A Paradigmatic Model for Learning Syntactic Categories

First Author
Affiliation / Address line 1
Affiliation / Address line 2
Affiliation / Address line 3
email@domain

Second Author
Affiliation / Address line 1
Affiliation / Address line 2
Affiliation / Address line 3
email@domain

Third Author
Affiliation / Address line 1
Affiliation / Address line 2
Affiliation / Address line 3
email@domain

Abstract

We introduce a paradigmatic representation of word context and demonstrate its utility in learning syntactic categories. Unlike the typical syntagmatic representations of word context which consist of properties of neighboring words, our paradigmatic representation consists of substitute vectors: possible substitutes of the target word and their probabilities. When word contexts are clustered based on their substitute vectors they reveal a grouping that largely match the traditional part of speech boundaries with a many-to-one accuracy of 70.82% on a 45-tag 24K word test corpus.

1 Introduction

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

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.

Computational models of syntactic category acquisition in the literature mainly rely on distributional analysis: Items that share the same distribution (i.e. that occur in the same context) are grouped into the same category. The definition of "the same context" vary across studies. Algorithms based on the Hidden Markov Model use class based n-grams to specify context (Brown et al., 1992), others use a frame of neighboring words around the target word (Schütze, 1995). Our main contribution in this study is to introduce paradigmatic features, i.e. features based on potential substitutes of the target word, to represent word context.

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.

paradi

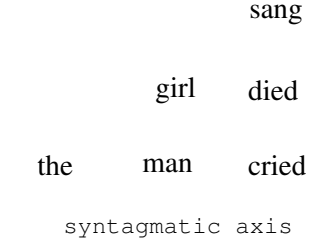

Figure 1: Syntagmatic vs. paradigmatic axes for words in a simple sentence (Chandler, 2007).

Both syntagmatic and paradigmatic relations of a

word can be used to represent its context. In the syntagmatic case the context is represented by a selection of neighboring words, in the paradigmatic case it is represented by a set of possible substitutes. In previous studies of syntactic category learning the context representation has been primarily syntagmatic, either implicit in the class based n-grams of the standard Hidden Markov Model, or explicit in the construction and clustering of left and right neighbors.

In this study we explore a paradigmatic representation of the context of a word in syntactic category acquisition. Specifically, the context of a word is represented by a list of its possible substitutes and their probabilities, which we call the *substitute vector*. Note that the substitute vector is a function of the context only, not the target word. Thus in effect we are clustering contexts, not words. When word contexts are clustered based on their substitute vectors they reveal a grouping that largely match the traditional part of speech boundaries (70.82% many-to-one score using a 45-tag 24K word test corpus).

Section 2 gives a detailed review of related work. The construction of the substitute vectors is described in Section 3. To find out how to best make use of this new paradigmatic representation, we explore different distance metrics (Section 4), dimensionality reduction methods (Section 5), and clustering algorithms (Section 6) for substitute vectors. We note that close to 95% of the word occurrences in human labeled data are tagged with their most frequent part of speech (Lee et al., 2010), making one-tag-per-word a fairly good first approximation. Even ambicategory words generally have fairly skewed part of speech distributions. Section 7 looks at ways to increase the sparsity of our solutions and demonstrates significant improvements using the one-tag-per-word assumption and similarity metrics that introduce sparsity. Section 8 discusses the results and Section 9 summarizes our contributions.

2 Related Work

There are several good reviews of algorithms for unsupervised part-of-speech induction (Christodoulopoulos et al., 2010; Gao and Johnson, 2008) and models of syntactic category acquisition (Ambridge and Lieven, 2011). In this review we focus on (i) the information captured by the inputs, and (ii) the constraints applied to the outputs of the algorithms.

This work is to be distinguished from supervised part-of-speech disambiguation systems, which use labeled training data (Church, 1988), unsupervised disambiguation systems, which use a dictionary of possible tags for each word (Merialdo, 1994), or prototype driven systems which use a small set of prototypes for each class (Haghighi and Klein, 2006). The problem of induction is important for studying under-resourced languages that lack labeled corpora and high quality dictionaries. It is also essential in modeling child language acquisition because every child manages to induce syntactic categories without access to labeled sentences, labeled prototypes, or dictionary constraints.

Clustering vs. HMMs: Models of unsupervised induction of part-of-speech categories fall into two broad groups. The first group uses algorithms that cluster word types based on their context statistics (Schütze, 1995). Work in modeling child syntactic category acquisition has generally followed this clustering approach (Redington et al., 1998; Mintz, 2003). The second group consists of probabilistic models based on the Hidden Markov Model (HMM) framework (Brown et al., 1992).

Clustering and data sparsity: 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) defines context similarity based on the number of common frames bypassing the data sparsity problem but achieves 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; Lamar et al., 2010b) employ SVD to enhance similarity between less frequently observed words and contexts. (Lamar et al., 2010a) represents each context by the currently assigned left and right tag (which eliminates data sparsity) and clusters word types using a soft k-means style iterative algorithm. They report the best clustering result to date of 70.8% many-to-one accuracy on a 45-tag 1M word corpus.

HMMs and distribution sparsity: The prototypical bitag HMM model maximizes the likelihood of the corpus $w_1 \dots w_n$ expressed as $P(w_1|c_1) \prod_{i=2}^{n} P(w_i|c_i)P(c_i|c_{i-1})$ where w_i are the word tokens and c_i are their (hidden) tags. One problem with such a model is its tendency to distribute probabilities equally and the resulting inability to model highly skewed word-tag distributions observed in hand-labeled data (Johnson, 2007). To favor sparse word-tag distributions one can enforce a strict one-tag-per-word solution (Brown et al., 1992; Clark, 2003), use sparse priors in a Bayesian setting (Goldwater and Griffiths, 2007; Johnson, 2007), or use posterior regularization (Ganchev et al., 2010). Each of these techniques provide significant improvements over the standard HMM model: for example (Gao and Johnson, 2008) shows that sparse priors can gain from 4% (62% to 66% with a 1M word corpus) to 30% (28% to 58% with a 24K word corpus) in cross-validated many-to-one accuracy. However (Christodoulopoulos et al., 2010) shows that the older one-tag-per-word models such as (Brown et al., 1992) outperform the more sophisticated sparse prior and posterior regularization methods both in speed and accuracy (the Brown model gets 68% many-to-one accuracy with a 1M word corpus). Given that close to 95% of the word occurrences in human labeled data are tagged with their most frequent part of speech (Lee et al., 2010), this is probably not surprising; one-tag-per-word is a fairly good first approximation for induction.

Poverty of features: Another problem with the plain HMM model is the poverty of its input features. Of the syntactic, semantic, and morphological information linguists claim underlie syntactic categories, bitag HMMs only represent limited syntactic information in their class based n-grams. (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. Semantic information is incorporated to some extent by some clustering based models and our paradigmatic model. To our knowledge, nobody has yet tried to incorporate phonological or prosodic features in a computational model for syntactic category acquisition.

Evaluation: It is natural to question the merit of evaluating unsupervised results by comparing them to gold standard tags. For example (Freudenthal et al., 2005) argue that a verb category (such as VB in Penn Treebank) that contains verbs that can and cannot be used in certain constructions (e.g. the imperative) and verbs that can be used as both auxiliaries and main verbs (e.g., *do, have*) does not in fact constitute a set of items that could be substituted for one another in particular sentences. Such a category fails 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?"*). They suggest evaluating models by incorporating a production component that allows the model's output to be compared to speech produced by children exposed to the same input. (Frank et al., 2009), motivated by the lack of gold standards for many novel languages, suggest comparing a system's clusters to a set of clusters created from *substitutable frames*. They create frames using two words appearing in the corpus with exactly one word between and calculate precision, recall, and F-score of the system's clustering. Statistical parsers or factored machine learning systems could also be sources of extrinsic evaluation for induced syntactic categories. We hope such extrinsic evaluation will be more widespread in the future but nevertheless use the 45-tag Penn Treebank gold standard to evaluate the current work.

Our work: Our work is more closely related to clustering based methods with two important differences: (i) we represent word contexts not by an ad-hoc selection of syntagmatic features based on

a single word neighborhood but by substitute vectors defined by a statistical language model representing paradigmatic features of a larger window, (ii) we cluster word tokens (or rather their contexts) instead of word types, which means not being constrained by the one-tag-per-word assumption from the beginning. The features in our model incorporate semantic as well as syntactic information from the context as illustrated by the examples in Section 3. Our best results on the 45-tag 24K corpus given in Section 7 (70.82% many-to-one) compare favorably with best previously reported results on the same corpus (58.2% in (Gao and Johnson, 2008)). This significant improvement is due to the additional information obtained from the statistical language model which also comes at a great computational cost. Developing faster algorithms to find high probability substitutes is in our current research agenda.

3 Substitute Vectors

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

The first 24,020 tokens of the Penn Treebank (Marcus et al., 1999) Wall Street Journal Section 00 was used as the test corpus. The treebank uses 45 part-of-speech tags which is the set we used as the gold standard for comparison in our experiments. To compute substitute probabilities we trained a language model using approximately 126 million tokens of Wall Street Journal data (1987-1994) extracted from CSR-III Text (Graff et al., 1995) (we excluded the test corpus). We used SRILM (Stolcke, 2002) to build a 4-gram language model with Kneser-Ney discounting. Words that were observed less than 500 times in the LM training data were replaced by UNK tags, which gave us a vocabulary size of 12,672. The perplexity of the 4-gram language model on the test corpus was 55.4 which is quite low due to using in-domain data and a small vocabulary.

We used both left and right neigbors to estimate the probabilities for potential substitutes of word w in a context c_w. 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 LM order). The probability of a substitute word in a given context can be estimated as:

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

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

After computing the unnormalized probability of each of the 12,672 vocabulary words at each of the 24,020 positions in the test corpus based on Equation 3, 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. Computation of substitute vectors is computationally expensive and we are investigating algorithmic methods that will provide the most likely substitutes for each position without going through the whole vocabulary. However, the focus of this work is to first demonstrate the usefulness of the substitute vector representation in learning syntactic categories.

The high probability substitutes reflect both semantic and syntactic features of the context as seen in the examples given in Table 1. 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.

These examples illustrate 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

word	*substitute*	*probability*
the	its	.9011
	the	.0978
	a	.0007
board	company	.5573
	firm	.2219
	bank	.0727

Table 1: Example substitutes for "the" and "board" in the sentence *"Pierre Vinken, 61 years old, will join the board as a nonexecutive director Nov. 29."*

standard, (ii) words that are generally not substitutable like "board" and "banana" are placed in the same category (NN). 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 most previous work and evaluate our results by comparing them to gold standard part-of-speech tags.

4 Distance Metric

We represent each context with a 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 2.

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 $m_i = (p_i + q_i)/2$

Table 2: Similarity metrics. JS is the Jensen-Shannon divergence and KL2 is a symmetric implementation of Kullback-Leibler divergence.

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 test corpus. The results are listed in Table 3 sorted by score.

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

Table 3: Supervised baseline scores with different distance metrics. Log-metric indicates that metric applied to the log of the probability vectors.

The entries with the log- 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 differences in low probability words are relatively unimportant and high probability substitutes 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 69% can be taken as a rough upper bound for an unsupervised learner using this space on the 45-tag 24K test corpus because 31% of the instances are assigned to the wrong part of speech by the majority of their closest neighbors. We will discuss ways to push this upper bound higher by including other features in Section 7.

5 Dimensionality Reduction

Using high dimensional vectors is problematic with many learning algorithms because of 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 et al., 2000), locally linear embedding (LLE) (Roweis and

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

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) changed based on the number of dimensions for each dimensionality reduction algorithm. All other parameters were set empirically to values that gave reasonable results: For algorithms that require a distance matrix rather than raw input vectors we used the KL2 distance 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.

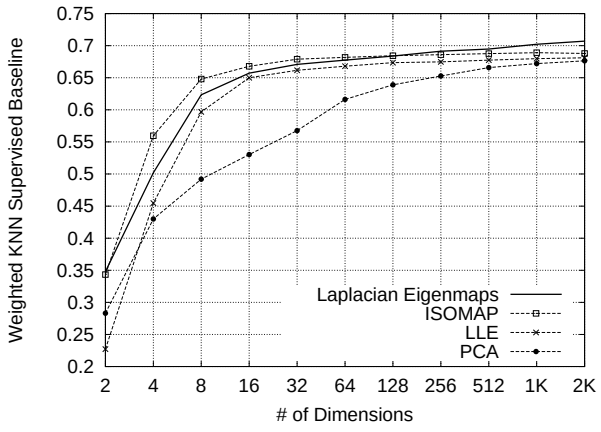

Figure 2: Supervised knn baselines for the four dimensionality reduction algorithms.

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.

6 Clustering

We compared three clustering algorithms applied to the original substitute vectors using many-to-one accuracy on the 45-tag 24K word test corpus. 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 et al., 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 et al., 2002). All three algorithms accept the distance matrix based on the KL2 distance (see Section 4) as input.

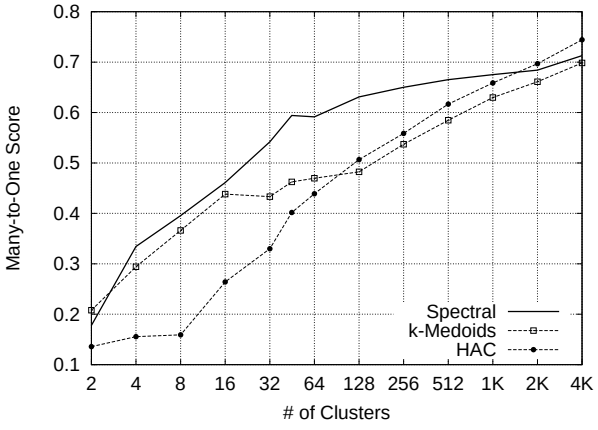

Figure 3: Many-to-one score for three clustering algorithms on the 45-tag 24K word corpus.

Figure 3 plots the many-to-one score versus number of clusters for the three algorithms on the 45-

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

tag 24K word test corpus. 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 59.41% many-to-one accuracy at 45 clusters.

7 Increasing Sparsity

We noted that the 45 cluster spectral clustering result described in the previous section assigned many more tags to each word than the gold standard. To quantify the difference we used a measure called tag perplexity defined as follows:

$$2^{\frac{1}{N}\sum_{i=1}^{N} -\log_2 p(t_i|w_i)}$$

Here N is the number of words in the corpus, w_i is the i'th word, t_i is its assigned cluster or tag, and $p(t_i|w_i)$ is the fraction of times word w_i has been assigned t_i. A model which had to choose from q equally likely tags for each word would have a tag perplexity of q. The tag perplexity of the gold standard 45-tag 24K word test corpus is 1.09, whereas the tag perplexity of the spectral clustering result is 2.76.

We experimented with two methods for reducing the number of tags assigned to each word: collapsing and word penalties. Collapsing enforces the one-tag-per-word constraint by re-tagging the corpus, whereas word penalties encourage it by increasing the distance between instances with different target words.

To collapse a given tag assignment for a corpus, we re-tag each word with its most frequent tag in the original assignment (we break ties randomly). This forcefully reduces the tag perplexity to 1 and removes any ambiguity. Collapsing improves the many-to-one accuracy by more than 10% from 59.41% to 70.82%.

Interestingly when we try to enforce the one-tag-per-word restriction before clustering (by giving the average substitute vector for each word type to spectral clustering) the results get worse (58.02% many-to-one accuracy). The information in individual instances seems to be necessary for good clusters to arise.

Word penalties include information about the target word in the distance metric. The substitute vectors and the KL2 distance metric based on them carry no information about the target word, only its context. We used the following distance metric which increases the distance between instances with different target words:

$$D(i,j) = KL2(s_i, s_j) + \delta I(w_i \neq w_j)$$

Here s_i is the substitute vector and w_i is the target word for the i'th position, δ is the regularization parameter, and I is the indicator function that gives 1 if the two words are different and 0 if they are the same. Increasing the δ decreases the tag perplexity, but the accuracy change is non-monotonic. At $\delta = 1$ we obtain a tag perplexity of 1.91 and the many-to-one accuracy increases from 59.41% to 64.35%. This demonstrates that we can significantly increase the accuracy by including more information on the target word without employing the full one-tag-per-word constraint.

8 Discussion

Figure 4 is the Hinton diagram showing the relationship between the most frequent tags and clusters found by the collapsed algorithm (70.82% many-to-one accuracy). In this section we present a qualitative comparison of gold standard tags and discovered clusters.

Nouns and adjectives: Most nouns (NN*) are split between the clusters represented by the first seven columns of the Hinton graph, but not in the way Penn Treebank splits them. For example cluster 27 brings together titles like *Mr.*, *Mrs.*, *Dr.* etc. which does not exist as a separate class in the gold tags. Cluster 29 is the largest adjective (JJ) cluster, however it also has noun members probably due to the difficulty of separating noun-noun compounds and adjective modification.

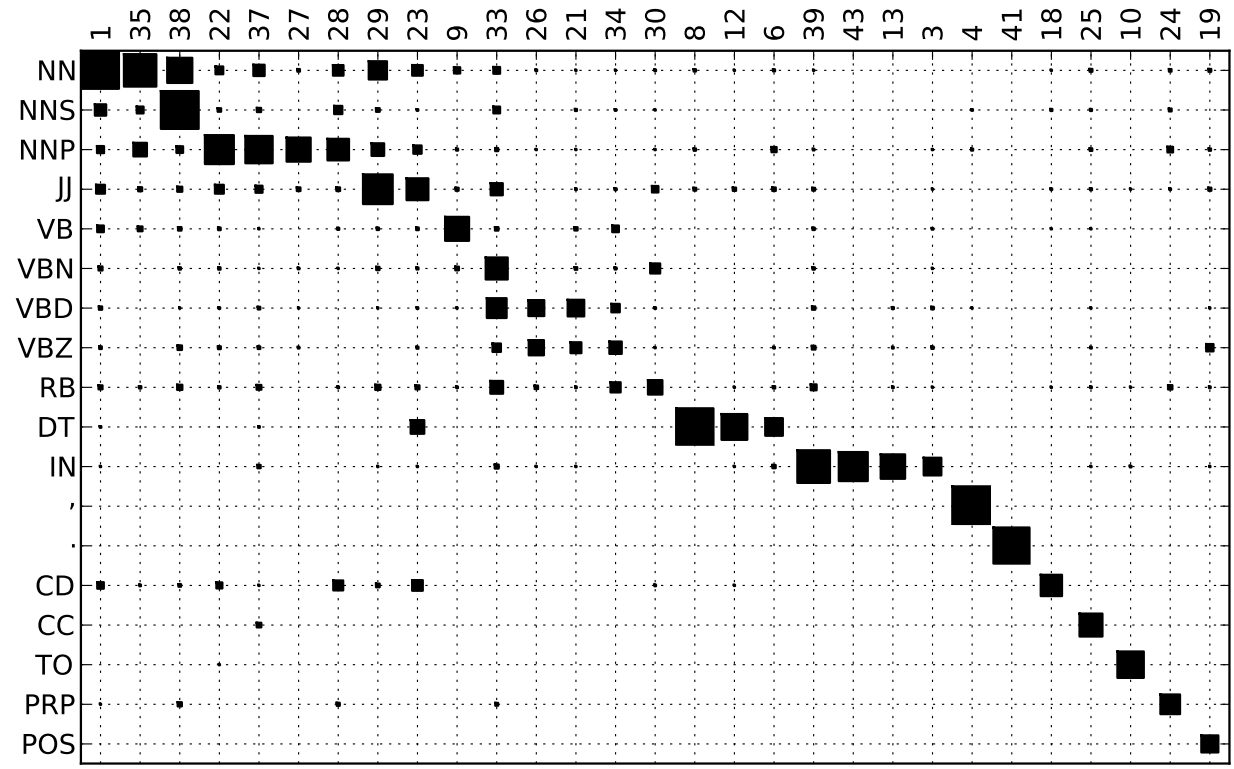

Figure 4: Hinton diagram of the most frequent tags (rows) and clusters (columns). Area of each square is proportional to the joint probability of the given tag and cluster.

Verbs and adverbs: Clusters 9 and 33 contain general verbs (VB*), but the verbs "be" (26), "say" (21), and "have" (34) have been split into their own clusters indicated in parantheses, presumably because they are not generally substitutable with the rest. Adverb (RB) is an amorphous class and the algorithm seems to have difficulty isolating it in a cluster.

Determiners and prepositions: We see a fairly clean separation of determiners (DT) and prepositions (IN) from other parts of speech, although each has been subdivided into further groups by the algorithm. For example cluster 39 contains general prepositions but "of" (43), "in" (13), and "for" (3) are split into their own clusters. Determiners "the" (8), "a" (12), and capitalized "The"/"A" (6) are also split into their own clusters.

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

9 Contributions

We introduced a paradigmatic representation for word context which contains possible substitutes for the target word and their probabilities. The substitute probabilities were estimated using a standard 4-gram language model with Kneser-Ney smoothing. We investigated distance metrics, dimensionality reduction techniques and clustering algorithms appropriate for substitute probability vectors and evaluated them using a supervised k-nearest-neighbor baseline and many-to-one accuracy relative to a 45-tag 24K word test corpus. We found that the KL2 distance metric (a symmetric version of Kullback-Leibler divergence), dimensionality reduction using Laplacian eigenmaps, and spectral clustering work well with substitute vectors. Spectral clustering of substitute vectors reveal a grouping that largely match traditional part of speech boundaries (59.41% many-to-one accuracy) which further improve when the one-tag-per-word constraint is imposed (70.82% many-to-one accuracy). These results compare favorably to previous results published for the 45-tag

24K word test corpus we have used.

References

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

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

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

C. Biemann. 2006. Unsupervised part-of-speech tagging employing efficient graph clustering. In *Proceedings of the 21st International Conference on computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 7–12. Association for Computational Linguistics.

Phil Blunsom and Trevor Cohn. 2011. A hierarchical pitman-yor process hmm for unsupervised part of speech induction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 865–874, Portland, Oregon, USA, June. Association for Computational Linguistics.

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

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

WY Chen, 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.

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

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

Alexander Clark. 2003. Combining distributional and morphological information for part of speech induction. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics - Volume 1*, EACL '03, pages 59–66, Stroudsburg, PA, USA. Association for Computational Linguistics.

S. Frank, S. Goldwater, and F. Keller. 2009. Evaluating models of syntactic category acquisition without using a gold standard. In *Proc. 31st Annual Conf. of the Cognitive Science Society*, pages 2576–2581.

D. Freudenthal, J.M. Pine, and F. Gobet. 2005. On the resolution of ambiguities in the extraction of syntactic categories through chunking. *Cognitive Systems Research*, 6(1):17–25.

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

Jianfeng Gao and Mark Johnson. 2008. A comparison of bayesian estimators for unsupervised hidden markov model pos taggers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 344–352, Stroudsburg, PA, USA. Association for Computational Linguistics.

Sharon Goldwater and Tom Griffiths. 2007. A fully bayesian approach to unsupervised part-of-speech tagging. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 744–751, Prague, Czech Republic, June. Association for Computational Linguistics.

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

Aria Haghighi and Dan Klein. 2006. Prototype-driven learning for sequence models. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, HLT-NAACL '06, pages 320–327, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mark Johnson. 2007. Why doesn't EM find good HMM POS-taggers? In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 296–305, Prague, Czech Republic, June. Association for Computational Linguistics.

L. Kaufman 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.

Michael Lamar, Yariv Maron, and Elie Bienenstock. 2010a. Latent-descriptor clustering for unsupervised

pos induction. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 799–809, Stroudsburg, PA, USA. Association for Computational Linguistics.

Michael Lamar, Yariv Maron, Mark Johnson, and Elie Bienenstock. 2010b. Svd and clustering for unsupervised pos tagging. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 215–219, Uppsala, Sweden, July. Association for Computational Linguistics.

Yoong Keok Lee, Aria Haghighi, and Regina Barzilay. 2010. Simple type-level unsupervised pos tagging. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 853–861, Stroudsburg, PA, USA. Association for Computational Linguistics.

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

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

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

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

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

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

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

Hinrich Schütze. 1995. Distributional part-of-speech tagging. In *Proceedings of the seventh conference on European chapter of the Association for Computational Linguistics*, EACL '95, pages 141–148, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

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

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