Unsupervised Instance-Based Part of Speech Induction Using Probable Substitutes

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

Abstract

We develop an instance (token) based extension of the state of the art word (type) based part-of-speech induction system introduced in (Yatbaz et al., 2012). Each word instance is represented by a feature vector that combines information from the target word and probable substitutes sampled from an n-gram model representing its context. Modeling ambiguity using an instance based model does not lead to significant gains in overall accuracy in part-of-speech tagging because most words in running text are used in their most frequent class (e.g. 93.69% in the Penn Treebank). However it is important to model ambiguity because most frequent words are ambiguous and not modeling them correctly may negatively affect upstream tasks. Our main contribution is to show that an instance based model can achieve significantly higher accuracy on ambiguous words at the cost of a slight degradation on unambiguous ones, maintaining a comparable overall accuracy. On the Penn Treebank, the overall many-to-one accuracy of the system is within 1% of the state-of-the-art (80%), while on highly ambiguous words it is up to 70% better. On multilingual experiments our results are significantly better than or comparable to the best published word or instance based systems on 15 out of 19 corpora in 15 languages. The vector representations for words used in our system are available for download for further experiments.

1 Introduction

Unsupervised part-of-speech (POS) induction aims to classify words into syntactic categories using unlabeled, plain text input. 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 (Ambridge and Lieven, 2011). Categories induced from data may point to shortcomings or inconsistencies of hand-labeled categories as discussed in Section 4. Finally, the induced categories or the vector representations generated by the induction algorithms may improve natural language processing systems when used as additional features.

Word-based POS induction systems classify different instances of a word in a single category (which we will refer to as the *one-tag-per-word assumption*). Instance-based systems classify each occurence of a word separately and can handle ambiguous words.

Examples of word-based systems include ones that represent each word with the vector of neighboring words (context vectors) and cluster them (Schütze, 1995; Lamar et al., 2010b; Lamar et al., 2010a), use a prototypical bi-tag HMM that assigns each word to a latent class (Brown et al., 1992; Clark, 2003), restrict a HMM based Pitman-Yor process to perform one-tag-per-word inference (Blunsom and Cohn, 2011), define a word-based Bayesian multinomial mixture model (Christodoulopoulos et al., 2011), or construct word vector representations based on co-occurrences with contextual features (Yatbaz et al., 2012).

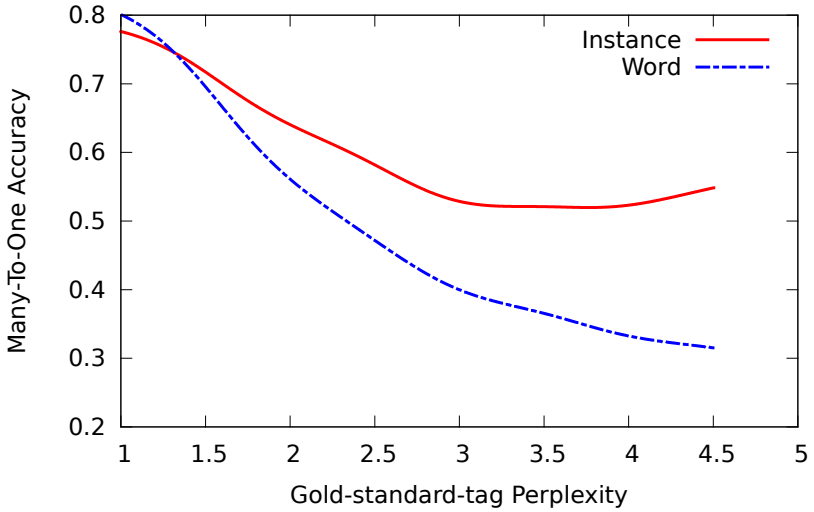

Figure 1: The accuracy comparison of word and instance based part-of-speech induction models as a function of target word ambiguity (as measured by gold-standard-tag perplexity described in Section 3.3) on the Penn Treebank.

The obvious limitation of the one-tag-per-word assumption is that instances of ambiguous words that have more than one POS role are grouped into the same class. For example, the word *offer* is tagged as NN(399), VB(105) and VBP(34)[1] in its 538 occurrences in the human labeled Wall Street Journal (WSJ) Section of the Penn Treebank (PTB) corpus (Marcus et al., 1999). If all instances of *offer* are assigned to the most frequent tag NN, 36% (139/538) will be erroneously labeled. In spite of this shortcoming, word-based POS induction systems generally do well because the one-tag-per-word assumption is mostly accurate: 93.69% of the word occurrences are tagged with their most frequent POS tag in the PTB (Toutanova et al., 2003).

In order to handle ambiguous words, models without a strict one-tag-per-word assumption need to group word *instances* into clusters according to their contexts. Some of these instance-based models bias words to have few tags using sparse priors in a Bayesian setting (Goldwater and Griffiths, 2007; Johnson, 2007; Gao and Johnson, 2008), or posterior regularization (Ganchev et al., 2010). Schütze (1993) represents the context of a word instance by concatenating context vectors of its left and right neighboring words, and clusters word instances. Berg-Kirkpatrick et al. (2010) use an EM algorithm where they replace the multinomial components with miniature logistic regressions and achieve the highest instance-based accuracy on PTB. Christodoulopoulos et al. (2010) select prototypes of each cluster from the output of Brown (1992) and feed them to a HMM model that can handle prototypes as features (Haghighi and Klein, 2006). However none of these models achieve results comparable to the best word-based systems.

In this work, we show that one can build an instance-based system that can perform significantly better on highly ambiguous words (see Figure 1) and yet is competitive with word-based systems in overall accuracy.

We follow the state of the art word-based system (Yatbaz et al., 2012) and use probable substitutes of a word instance as its contextual features. The following examples illustrate how such paradigmatic (substitute based) contextual features can capture the similarity between two contexts where a syntagmatic (neighbor based) representation would fail:

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

[1]NN, VB and VBP are three POS tags from the Penn Treebank corpus and they correspond to singular noun, verb in base form and non-3^{rd}person singular verb in present tense, respectively. The numbers in parentheses are the frequencies.

(2) *"... Joseph Corr was succeeded by Frank Lorenzo,* **chief** *of parent Texas Air."*
chief → chairman (.9945), president (.0031), directors (.0012) ...

Each sentence has the target words marked in bold (**director** and **chief**) and their likely substitutes listed with probabilities[2] in parentheses. Note that the two contexts have no words in common, therefore syntagmatic (neighbor based) contextual features will fail to capture their similarity. However, paradigmatic features such as the top substitutes "chairman", "directors", etc. clearly indicate the similarity and help place these two instances into the same category.

Following (Globerson et al., 2007), we embed words and their contextual, orthographic, and morphological features in a high dimensional Euclidean space that relates their joint probability to distance. In contrast to (Yatbaz et al., 2012) we build an *instance-based* POS induction system where each instance has a vector representation that concatenates the word vector with the average of the contextual feature vectors. We show that clustering of these instance vectors separate different roles of ambiguous words well, and achieve comparable or better performance than the best word-based systems in matching the gold tags on 19 corpora in 15 languages. All the code that can be used to replicate our findings is available at `https://github.com/ai-ku/upos_2014`.

Section 2 describes the instance-based POS induction algorithm, Section 3 gives the results of our experiments, Section 4 compares the output of the induction system with the gold tags, and Section 5 summarizes our contributions.

2 Algorithm

In this section, we describe the steps of our instance-based POS-induction algorithm:

1. Sample r substitutes for each word instance in the target corpus using an n-gram language model.
2. Construct r tuples for each instance where each tuple consists of a sampled substitute, the target word, and the morphological and orthographical features of the target word (see Table 1).
3. Construct Euclidean embeddings of each word and each feature based on all tuples following Gleberson et al.(2007) and Maron et al.(2010).
4. Construct a vector representation for each instance by concatenating the embedding of the target word with the average of its substitute embeddings.
5. Use k-means clustering to cluster the instance vectors where k is equal to the number of gold tags.

Steps 1 and 2 construct a tuple representation for each instance. Table 1 gives some example tuples for Sentence (1) from the previous section. In this example $r = 3$, so three substitutes are sampled for each instance as contextual features. The sampling is with replacement from the substitute word distribution of a context given by the n-gram language model, so some substitute words may be repeated. The target word and its other features are identical for each of the r tuples representing a single instance.

In step 3, we construct Euclidean embeddings for each unique word and feature value using the multivariable version of the CODE algorithm described in (Globerson et al., 2007). Given two categorical variables W and F, the CODE algorithm constructs Euclidean embeddings (vectors) for each of their distinct values in the same space. The distance between the embedding of a w value, $\phi(w)$, and the embedding of an f value, $\psi(f)$, is related to their joint distribution $p(w, f)$ as follows[3]

$$p(w, f) = \frac{1}{Z}\bar{p}(w)\bar{p}(f)e^{-d^2_{w,f}}$$

where $\bar{p}$ represents empirical probabilities (frequencies from the training data), $d_{w,f}$ is the distance between the embeddings $\phi(w)$ and $\psi(f)$ and $Z = \sum_{w,f} \bar{p}(w)\bar{p}(f)e^{-d^2_{w,f}}$ is a normalization constant.

[2] Substitute probabilities are computed using a 4-gram language model.

[3] (Globerson et al., 2007) describes several ways to relate distances to probabilities, the model used here is the marginal-marginal (MM) model.

Word	Subst	Suf	Cap	Num
Vinken	Makhlouf	–	T	F
Vinken	Makhlouf	–	T	F
Vinken	<unk>	–	T	F
61	20	–	F	T
61	2000	–	F	T
61	eleven	–	F	T
years	years	-s	F	F
years	years	-s	F	F
years	years	-s	F	F

Table 1: The tuples constructed for the instances of "Vinken", "61" and "years" from Sentence (1). The elements of each tuple are the target word, sampled substitute, suffix, capitalization, and number features.

Starting with random vectors for each distinct value of w and f, we use stochastic gradient ascent to move the embedding vectors around to maximize the likelihood given by this model. Calculating the normalization constant Z is the most expensive part of this procedure. We solve this problem following (Maron et al., 2010) who suggest that a constant Z approximation can be used if the embedding vectors are kept on the unit sphere.

As Table 1 shows, considering the target word and its contextual, morphological and orthographic features gives us more than two variables. Yatbaz et al. (2012) adopt the two variable CODE algorithm to this multi-variable case in an ad-hoc manner by considering the target word as w and all other features as distinct f values. We implement the multi-variable extension of CODE suggested by (Globerson et al., 2007) (Section 6.2) which optimizes the following likelihood function:

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

where w are the target words, ϕ are the embeddings of target words, K is the number of different types of features[4], $f^{(i)}$ are the values of the i'th feature, and $\psi^{(i)}$ are the embeddings for the values of the i'th feature. This extension can be seen as a set of K bivariate CODE models $p(w, f^{(i)})$ that share the same target word embeddings $\phi(w)$ but build their own feature embeddings $\psi^{(i)}(f^{(i)})$.

Step 4 constructs a vector representation for each word instance with the concatenation of its word type embedding and the average of its r substitute embeddings. If the original embeddings are in d dimensional space, this results in a $2d$ dimensional vector representing an instance.

Step 5 clusters these $2d$ dimensional instance vectors using a modified k-means algorithm with smart initialization (Arthur and Vassilvitskii, 2007) and assigns each instance to one of k clusters.

3 Experiments

In this section we present our instance-based POS induction experiments. Section 3.1 describes the accuracy metrics that we use to evaluate our results. Section 3.2 details the test corpus and the experimental parameters used in the English experiments and compares our results with previous work. Section 3.3 compares the performance of type and instance based systems on ambiguous words. Finally, Section 3.4 extends the language and corpus coverage by applying the best performing instance based models to 19 corpora in 15 languages.

3.1 Evaluation

We report many-to-one and V-measure scores for our experiments as suggested in (Christodoulopoulos et al., 2010). The many-to-one (MTO) evaluation maps each cluster to its most frequent gold tag and

[4]For example the number of features $K = 4$ in Table 1: Subst, Suf, Cap, and Num.

Model	MTO	VM
Clark (2003)	.712	.655
Christodoulopoulos et al. (2011)	.728	.661
Berg-Kirkpatrick et al. (2010)	.755	-
Christodoulopoulos et al. (2010)	.761	.688
Blunsom and Cohn (2011)	.775	.697
Yatbaz et al. (2012)	.8023 (.0070)	.7207 (.0041)
Instance based (Sec. 2)	.7952 (.0030)	.6908 (.0027)

Table 2: Summary of results with MTO and VM scores for POS induction on the Penn Treebank. Standard errors are given in parentheses when available. All the models incorporate orthographic and morphological features. Berg-Kirkpatrick et al. (2010) and Christodoulopoulos et al. (2010) use instance based models.

reports the percentage of correctly tagged instances. The MTO score can be increased by simply increasing number of clusters, thus the number of clusters is fixed to match the number of gold tags in each experiment. The V-measure (VM) (Rosenberg and Hirschberg, 2007) is an information theory motivated metric that calculates the harmonic mean of completeness and homogeneity of the clusters. Completeness of a cluster is maximized when all instances of a gold-tag are grouped into the same cluster and the homogeneity is maximized when the members of a cluster belong to the same gold-tag.

3.2 Experimental Settings and Results

To make a direct comparison with previously published results, the Wall Street Journal Section of the Penn Treebank was used as the test corpus (1,173,766 instances, 49,206 unique tokens) for English experiments. PTB uses 45 part-of-speech tags which we used as the gold standard for evaluation in our experiments.

To compute substitutes in a given context we trained a language model using the ukWaC corpus ($\approx$ 2 billion tokens) constructed by crawling the ".uk" Internet domain (Ferraresi et al., 2008)[5]. We used SRILM (Stolcke, 2002) to build a 4-gram language model with interpolated Kneser-Ney discounting. Words that were observed less than 2 times in the language model training data were replaced by <unk> tags, which gave us a vocabulary size of 4,254,946. The perplexity of the 4-gram language model on the PTB is 303 and the unknown word rate is 0.008. For computational efficiency only the top 100 substitutes and their probabilities were computed for each position in the PTB using the FASTSUBS algorithm (Yuret, 2012). We use the same orthographic features defined in (Yatbaz et al., 2012) and generated morphological features using the unsupervised algorithm Morfessor (Creutz and Lagus, 2005).

The experiments were run using the following default settings (unless otherwise stated): (1) each word was kept with its original capitalization; (2) 90 substitutes sampled per instance; (3) the learning rate parameters for S-CODE were set to $\varphi_0 = 50$, $\eta_0 = 0.2$; (4) S-CODE convergence threshold, the log-likelihood difference between two consecutive iterations, was set to 0.001; (5) the S-CODE dimensions and $\tilde{Z}$ were set to 25 and 0.166, respectively; (6) a modified k-means algorithm with smart initialization was used (Arthur and Vassilvitskii, 2007); (7) the number of k-means restarts was set to 128 to improve clustering and reduce variance.

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

3.3 Word vs. Instance-Based Induction

Table 2 shows that the overall many-to-one accuracy of our instance based induction system is comparable to (Yatbaz et al., 2012)[6] and significantly higher than the other published results on the Penn Treebank. However Figure 1 in the introduction suggests that this summary hides the large difference in the answers given by the different systems. In this section we compare the performance of our instance-based

[5]We use the Penn Treebank Tokenizer to make the training data compatible with PTB.

[6]The difference is not statistically significant at $p = 0.05$.

model to the word-based model of (Yatbaz et al., 2012) on word types at different levels of ambiguity using the English Penn Treebank results.

We propose the gold-tag perplexity of a word as a measure of its degree of ambiguity defined as:

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

where w is a word, t is a tag, p_w is the gold POS tag distribution of the word w and $H(p_w)$ is the entropy of the p_w distribution. A GP of 1 for a word w indicates that w is always associated with the same POS tag. A word with N equally probable tags would have a GP of N.

Figure 1 plots the gold-tag perplexity versus the smoothed MTO accuracy for the word-based and the instance-based POS induction systems on the Penn Treebank. To compose the plot, we found the best mapping from the induced clusters to the gold-standard tags, then we computed the MTO accuracy for each word using this mapping and plotted the MTO as a function of the word's GP. We used the Nadaraya-Watson kernel regression estimate (Nadaraya, 1964; Watson, 1964) with normal kernel of bandwidth 1.0 to obtain smooth regression lines. The figure shows that the performance of the instance-based induction model does not degrade as much as the word-based model as the ambiguity of the words increase. However, only 14.94% of the instances in the PTB consists of words with GP greater than 1.5 and 45.71% consists of words with GP exactly 1. Thus, the overall accuracy numbers do not adequately reflect the improvement on highly ambiguous words.

3.4 Multilingual Experiments

Following Christodoulopoulos et al. (2011), 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).

To sample substitutes, we trained language models of Bulgarian, Czech, Estonian, Romanian, Danish, German, Dutch, Portuguese, Spanish, Swedish and Turkish with their corresponding TenTen corpora (Jakubíček et al., 2013), and Hungarian, Slovene and Serbian with their corresponding Wikipedia dump files[7]. Serbian shares a common basis with Crotian 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 MULTEXT-East Serbian corpus. We use ukWaC corpora to train English language models.

We used the default settings in Section 3.2 and incorporated only the orthographic features[8]. Extracting unsupervised morphological features for languages with different characteristics would be of great value, but it is beyond the scope of this paper. For each language the number of induced clusters is set to the number of tags in the gold-set. To perform meaningful comparisons with the previous work we train and evaluate our models on the training section of MULTEXT-East[9] and CONLL-X languages (Lee et al., 2010).

Table 3 presents the performance of our instance based model on 19 corpora in 15 languages together with the corresponding best published results from $^{\diamond}$(Yatbaz et al., 2012), ‡(Blunsom and Cohn, 2011), *(Christodoulopoulos et al., 2011) and †(Clark, 2003). All of the state-of-the-art systems in Table 3 are word-based and incorporate morphological features.

Our MTO results are lower than the best systems on all of data-sets that use language models trained on the Wikipedia corpora. ukWaC and TenTen corpora are cleaner and tokenized better compared to the Wikipedia corpora. These corpora also have larger vocabulary sizes and lower out-of-vocabulary rates. Thus language models trained on these corpora have much lower perplexities and generate better

[7]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/`)

[8]All corpora (except German, Spanish and Swedish) label the punctuation marks with the same gold-tag therefore we add an extra *punctuation* feature for those languages.

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

	Language	Tags	Best Published MTO VM	Instance Based MTO / VM
WSJ	English	45	.802 / **.721** $^{\diamond}$	.795 / .691
MULTEXT-East	Bulgarian	12+1	.665 / **.556***	.664 / .513
	Czech	12+1	.642 / **.539***	**.705** / .511
	English	12+1	.733 / .633*	**.835** / **.661**
	Estonian	11+1	.644 / **.533***	.643 / .457
	Hungarian	12+1	**.682** / **.548***	.647/ .459
	Romanian	14+1	.611 / .523*	**.660** / .528
	Slovene	12+1	**.679** / **.567***	.667 / .451
	Serbian	12+1	**.641** / **.510**$\dagger$	.594/ .402
CoNLL-X Shared Task	Bulgarian	54	.704 / **.596**$\dagger$	**.751** / .583
	Czech	12	.701‡ / .484*	.701 / .486
	Danish	25	.761‡ / .591*	.761 / .584
	Dutch	13	.711‡ / **.547***	.712 / .537
	German	54	.744* / **.630**$\dagger$	.749 / .618
	Portuguese	22	.785‡ / **.639***	.782 / .607
	Slovene	29	.642* / **.539**$\dagger$	.638 / .469
	Spanish	47	**.788**‡ / **.632***	.753/ .602
	Swedish	41	.682 / **.589**$\dagger$	.681 / .546
	Turkish	30	.628 / .408*	.637 / .401

Table 3: The MTO and VM scores on 19 corpora in 15 languages together with number of induced clusters. Statistically significant results shown in bold ($p < 0.05$).

substitutes than the Wikipedia based models. Our model has lower VM scores in spite of good MTO scores on 14 corpora which is discussed in Section 4.

Among the languages for which clean language model corpora were available, our model performs comparable to or significantly better than the best systems on most languages. We show significant improvements on MULTEXT-East Czech, Romanian, and CoNLL-X Bulgarian. Our model achieves the state-of-the-art MTO on MULTEXT-East English and scores comparable MTO with the best model on WSJ. Our model shows comparable results on MULTEXT-East Bulgarian and Estonian, and CoNLL-X Czech, Danish, Dutch, German, Portuguese, Swedish and Turkish in terms of the MTO score. One reason for comparably low MTO on Spanish might be the absence of morphological features.

4 Discussion

In this section we perform further analysis on the clustering output of our model. The example below illustrates the advantage of the instance-based approach:

(1) ... it will also **offer** buyers the option ...
Substitutes: give, help, attract

(2) The **offer** is being launched ...
Substitutes: campaign, project, scheme

The word **offer** is a *verb* in the first sentence and a *noun* in the second one. Clustering the word embeddings can not distinguish the different occurrences of the words (Yatbaz et al., 2012). On the other hand, the substitutes of *offer* in the two sentences can disambiguate the correct category of the corresponding occurrences. In our actual experiments our instance based representation distinguishes the instances of **offer** as *noun* (cluster 26 and 12) and *verb* (cluster 35).

To illustrate how words are distributed in the induced clusters, we compare the most frequent clusters of our model in Section 3 with the most frequent gold-tags of PTB in Figure 2.

The low VM performance of our instance-based model compared to the state-of-the-art word-based systems on some languages is due to the completeness part of the VM score. The Hinton diagram in Figure 2 shows that large gold-tag groups are split into several clusters based on the substitutability of words in that particular cluster (rows of the Hinton diagram). For example, proper nouns (*NNP*) are split into three major clusters such that titles like *Mr.* or person names are in (40), nationality or country related words like *Japanese* or U.S are in (22), and the rest of the proper nouns in cluster (30).

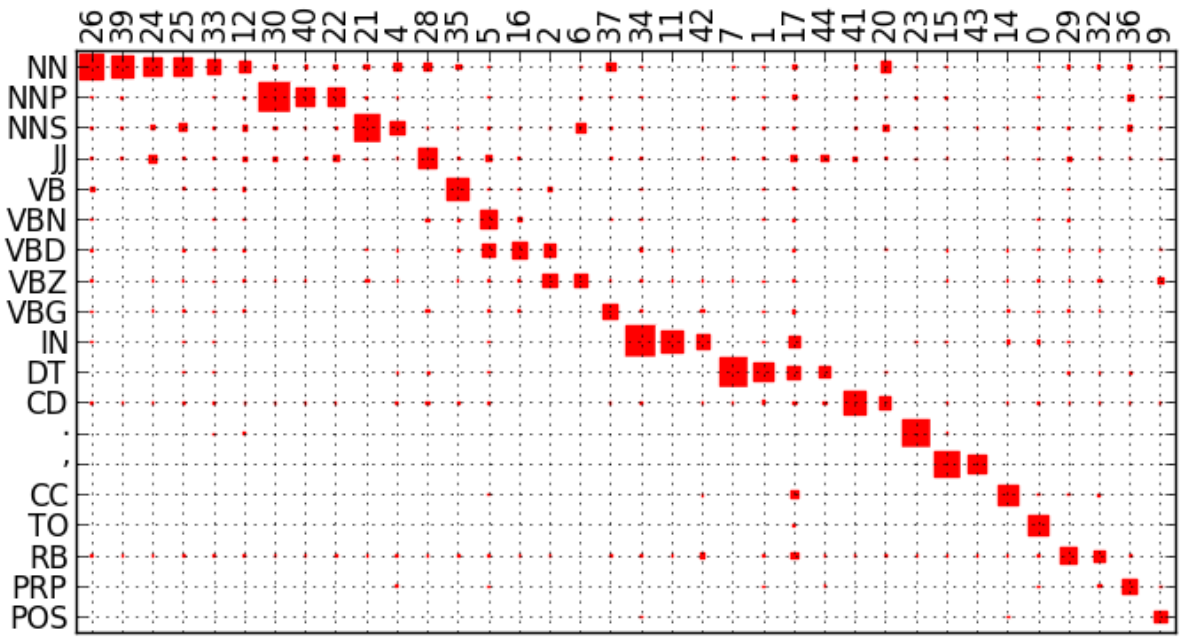

Figure 2: Each row corresponds to a gold tag and each column is an induced tag in the Hinton diagram above. The area of each square is proportional to the joint probability of the given tag and cluster.

The gold-tags of PTB, on the other hand, do not always respect whether words with the same tag are substitutable for one another. Freudenthal et al. (2005) argues, from the child language acquisition perspective, that the standard linguistic definition of syntactic groups requires the substitutability of words in a syntactic category. Word pairs that are placed in the same category in the PTB, such as "Mr." and "Friday", "be" and "run", "not" and "gladly", "of" and "into" are clearly not generally substitutable.

Another noteworty example of completeness error is that our model splits the punctuation mark (,) class of PTB into the clusters 15 and 43 based on the different usage patterns. The majority of the (,) instances in cluster 15 are used in relative clauses, reported speech clauses or conjunctions while cluster 43 generally consists of (,) instances that are used in non-essential clauses (ex: Time, the largest newsweekly, . . .).

5 Contributions

Our main contributions can be summarized as follows:

- We introduced an instance based POS induction system that can handle ambiguous words and is competitive with the word-based systems in overall accuracy.
- We extended the S-CODE framework to handle more than two categorical variables.
- Our instance based system scores 79.5% many-to-one accuracy on the Penn Treebank and achieves results that are significantly better than or comparable with the best published systems on 15 out of 19 corpora in 15 languages.
- All our code and data, including the substitute distributions and word vectors for the PTB, MULTEXT-East and CoNLL-X shared task corpora are available at the authors' website at `https://github.com/ai-ku/upos_2014`.

Acknowledgements

We would like to thank Adam Kilgarriff and the Sketch Engine[10] team for making their corpora available to us. This work was supported in part by the Scientific and Technical Research Council of Turkey (TÜBİTAK Project 112E277).

References

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

D. Arthur and S. Vassilvitskii. 2007. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics.

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

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.

Sabine Buchholz 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.

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

Christos Christodoulopoulos, Sharon Goldwater, and Mark Steedman. 2011. A bayesian mixture model for pos induction using multiple features. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 638–647, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

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

Mathias Creutz and Krista Lagus. 2005. Inducing the morphological lexicon of a natural language from unannotated text. In *Proceedings of AKRR'05, International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning*, pages 106–113, Espoo, Finland, June.

Tomaž Erjavec. 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.

Adriano Ferraresi, Eros Zanchetta, Marco Baroni, and Silvia Bernardini. 2008. Introducing and evaluating ukwac, a very large web-derived corpus of english. In *Proceedings of the 4th Web as Corpus Workshop (WAC-4) Can we beat Google*, pages 47–54.

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.

[10] `https://www.sketchengine.co.uk`

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

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

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

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.

Miloš Jakubíček, Adam Kilgarriff, Vojtěch Kovář, Pavel Rychlỳ, and Vít Suchomel. 2013. The tenten corpus family. In *International Conference on Corpus Linguistics, Lancaster*.

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

Michael Lamar, Yariv Maron, and Elie Bienenstock. 2010a. Latent-descriptor clustering for unsupervised pos induction. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 799–809, Stroudsburg, PA, USA. Association for Computational Linguistics.

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

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

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

Yariv Maron, Michael Lamar, and Elie Bienenstock. 2010. Sphere embedding: An application to part-of-speech induction. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1567–1575.

Elizbar A Nadaraya. 1964. On estimating regression. *Theory of Probability & Its Applications*, 9(1):141–142.

A. Rosenberg and J. Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 410–420.

H. Schütze and J. Pedersen. 1993. A Vector Model for syntagmatic and paradigmatic relatedness. In *Proceedings of the 9th Annual Conference of the University of Waterloo Centre for the New OED and Text Research*, Oxford, England.

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.

Kristina Toutanova, 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.

Geoffrey S Watson. 1964. Smooth regression analysis. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 359–372.

Mehmet Ali Yatbaz, Enis Sert, and Deniz Yuret. 2012. Learning syntactic categories using paradigmatic representations of word context. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 940–951, Jeju Island, Korea, July. Association for Computational Linguistics.

Deniz Yuret. 2012. Fastsubs: An efficient and exact procedure for finding the most likely lexical substitutes based on an n-gram language model. *Signal Processing Letters, IEEE*, 19(11):725–728, Nov.