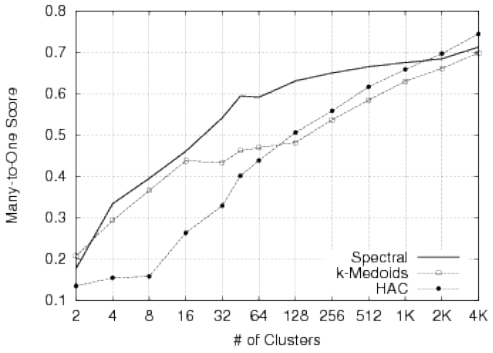

Figure 3
Many-to-one score for three clustering algorithms on the 45-tag 24K word corpus.

Figure 3 plots the many-to-one score versus number of clusters for the three algorithms on the PTB24K. The many-to-one score naturally increases as we approach the one cluster per word limit, however we find the evolution of the curves informative. At the high end (more than 2000 clusters) HAC performs best with its conservative clusters, but its performance degrades fast as we reduce the number of clusters because it cannot reverse the accumulating mistakes. At the low end (less than 16 clusters) k-medoids and spectral have similar performance. However for the region of interest (between 16 to 2000 clusters) spectral clustering is clearly superior with .5841MTO accuracy at 45 clusters.

We noted that the 45 cluster spectral clustering result assigned many more tags to each word than the gold standard. To apply one-tag-per-word restriction we collapse the tag assignment of spectral clustering by re-taging each word with its most frequent tag in the original assignment (we break ties randomly). Collapsing improves the many-to-one accuracy by more than 10% from .5841% to .7082%.

In accord with the findings and results on the PTB24K, we perform spectral clustering on the PTB by using the language model defined in Section **??** together with the Manhattan similarity metric[3] and achieve .58?MTO (.68??VM) accuracy which is comparable with the results on the PTB24.

4. Co-occurrence Modeling

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 in Section 3.4 indicated that using the context information alone (e.g. clustering substitute vectors) without the target word identity and features had limited success. Moreover incoorporating word identities (i.e. one-tag-per-word constraint) even in an ad-hoc manner by collapsing the clustering output significantly improves the MTO accuracy. Thus it is the co-

3 The new language model in Section **??** only calculates the top 100 substitutes and set the ones other than the top 100 to 0 which results in sparse substitute vectors. As a results the new model is computationally more efficient than the one in this section and it is more appropriate for large datasets like the PTB. KL2 is undefined on sparse vectors, therefore we use Manhattan which is the successor of KL2 on both the PTB24K and PTB (see Appendix 1).

occurrence of a target word with a particular type of context that best predicts the syntactic category.

In this section we present theory and applications of substitute vectors as representations of word context within the Co-occurrence Data Embedding (CODE)(Globerson et al. 2007) framework. Section 4.1 reviews the unsupervised methods we used to model co-occurrence statistics: the CODE method and its spherical extension (S-CODE) introduced by (Maron, Lamar, and Bienenstock 2010). 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 using the default parameters defined in Section 4.2. Section 4.3 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 4.4 presents an even simpler model which pairs each word with a random substitute. Section 4.5 replicates the bigram based S-CODE results from (Maron, Lamar, and Bienenstock 2010) as a baseline. 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 (syntagmatic features). Section 4.6 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.

4.1 CODE Theory

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:

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

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 expressed as:

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

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 $d_{x,y}^2$ term pulls neighbors

closer in proportion to the empirical joint probability:

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

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

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

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, Lamar, and Bienenstock 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.

4.2 Experiment Settings

To make a meaningful comparison on the PTB we re-ran all the experiments using the following default settings: (i) each word 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) the s-code dimensions and Z were set to 25 and 0.166, respectively, (v) k-means initialization was improved using (Arthur and Vassilvitskii 2007), and (vi) the number of k-means restarts were increased to 128 to improve clustering and reduce variance.

Section 3.2 shows that low probability substitutes are relatively unimportant thus for computational efficiency only the top 100 substitutes and their unnormalized probabilities were computed for each positions in the PTB[4]. 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. Another advantage of using the top 100 substitutes is we are able to increase the vocabulary size to 78,498 by reducing the vocabulary threshold to 20.

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 paper and the ones we cite from the literature.

4.3 Random partitions

In (Maron, Lamar, and Bienenstock 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. 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

4 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.

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
Bigrams (Sec. 4.5)	.7314 (.0096)	.6558 (.0052)	Substitutes and Features (Sec. 4.6)	.8004 (.0075)	.7160 (.0044)
Partitions (Sec. 4.3)	.7554 (.0055)	.6703 (.0037)			
Substitutes (Sec. 4.4)	.7680 (.0038)	.6822 (.0029)			

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. Using default settings with 64K random partitions 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.

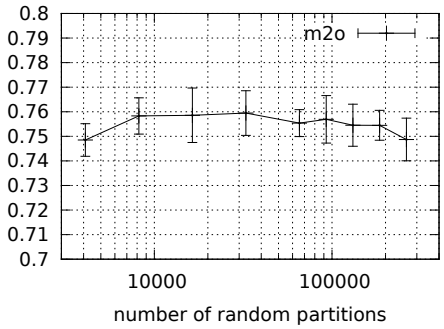

Figure 4
MTO is not sensitive to the number of partitions used to discretize the substitute vector space within our experimental range.

Figure 4 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 5 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 6 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

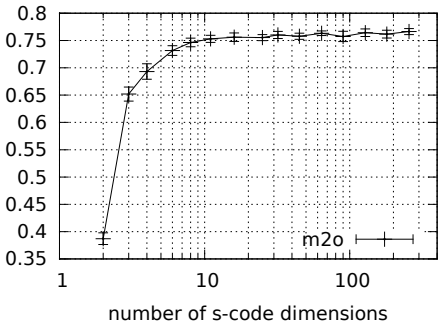

Figure 5
MTO falls sharply for less than 10 S-CODE dimensions, but more than 25 do not help.

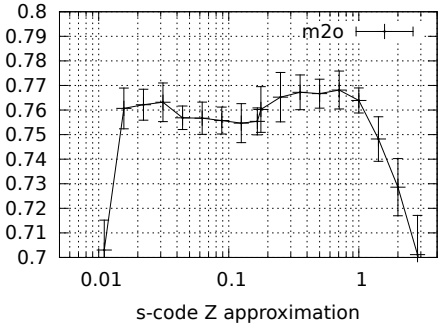

Figure 6
MTO is fairly stable as long as the $\tilde{Z}$ constant is within an order of magnitude of the real Z value.

to converge to the same location. When $\tilde{Z}$ is too high, it prevents meaningful clusters from coalescing.

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 state-of-the-art featureless distributional models.

4.4 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 pre-computed substitute vectors (see Section 3). We ran the random-substitutes algorithm to generate 76 million word (X) – random-substitute (Y) pairs (64 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. Figure 7 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.

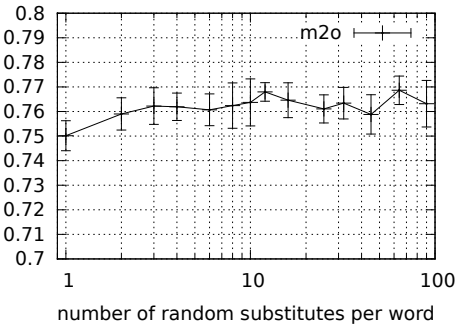

Figure 7
MTO is not sensitive to the number of random substitutes sampled per word token.

4.5 Bigram model

This model uses bigrams as X, Y samples and 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 PTB). If only ϕ_w vectors are clustered without concatenation we found the performance drops significantly to about .62. Using our default settings bigram model achieves .7314 (.0096)MTO and .6558 (.0052)VM accuracies. Both results are significantly lower than the random partition and substitute MTO and VM accuracies.

4.6 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 of other approaches that show similar improvements. This section describes one way to integrate additional features to the random-substitute model.

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

$$\ell(\phi, \psi^{(1)}, \ldots, \psi^{(K)}) = \sum_k w_k \sum_{x, y^{(k)}} \bar{p}(x, y^{(k)}) \log p(x, y^{(k)}) \tag{8}$$

where $Y^{(1)}, \ldots, Y^{(K)}$ are K different variables whose empirical joint distributions with X, $\bar{p}(x, y^{(1)}) \ldots \bar{p}(x, y^{(K)})$, are known. Eq. 8 then represents a set of CODE models $p(x, y^{(k)})$ where each $Y^{(k)}$ has an embedding $\psi_y^{(k)}$ but all models share the same ϕ_x embedding. The weights w_k reflect the relative importance of each $Y^{(k)}$ and all embeddings are mapped to unit-sphere.

We adopt this likelihood function, set all $w_k = 1$, let X represent a word, $Y^{(1)}$ represent a random substitute and $Y^{(2)}, \ldots, Y^{(K)}$ stand for one morphological and various orthographic

features of the word. With this setup, the training procedure needs to change little: instead of sampling a word – random-substitute pair, the word – substitute – features tuple is sampled and input to the gradient ascent algorithm.

One problem with this setup is that unobserved features misguide the gradient search algorithm and lead to a suboptimal convergence point. For example, "**car**" and "**red**" belong to the "Noun" and "Adjective" clusters, respectivly, and neither of them have a morphological feature, thus their morphological features are represented by a null value, "X". However setting the unobserved features of words from different clusters to "X" leads to a false similarity between these words. To solve this problem, during the gradient search we don't perform any pull or push updates on embeddings if the corresponding $y^{(k)}$ is null[5].

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[6]. The program induced 5575 suffix types that are present in a total of 19223 word types.

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 .8004 (.0075) and V-measure to .7160 (.0044). Both these results improve the state-of-the-art in part-of-speech induction significantly as seen in Table 3.

5. Experiments

We perform 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 4 (i.e. the random substitute model) with default settings. To make meaningful comparison with the previous work we only use the training section of MULTEXT-East and

5 x represents a word type therefore it is always observed.

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

CONLL-X languages (Lee, Haghighi, and Barzilay 2010). The number of word types and clusters of each language are summarized in Table 6[7].

5.1 Substitute Vectors and Features

To calculate the top 100 substitutes of each position, we trained a 4-gram language model with the corresponding training corpora of each language as described in Section 4.2. Table 4 presents statistics related to the language model training and testing corpora. For all languages except Serbian, English and Turkish, we trained the language models by using the corresponding Wikipedia dump files[8][Should we talk about tokenizer??].

Table 4
Summary of language model training and test corpora statistics for each language in the test set.

		Language Model			Test set			
	Language	Source	Sentence Count	Word Count	Sentence Count	Word Count	Perplexity (ppl)	Unknown Word
WSJ	English	News	5,187,874	126,170,376	49,208	1,173,766	79.926	0.012
MULTEXT-East	Bulgarian	Wikipedia	1,596,399	32,511,616	6,682	101,173	655.202	.0565
	Czech	Wikipedia	3,059,678	59,698,049	6,752	100,368	1,069.67	.0299
	English	News	5,187,874	126,170,376	6,737	118,424	265.246	.0288
	Estonian	Wikipedia	833,677	14,513,571	6,478	94,898	871.765	.0654
	Hungarian	Wikipedia	3,250,267	66,069,788	6,768	98,426	742.676	.0449
	Romanian	Wikipedia	3,250,267	66,069,788	6,520	118,328	666.855	.1074
	Slovene	Wikipedia	899,329	18,969,846	6,689	112,278	658.711	.0389
	Serbian	Wikipedia	782,278	17,129,679	6,677	108,809	804.962	.0580
CoNLL-X Shared Task	Bulgarian	Wikipedia	1,596,399	32,511,616	12,823	190,217	538.972	.0430
	Czech	Wikipedia	3,059,678	59,698,049	72,703	1,249,408	1,233.95	.0250
	Danish	Wikipedia	1,672,003	35,863,945	5,190	94,386	351.24	.0393
	Dutch	Wikipedia	8,266,922	159,978,524	13,349	195,069	390.818	.0476
	German	Wikipedia	22,454,543	437,777,863	39,216	699,610	680.036	.0487
	Portuguese	Wikipedia	5,706,037	150,099,154	9071	206,678	378.656	.0861
	Slovene	Wikipedia	899,329	18,969,846	1,534	28,750	663.053	.0414
	Spanish	Wikipedia	11,534,351	332,311,650	3,306	89,334	274.418	.0424
	Swedish	Wikipedia	1,953,794	32,004,538	11,042	191,467	1,233.95	.0250
	Turkish	Web	39,595,781	491,195,991	4,997	47,605	868.829	.0508

Serbian shares 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 Serbian test corpus. We chose the Croatian language model since it achieved the lowest perplexity score and unknown word ratio on Serbian test corpus.

To train statistical language model of English, we used Wall Street Journal data (1987-1994) extracted from CSR-III Text (Graff, Rosenfeld, and Paul 1995) (excluding WSJ Section 00) and

7 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.

8 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/`)

for the Turkish language modeling we used the web corpus that was collected from Turkish news and blog sites (Sak, Güngör, and Saraçlar 2008).

Language model training files vary across the languages, in order to reduce the unknown word ratio of resource poor languages and to standardize the process we set the unknown word threshold to 2 for all languages except English. English has relatively low unknown word ratio therefore we set the threshold to 20 instead of 2. [??double check for Multext-East English.??]

We use the same set of orthographic features described in Section 4.6 except we add an "Only-Punctuation" feature to the languages of MULTEXT-East corpora. The "Only-Punctuation" feature is generated when a token consists of punctuations.

Morphological features of each language is extracted by the method described in Section 4.6. Language specific morphological feature statistics are summarized in Table 5.

Table 5
Number of induced suffix parts and word types with these suffix parts after the morfological feature extraction.

	Language	Word types	Suffix Parts
WSJ	English	19223	5575
MULTEXT-East	Bulgarian	4209	609
	Czech	12848	2787
	English	4783	1251
	Estonian	13638	4448
	Hungarian	15995	5423
	Romanian	9445	2064
	Slovene	11834	2093
	Serbian	12476	2722
CoNLL-X Shared Task	Bulgarian	8225	926
	Czech	85673	12443
	Danish	10897	3708
	Dutch	13407	5250
	German	45414	15219
	Portuguese	15721	5033
	Slovene	4781	1257
	Spanish	9316	2648
	Swedish	12725	3897
	Turkish	14227	5651

5.2 Results

For each language we report results: (1) without features (uPos), (2) with orthographic features (uPos+O) and (3) with both orthographic and morphological features (uPos+O+M). In accord with the results of Section 4.4, we use the 25 dimensional sphere with 64 substitutes for all languages. As a baseline model we chose the bi-gram version of S-Code described in Section 4.5 which is a very strong baseline compared to the ones used in (Christodoulopoulos, Goldwater, and Steedman 2011). Table 6 summarizes the MTO and VM scores of our models together with the bi-gram baseline and the best published accuracies on each language corpus.

uPos significantly outperformed the bi-gram baseline in both MTO and VM scores on 14 languages while bi-gram model performed better on Romanian, Serbian and Czech (CONLL-X).

Table 6
The MTO and VM scores on 19 corpora in 15 languages together with the number of types and gold tag-set clusters 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 represents the best MTO and VM accuracies of the corresponding language. 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	Bigram	uPos	uPos+O	uPos+O+M
WSJ	English	49,190	45	.775 / .697‡	.7314 / .6558	.7680 / .6822	? / ?	**.8004 / .7160**
MULTEXT-East	Bulgarian	16,352	12+1	.665 / **.556**⋆	.6732 / .4119	.6883 / .5291	**.7039** / .5496	.6754 / .5246
	Czech	19,115	12+1	.642 / **.539**⋆	.6269 / .4586	.6781 / .4829	.6742 / .4854	**.6977** / .5042
	English	9,773	12+1	.733 / .633⋆	.7690 /. 6131	.8229 / .6610	.8282 / .6719	**.8343 / .6787**
	Estonian	17,845	12+1	.644 / **.533**⋆	.6089 / .4119	.6555 / .4437	**.6634** / .4606	.6526 / .4418
	Hungarian	20,321	11+1	.682 / **.548**⋆	.6181 / .4514	.6914 / .5046	.7052 / .5244	**.7287** / .5444
	Romanian	15,189	12+1	.611 / .523⋆	.6565 / .5202	.6469 / .5012	**.6675 / .5269**	.6488 / .5251
	Slovene	17,871	12+1	.679 / **.567**⋆	.6772 / .5044	.6873 / .4845	**.6892** / .4901	.6833 / .4941
	Serbian	18,095	12+1	.641 / **.510**†	.6267 / .4510	.6240 / .4479	.6303 / .4554	**.6368** / .4650
CoNLL-X Shared Task	Bulgarian	32,439	54	.704 / **.596**†	.6972 / .5532	**.7399** / .5824	.7391 / .5856	.7207 / .5673
	Czech	130,208	12	.701‡ / .484⋆	.6944 / .5036	.6764 / .4867	**.7149 / .5330**	.6903 / .5227
	Danish	18,356	25	**.761**‡ / .591⋆	.6757 / .5290	.7214 / .5559	.7520 / .5927	.7482 / **.5958**
	Dutch	28,393	13	.711‡ / .547	.6703 / .5205	.7014 / .5405	**.7393 / .5980**	.7228 / .5925
	German	72,326	54	.744⋆ / .630†	.7525 / .6285	.7637 / .6314	**.7735 / .6554**	.7529 / .6403
	Portuguese	28,931	22	.785‡ / .639⋆	.7031 / .5617	.7381 / .5770	.7907 / .6317	**.7948 / .6405**
	Slovene	7,128	29	.642⋆ / **.539**†	.6384 / .4976	.6503 / .4925	.6555 / .5036	**.6572** / .5023
	Spanish	16,458	47	**.788**‡ / .632⋆	.7086 / .5844	.7492 / .6083	.7718 / **.6372**	.7627 / .6331
	Swedish	20,057	41	.682 / **.589**†	.6721 / .5558	.6931 / .5654	**.6946** / .5721	.6649 / .5613
	Turkish	17,563	30	.628 / .408⋆	.6069 / .3551	.6228 / .3804	.6348 / .4109	**.6500 / .4246**

uPos+O+M has the state-of-the-art MTO and VM accuracy on the PTB. uPos+O and uPos+O+M achieved the highest MTO scores on all languages of MULTEXT-East corpora while scoring the highest VM accuracies on English and Romanian. On the CoNLL-X languages Pos+O and uPos+M models perform better than the best published MTO score on 7 languages (Czech, Dutch, German, Portuguese, Slovene, Swedish and Turkish). Similarly, these models achieved the top VM scores on 7 languages (Czech, Danish, Dutch, German, Portuguese, Spanish and Turkish). uPos achieves the best published MTO score on CONLL-X Bulgarian corpora.

6. Discussion

Figure 8 is the Hinton diagram of Penn Treebank WSJ showing the relationship between the most frequent tags and clusters from the experiment in Section 4.6. 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).

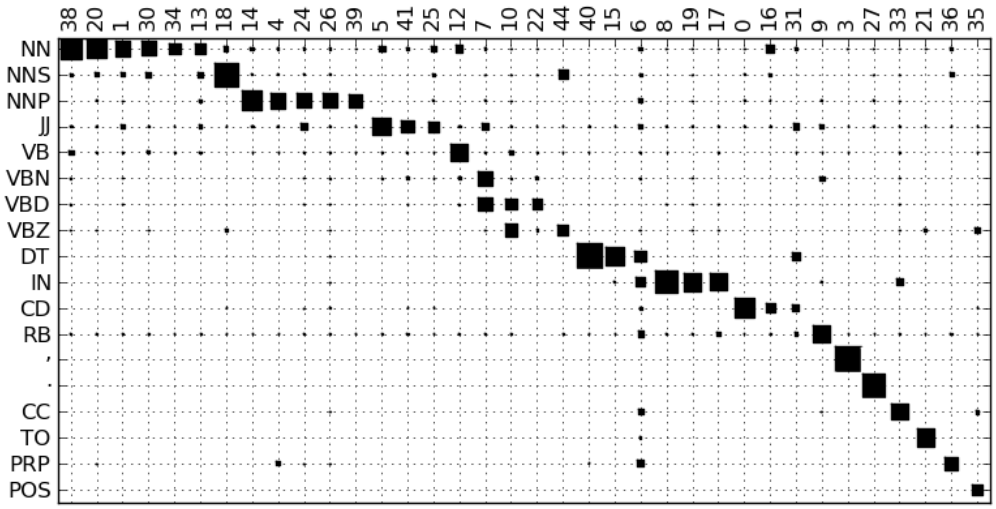

Figure 8
Hinton diagram comparing most frequent tags and clusters.

- 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 same languages as shown on Table 6. 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.

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 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 baseline 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 one million word Penn Treebank Wall Street Journal dataset and Multext-East and CoNLL-X shared task corpora are available at the authors' website at `xxx.xxx.xxx`.

1. Appendix B

Table 7
Supervised baseline scores with different distance metrics on 1M word WSJ Penn Treebank corpus.

Metric	Accuracy(%)
KL2	-
Manhattan	.7353
Jensen	.7317
Cosine	.7240
Maximum	??
Euclid	.6109

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.

Belkin, M. and P. Niyogi. 2003. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396.

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.

Chen, WY, Y. Song, H. Bai, CJ Lin, and EY Chang. 2011. Parallel spectral clustering in distributed systems. *IEEE transactions on pattern analysis and machine intelligence*, 33(3):568–586.

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.

Kaufman, L. and P.J. Rousseeuw. 2005. *Finding groups in data: an introduction to cluster analysis*. Wiley series in probability and mathematical statistics. Applied probability and statistics. Wiley.

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.

Manning, C.D., P. Raghavan, and H. Schütze, 2008. *Introduction to information retrieval*, chapter 17. Cambridge University Press.

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.

Ng, A.Y., M.I. Jordan, and Y. Weiss. 2002. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 2:849–856.

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.

Roweis, S.T. and L.K. Saul. 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323.

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, 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.

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

Tenenbaum, J.B., V. Silva, and J.C. Langford. 2000. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319.

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 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.