Unsupervised Part of Speech Induction Using Paradigmatic Representations of Word Context

Mehmet Ali Yatbaz *
Koç University

Enis Sert
Koç University

Deniz Yuret
Koç University

We investigate paradigmatic representations of word context in the domain of unsupervised part of speech induction. Paradigmatic representations of word context are based on potential substitutes of a word in contrast to syntagmatic representations based on properties of neighboring words. We demonstrate paradigmatic representations within two frameworks: (1) context clustering and (2) co-occurance modeling. In context clustering we cluster word contexts based on the potential substitutes and they reveal a grouping that largely match the traditional part of speech boundaries. In co-occurrence modelling we construct a Euclidean embedding that models the co-occurrence of word types and their contexts. Clustering the points that correspond to word types in the Euclidean embedding gives state-of-the-art results in unsupervised part of speech induction, including 80% many-to-one accuracy on the Penn Treebank and significant improvements on 17 out of 19 corpora in 15 languages.

1. Introduction

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

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

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

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

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

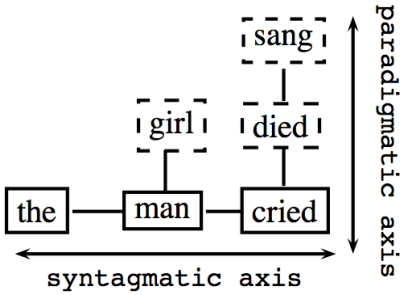

Figure 1
Syntagmatic vs. paradigmatic axes for words in a simple sentence (**?**).

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

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

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

2. Representing Word Context

The two examples below illustrate the advantage of paradigmatic representations in uncovering similarities where no overt similarity that can be captured by a syntagmatic representation exists. The word "board" from the first sentence and the word "council" from the second sentence have no common neighbors except the determiner "the". The paradigmatic representation captures

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

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

the similarity of these words by suggesting the same top substitutes for both (the numbers in parentheses give substitute probabilities):

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

(2) *". . . and hold only 25 % of the seats on the* **council** *."*
council: board (.6591), company (.0795), firm (.0542), bank (.0154), . . .

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

2.1 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 the left and the right context when estimating the probabilities for potential lexical substitutes. For example, in *"He lived in San Francisco suburbs."*, the token *San* would be difficult to guess from the left context but it is almost certain looking at the right context. We define c_w as the $2n-1$ word window centered around the target word position: $w_{-n+1} \dots w_0 \dots w_{n-1}$ ($n = 4$ is the n-gram order). 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 **??**, $P(w|c_w)$ 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 **??** therefore they have been dropped in Equation **??**. 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 **??**. Specifically, at the beginning of the sentence shorter n-gram contexts were used and at the end of the sentence terms beyond the end-of-sentence token were dropped.

To 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 (**?**) (excluding sections of the PTB). We used SRILM (**?**) 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.

The corpus size was 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. Part-of-speech Induction

There are several good reviews of algorithms for unsupervised part of speech induction (**?**; **?**) and models of syntactic category acquisition (**?**).

This work is to be distinguished from supervised part of speech disambiguation systems, which use labeled training data (**?**), unsupervised disambiguation systems, which use a dictionary of possible tags for each word (**?**), or prototype driven systems which use a small set of prototypes for each class (**?**). 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.

3.1 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 (**?**). Work in modeling child syntactic category acquisition has generally followed this clustering approach (**?**; **?**). The second subgroup consists of probabilistic models based on the Hidden Markov Model (HMM) framework (**?**). A third group of algorithms constructs a low dimensional representation of the data that represents the empirical co-occurrence statistics of word types (**?**), which is covered in more detail in Section **??**.

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 (**?**)). 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. (**?**) define context similarity based on the number of common frames bypassing the data sparsity problem but achieve mediocre results. Mintz (**?**) only uses the most frequent 45 frames and Biemann (**?**) clusters the most frequent 10,000 words using contexts formed from the most frequent 150-200 words. Schütze (**?**) and Lamar et al. (**?**) employ SVD to enhance similarity between less frequently observed words and contexts. Lamar et al. (**?**) represent each context by the currently assigned left and right tag (which eliminates data sparsity) and cluster word types using a soft k-means style iterative algorithm. They report the best clustering result to date of .708 many-to-one accuracy on the PTB.

HMMs. The prototypical bitag HMM model maximizes the likelihood of the corpus $w_1 \dots w_n$ expressed as $P(w_1|c_1) \prod_{i=2}^{n} P(w_i|c_i)P(c_i|c_{i-1})$ where w_i are the word tokens and c_i are their (hidden) tags. One problem with such a model is its tendency to distribute probabilities equally

and the resulting inability to model highly skewed word-tag distributions observed in hand-labeled data (**?**). To favor sparse word-tag distributions one can enforce a strict one-tag-per-word solution (**?**; **?**), use sparse priors in a Bayesian setting (**?**; **?**), or use posterior regularization (**?**). Each of these techniques provide significant improvements over the standard HMM model: for example Gao and Johnson (**?**) 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. (**?**) show that the older one-tag-per-word models such as (**?**) 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 (**?**), this is probably not surprising; one-tag-per-word is a fairly good first approximation for induction.

3.2 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. (**?**; **?**; **?**) incorporate similar orthographic features and report improvements of 3, 7, and 10% respectively over the baseline Brown model. Christodoulopoulos et al. (**?**) use prototype based features as described in (**?**) with automatically induced prototypes and report an 8% improvement over the baseline Brown model. Abend et al. (**?**) 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. (**?**) 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.

3.3 Paradigmatic representations

Yatbaz et al. (**?**) explore the paradigmatic representation of word contexts by modeling the co-occurrence of words and their substitutes within the CODE framework. Their experiments on the PTB shows that paradigmatic representation improves the state-of-the-art MTO and V-measure (VM) accuracies of both distributional models and models with additional word features. This paper builds on that preliminary work by (1) exploring clustering of substitute vectors, (2) improving the model for using additional features, and (3) experimenting with additional languages.

Sahlgren (**?**) 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.

Similarly, Schütze and Pedersen (**?**) define the words that frequently co-occur together as the *syntagmatic associates* and words that have similar left and right neighbors as the *paradigmatic parallels*. Turney and Pantel (**?**) give a broad overview of syntagmatic approaches and

their applications within the Vector Space Modeling framework. 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 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 a co-occurrence modeling algorithm.

3.4 Evaluation

We report many-to-one and V-measure scores for our experiments as suggested in (**?**). 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) (**?**) 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 **??** 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.

4. Co-occurrence Modeling

In this section we combine the paradigmatic representation of the word context with the identity and features of the target word for part of speech induction using a co-occurrence modeling framework. Our preliminary experiments in Section **??** indicated that using the context information alone (e.g. clustering substitute vectors) without the target word identity and features had limited success. Moreover incorporating word identities (i.e. the one-tag-per-word constraint) even in an ad-hoc manner by re-tagging the clustering output significantly improves the MTO accuracy. It seems that the co-occurrence of a target word with a particular type of context may be a better predictor of the syntactic category.

Section **??** reviews the unsupervised methods we use to model co-occurrence statistics: the Co-occurrence Data Embedding (CODE) (**?**) method and its spherical extension (S-CODE) introduced by (**?**). Section **??** describes the general experimental settings. The S-CODE algorithm works with discrete inputs. The substitute vectors as described in Section **??** are high dimensional and continuous. We experimented with two approaches to use substitute vectors in a discrete setting. Section **??** 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 **??** presents an even simpler model which pairs each word with random substitutes sampled from the substitute vector. Section **??** replicates the bigram based S-CODE results from (**?**) as a direct comparison of paradigmatic vs syntagmatic representations of word context. 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 **??** explores morphological and orthographic features as additional sources of information and its results improve the state-of-the-art in the field of unsupervised part of speech induction.

4.1 The CODE Model

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:

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

where $d^2_{x,y}$ is the squared distance between the embeddings of x and y, $\bar{p}(x)$ and $\bar{p}(y)$ are empirical probabilities, and $Z = \sum_{x,y} \bar{p}(x)\bar{p}(y)e^{-d^2_{x,y}}$ 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 $d^2_{x,y} = \|\phi_x - \psi_y\|^2$. The log-likelihood of a given embedding $\ell(\phi, \psi)$ can be expressed as:

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

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 $d^2_{x,y}$ term pulls neighbors closer in proportion to the empirical joint probability:

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

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 (**?**) 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.

Table 2
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 (**?**). 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
Lamar et al. (**?**)	.708	-
Brown et al. (**?**)*	.678	.630
Goldwater et al. (**?**)*	.632	.562
Ganchev et al. (**?**)*	.625	.548
Maron et al. (**?**)	.688 (.0016)	-
Bigrams (Sec. **??**)	.7314 (.0096)	.6558 (.0052)
Partitions (Sec. **??**)	.7554 (.0055)	.6703 (.0037)
Substitutes (Sec. **??**)	.7680 (.0038)	.6822 (.0029)

Models with Additional Features	MTO	VM
Clark (**?**)*	.712	.655
Christodoulopoulos et al. (**?**)	.728	.661
Berg-Kirkpatrick et al. (**?**)	.755	-
Christodoulopoulos et al. (**?**)	.761	.688
Blunsom and Cohn (**?**)	.775	.697
Substitutes and Features (Sec. **??**)	.8004 (.0075)	.7160 (.0044)

5. POS Induction for Word Types

5.1 Experimental Settings

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

Section **??** 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 (1,173,766 tokens, 49,206 types) using the FASTSUBS algorithm (**?**)[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. We set the vocabulary threshold to 20 which increases the vocabulary size to 78,498.

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 **??** summarizes all the results reported in this section and the ones we cite from the literature.

5.2 Random Partitions

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 uses stochastic gradient ascent to find the ϕ_x, ψ_y embeddings for word and partition-id in these pairs on a single 25-dimensional sphere. The algorithm cycles through the data until we get approximately 50 million updates. The resulting ϕ_x vectors are clustered using an instance weighted k-means algorithm and the resulting groups are compared to the correct

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

part of speech tags. 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.

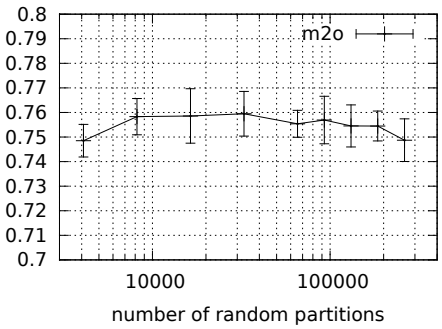

Figure 2
MTO is not sensitive to the number of partitions used to discretize the substitute vector space within our experimental range.

Figure **??** 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.

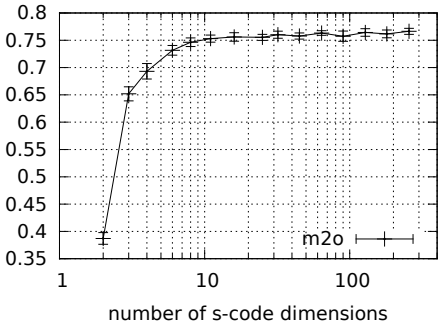

Figure 3
MTO falls sharply for less than 10 S-CODE dimensions, but more than 25 do not help.

Figure **??** 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 **??** 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. **??** dominates the repulsion in Eq. **??** and all points tend 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 distributional models.

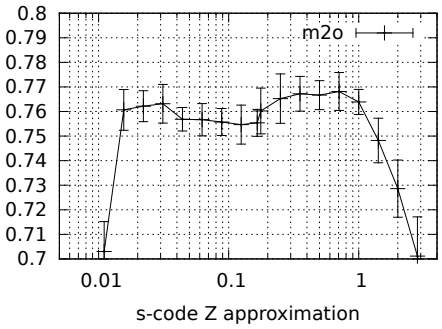

Figure 4
MTO is fairly stable as long as the $\tilde{Z}$ constant is within an order of magnitude of the real Z value.

5.3 Random Substitutes

Another way to use substitute vectors in a discrete setting is simply to sample individual substitute words from them according to the corresponding probabilities. 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 **??**). 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 **??** 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.

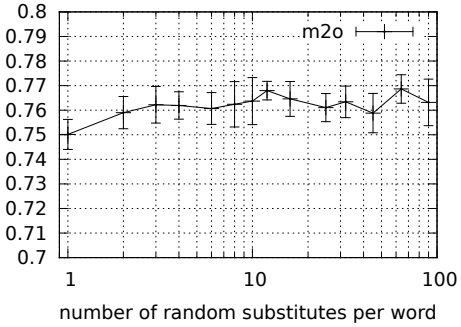

Figure 5
MTO is not sensitive to the number of random substitutes sampled per word token.

5.4 Paradigmatic vs Syntagmatic Representations of Word Context

To get a direct comparison of the paradigmatic and syntagmatic context representations we feed the adjacent word pairs (bigrams) in the corpus into the S-CODE algorithm as X, Y samples (**?**) instead of pairing each word with a paradigmatic representation of its context. At the end

each word w in the vocabulary ends up with two points on the sphere, a ϕ_w point representing the behavior of w as the left word of a bigram and a ψ_w point representing it as the right word. The two vectors for w are concatenated to create a 50-dimensional representation at the end. These 50-dimensional vectors are clustered using the k-means algorithm. Maron et al. (**?**) 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 the 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.

6. POS Induction for Word Tokens

7. Morphological and orthographic features

Clark (**?**) demonstrates that using morphological and orthographic features significantly improves part of speech induction with an HMM based model. Section **??** 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. (**?**) suggest the following likelihood function:

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

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. **??** 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 morphological and orthographic features of the word thus each word is a (K+1)-tuple, $(X, Y^{(1)}, \ldots Y^{(K)})$. With this setup, the training procedure needs to change little: instead of sampling a word – random-substitute pair, the word – random-substitute – features tuple is sampled and input to the gradient ascent algorithm. The gradient search algorithm updates the embeddings according to $p(x, y^{(i)})$ where $i = 1 \ldots k$ and no updates are performed between $y^{(i)}$s since they do not have any co-occurrence statistics and x is the shared variable.

Word tuples might have null values due to the unobserved features. For example, the word "**car**" has no morphological or orthographic features therefore all the elements of the tuple have null value except the word type (X) and the random-substitute ($Y^{(1)}$). We do not perform any pull or push updates on embeddings during the gradient search if the corresponding $y^{(k)}$ is null[2].

The orthographic features we used are similar to the ones in (**?**) with small modifications:

- Initial-Capital: this feature is generated for capitalized words with the exception of sentence initial words.

2 X and $Y^{(1)}$ represents the word type and random-substitute therefore they are always observed.

- 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 (**?**). 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[3]. The program induced 5575 suffix types that are present in a total of 19223 word types.

Using the training settings of the previous section, the addition of morphological and orthographic features increased the many-to-one score of the random-substitute model to .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 **??**.

8. Multilingual Experiments

We performed experiments with a range of languages and three different feature configurations to establish both the robustness of our model across languages and to observe the effects of different features. Following Christodoulopoulos et al. (**?**), in addition to the PTB we extend our experiments to 8 languages from MULTEXT-East (Bulgarian, Czech, English, Estonian, Hungarian, Romanian, Slovene and Serbian) (**?**) and 10 languages from the CoNLL-X shared task (Bulgarian, Czech, Danish, Dutch, German, Portuguese, Slovene, Spanish, Swedish and Turkish) (**?**). For all experiments, we use the best performing model of Section **??** (i.e. the random substitute model) with default settings. To perform meaningful comparisons with the previous work we train and evaluate our models on the training section of MULTEXT-East[4] and CONLL-X languages (**?**).

Section **??** details the language model and feature statistics of each language. Section **??** summarizes the results of our models for all of the languages in our corpora. In the rest of this section we refer to the MULTEXT-East and CONLL-X corpora as the testing corpora and the language model training corpora as the training corpora.

8.1 Substitute Vectors and Features

To calculate the top 100 substitutes of each position, we train a 4-gram language model with the corresponding training corpora of each language as described in Section **??**. Table **??** presents statistics related to the language model training and testing corpora. For all languages except Serbian, English and Turkish, we train the language models by using the corresponding Wikipedia dump files[5].

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

4 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.

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

Table 3
Summary of language model training and test corpora statistics for each language in the test set. Last two columns present the number of induced suffix parts and word types with these suffix parts after the morphological feature extraction.

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

Serbian shares a common basis with Croatian and Bosnian therefore we trained 3 different language models using Wikipedia dump files of Serbian together with these two languages and measured the perplexities on the MULTEXT-East Serbian corpus. We chose the Croatian language model since it achieved the lowest perplexity score and unknown word ratio on the MULTEXT-East Serbian corpus.

To train the statistical language model of English, we use Wall Street Journal data (1987-1994) extracted from CSR-III Text (**?**) (excluding sections of the PTB) and for the Turkish language modeling we use the web corpus collected from Turkish news and blog sites (**?**).

Language model training files vary across the languages in terms of quality and quantity. In order to reduce the unknown word ratio of resource poor languages and to standardize the process we set the vocabulary threshold to 2 for all languages except English. English has relatively low unknown word ratio therefore we set the threshold to 20 instead of 2.

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

Morphological features are extracted by the method described in Section **??** using the training sections of each language in MULTEXT-East and CoNLL-X corpora[6]. Language specific morphological feature statistics are summarized in Table **??**.

Table 4
The MTO and VM scores on 19 corpora in 15 languages together with the number of types and tags in gold–set which equals to number of induced clusters in all languages. Best published results are from ‡(**?**), *(**?**) and †(**?**). Bold results represent 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	Syntagmatic Bigram	DIS	DIS+O	DIS+O+M
WSJ	English	49,190	45	.775 / .697‡	.7314 / .6558	.7680 / .6822	.7830 / .7026	**.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	11+1	.644 / **.533***	.6089 / .4119	.6555 / .4437	**.6634** / .4606	.6526 / .4418
	Hungarian	20,321	12+1	.682 / **.548***	.6181 / .4514	.6914 / .5046	.7052 / .5244	**.7287** / .5444
	Romanian	15,189	14+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**

8.2 Results

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

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

DIS significantly outperforms the syntagmatic bigram baseline in both MTO and VM scores on 14 languages. DIS+O+M has the state-of-the-art MTO and VM accuracy on the PTB. DIS+O and DIS+O+M achieve the highest MTO scores on all languages of MULTEXT-East corpora

6 We don't use the language model corpora to extract morphological features.

while scoring the highest VM accuracies on English and Romanian. On the CoNLL-X languages our models perform better than the best published MTO or VM accuracies on 10 languages.

9. Discussion

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

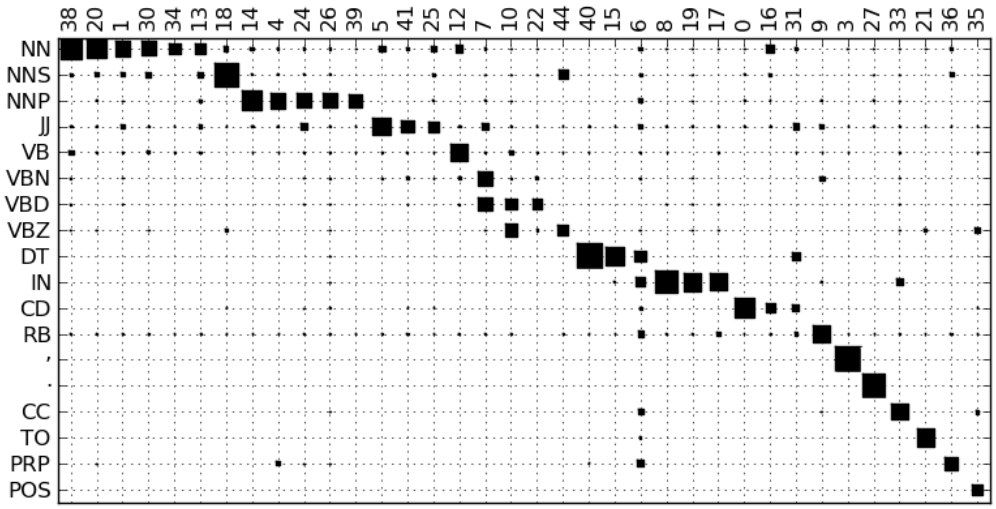

Figure 6
Hinton diagram comparing most frequent tags and clusters. Area of each square is proportional to the joint probability of the given tag and cluster.

Figure **??** is the Hinton diagram of the PTB showing the relationship between the most frequent tags and clusters from the experiment in Section **??**. 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 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 **??**. On CONLL-X languages the effect of completeness errors is less noticeable since all languages except Czech and Dutch have fine grained tag-sets.

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

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

10. Contributions

Our main contributions can be summarized as follows:

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