Unsupervised POS Induction with Word Embeddings

Chu-Cheng Lin **Waleed Ammar** **Chris Dyer** **Lori Levin**
School of Computer Science
Carnegie Mellon University
{chuchenl,wammar,cdyer,lsl}@cs.cmu.edu

Abstract

Unsupervised word embeddings have been shown to be valuable as features in supervised learning problems; however, their role in unsupervised problems has been less thoroughly explored. In this paper, we show that embeddings can likewise add value to the problem of unsupervised POS induction. In two representative models of POS induction, we replace multinomial distributions over the vocabulary with multivariate Gaussian distributions over word embeddings and observe consistent improvements in eight languages. We also analyze the effect of various choices while inducing word embeddings on "downstream" POS induction results.

1 Introduction

Unsupervised POS induction is the problem of assigning word tokens to syntactic categories given only a corpus of untagged text. In this paper we explore the effect of replacing words with their vector space embeddings[1] in two POS induction models: the classic first-order HMM (Kupiec, 1992) and the newly introduced conditional random field autoencoder (Ammar et al., 2014). In each model, instead of using a conditional multinomial distribution[2] to generate a word token $w_i \in V$ given a POS tag $t_i \in T$, we use a conditional Gaussian distribution and generate a d-dimensional word embedding $\mathbf{v}_{w_i} \in \mathbb{R}^d$ given t_i.

[1]Unlike Yatbaz et al. (2014), we leverage easily obtainable and widely used embeddings of word types.

[2]Also known as a categorical distribution.

Our findings suggest that, in both models, substantial improvements are possible when word embeddings are used rather than opaque word types. However, the independence assumptions made by the model used to induce embeddings strongly determines its effectiveness for POS induction: embedding models that model short-range context are more effective than those that model longer-range contexts. This result is unsurprising, but it illustrates the lack of an evaluation metric that measures the syntactic (rather than semantic) information in word embeddings. Our results also confirm the conclusions of Sirts et al. (2014) who were likewise able to improve POS induction results, albeit using a custom clustering model based on the the distance-dependent Chinese restaurant process (Blei and Frazier, 2011).

Our contributions are as follows: (i) reparameterization of token-level POS induction models to use word embeddings; and (ii) a systematic evaluation of word embeddings with respect to the syntactic information they contain.

2 Vector Space Word Embeddings

Word embeddings represent words in a language's vocabulary as points in a d-dimensional space such that nearby words (points) are similar in terms of their distributional properties. A variety of techniques for learning embeddings have been proposed, e.g., matrix factorization (Deerwester et al., 1990; Dhillon et al., 2011) and neural language modeling (Mikolov et al., 2011; Collobert and Weston, 2008).

For the POS induction task, we specifically need embeddings that capture syntactic similarities. Therefore we experiment with two types of embeddings

Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL, pages 1311–1316,
Denver, Colorado, May 31 – June 5, 2015.

that are known for such properties:

- **Skip-gram embeddings** (Mikolov et al., 2013) are based on a log bilinear model that predicts an unordered set of context words given a target word. Bansal et al. (2014) found that smaller context window sizes tend to result in embeddings with more syntactic information. We confirm this finding in our experiments.
- **Structured skip-gram embeddings** (Ling et al., 2015) extend the *standard* skip-gram embeddings (Mikolov et al., 2013) by taking into account the relative positions of words in a given context.

We use the tool `word2vec`[3] and Ling et al. (2015)'s modified version[4] to generate both plain and structured skip-gram embeddings in nine languages.

3 Models for POS Induction

In this section, we briefly review two classes of models used for POS induction (HMMs and CRF autoencoders), and explain how to generate word embedding observations in each class. We will represent a sentence of length ℓ as $\boldsymbol{w} = \langle w_1, w_2, \ldots, w_\ell \rangle \in V^\ell$ and a sequence of tags as $\boldsymbol{t} = \langle t_1, t_2, \ldots, t_\ell \rangle \in T^\ell$. The embeddings of word type $w \in V$ will be written as $\mathbf{v}_w \in \mathbb{R}^d$.

3.1 Hidden Markov Models

The hidden Markov model with multinomial emissions is a classic model for POS induction. This model makes the assumption that a latent Markov process with discrete states representing POS categories emits individual words in the vocabulary according to state (i.e., tag) specific emission distributions. An HMM thus defines the following joint distribution over sequences of observations and tags:

$$p(\boldsymbol{w}, \boldsymbol{t}) = \prod_{i=1}^{\ell} p(t_i \mid t_{i-1}) \times p(w_i \mid t_i) \qquad (1)$$

where distributions $p(t_i \mid t_{i-1})$ represents the transition probability and $p(w_i \mid t_i)$ is the emission probability, the probability of a particular tag generating the word at position i.[5]

We consider two variants of the HMM as baselines:

- $p(w_i \mid t_i)$ is parameterized as a "naïve multinomial" distribution with one distinct parameter for each word type.
- $p(w_i \mid t_i)$ is parameterized as a multinomial logistic regression model with hand-engineered features as detailed in (Berg-Kirkpatrick et al., 2010).

Gaussian Emissions. We now consider incorporating word embeddings in the HMM. Given a tag $t \in T$, instead of generating the observed word $w \in V$, we generate the (pre-trained) embedding $\mathbf{v}_w \in \mathbb{R}^d$ of that word. The conditional probability density assigned to $\mathbf{v}_w \mid t$ follows a multivariate Gaussian distribution with mean $\boldsymbol{\mu}_t$ and covariance matrix $\boldsymbol{\Sigma}_t$:

$$p(\mathbf{v}_w; \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t) = \frac{\exp\left(-\frac{1}{2}(\mathbf{v}_w - \boldsymbol{\mu}_t)^\top \boldsymbol{\Sigma}_t^{-1} (\mathbf{v}_w - \boldsymbol{\mu}_t)\right)}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}_t|}} \qquad (2)$$

This parameterization makes the assumption that embeddings of words which are often tagged as t are concentrated around some point $\boldsymbol{\mu}_t \in \mathbb{R}^d$, and the concentration decays according to the covariance matrix $\boldsymbol{\Sigma}_t$.[6]

Now, the joint distribution over a sequence of observations $\mathbf{v} = \langle \mathbf{v}_{w_1}, \mathbf{v}_{w_2} \ldots, \mathbf{v}_{w_\ell} \rangle$ (which corresponds to word sequence $\boldsymbol{w} = \langle w_1, w_2, \ldots, w_\ell, \rangle$) and a tag sequence $\boldsymbol{t} = \langle t_1, t_2 \ldots, t_\ell \rangle$ becomes:

$$p(\mathbf{v}, \boldsymbol{t}) = \prod_{i=1}^{\ell} p(t_i \mid t_{i-1}) \times p(\mathbf{v}_{w_i}; \boldsymbol{\mu}_{t_i}, \boldsymbol{\Sigma}_{t_i})$$

We use the Baum–Welch algorithm to fit the $\boldsymbol{\mu}_t$ and $\boldsymbol{\Sigma}_{t_i}$ parameters. In every iteration, we update $\boldsymbol{\mu}_{t^*}$ as follows:

$$\boldsymbol{\mu}_{t^*}^{new} = \frac{\sum_{\mathbf{v} \in \mathcal{T}} \sum_{i=1\ldots\ell} p(t_i = t^* \mid \mathbf{v}) \times \mathbf{v}_{w_i}}{\sum_{\mathbf{v} \in \mathcal{T}} \sum_{i=1\ldots\ell} p(t_i = t^* \mid \mathbf{v})} \qquad (3)$$

where $\mathcal{T}$ is a data set of word embedding sequences $\mathbf{v}$ each of length $|\mathbf{v}| = \ell$, and $p(t_i = t^* \mid \mathbf{v})$ is the posterior probability of label t^* at position i in the sequence $\mathbf{v}$. Likewise the update to $\boldsymbol{\Sigma}_{t^*}$ is:

$$\boldsymbol{\Sigma}_{t^*}^{new} = \frac{\sum_{\mathbf{v} \in \mathcal{T}} \sum_{i=1\ldots\ell} p(t_i = t^* \mid \mathbf{v}) \times \boldsymbol{\delta}\boldsymbol{\delta}^\top}{\sum_{\mathbf{v} \in \mathcal{T}} \sum_{i=1\ldots\ell} p(t_i = t^* \mid \mathbf{v})} \qquad (4)$$

where $\boldsymbol{\delta} = \mathbf{v}_{w_i} - \boldsymbol{\mu}_{t^*}^{new}$.

[3] `https://code.google.com/p/word2vec/`

[4] `https://github.com/wlin12/wang2vec/`

[5] Terms for the starting and stopping transition probabilities are omitted for brevity.

[6] *"Essentially, all models are wrong, but some are useful."* — George E. P. Box

3.2 Conditional Random Field Autoencoders

The second class of models this work extends is called CRF autoencoders, which we recently proposed in (Ammar et al., 2014). It is a scalable family of models for feature-rich learning from unlabeled examples. The model conditions on one copy of the structured input, and generates a reconstruction of the input via a set of interdependent latent variables which represent the linguistic structure of interest. As shown in Eq. 5, the model factorizes into two distinct parts: the encoding model $p(\boldsymbol{t} \mid \boldsymbol{w})$ and the reconstruction model $p(\hat{\boldsymbol{w}} \mid \boldsymbol{t})$; where $\boldsymbol{w}$ is the structured input (e.g., a token sequence), $\boldsymbol{t}$ is the linguistic structure of interest (e.g., a sequence of POS tags), and $\hat{\boldsymbol{w}}$ is a generic reconstruction of the input. For POS induction, the encoding model is a linear-chain CRF with feature vector $\boldsymbol{\lambda}$ and local feature functions $\mathbf{f}$.

$$p(\hat{\boldsymbol{w}}, \boldsymbol{t} \mid \boldsymbol{w}) = p(\boldsymbol{t} \mid \boldsymbol{w}) \times p(\hat{\boldsymbol{w}} \mid \boldsymbol{t})$$
$$\propto p(\hat{\boldsymbol{w}} \mid \boldsymbol{t}) \times \exp \boldsymbol{\lambda} \cdot \sum_{i=1}^{|\boldsymbol{w}|} \mathbf{f}(t_i, t_{i-1}, \boldsymbol{w}) \quad (5)$$

In (Ammar et al., 2014), we explored two kinds of reconstructions $\hat{\boldsymbol{w}}$: surface forms and Brown clusters (Brown et al., 1992), and used "stupid multinomials" as the underlying distributions for re-generating $\hat{\boldsymbol{w}}$.

Gaussian Reconstruction. In this paper, we use d-dimensional word embedding reconstructions $\hat{w}_i = \mathbf{v}_{w_i} \in \mathbb{R}^d$, and replace the multinomial distribution of the reconstruction model with the multivariate Gaussian distribution in Eq. 2. We again use the Baum–Welch algorithm to estimate $\boldsymbol{\mu}_{t^*}$ and $\boldsymbol{\Sigma}_{t^*}$ similar to Eq. 3. The only difference is that posterior label probabilities are now conditional on both the input sequence $\boldsymbol{w}$ and the embeddings sequence $\mathbf{v}$, i.e., replace $p(t_i = t^* \mid \mathbf{v})$ in Eq. 2 with $p(t_i = t^* \mid \boldsymbol{w}, \mathbf{v})$.

4 Experiments

In this section, we attempt to answer the following questions:

- §4.1: Do syntactically-informed word embeddings improve POS induction? Which model performs best?
- §4.2: What kind of word embeddings are suitable for POS induction?

4.1 Choice of POS Induction Models

Here, we compare the following models for POS induction:

- Baseline: HMM with multinomial emissions (Kupiec, 1992),
- Baseline: HMM with log-linear emissions (Berg-Kirkpatrick et al., 2010),
- Baseline: CRF autoencoder with multinomial reconstructions (Ammar et al., 2014),[7]
- Proposed: HMM with Gaussian emissions, and
- Proposed: CRF autoencoder with Gaussian reconstructions.

Data. To train the POS induction models, we used the plain text from the training sections of the CoNLL-X shared task (Buchholz and Marsi, 2006) (for Danish and Turkish), the CoNLL 2007 shared task (Nivre et al., 2007) (for Arabic, Basque, Greek, Hungarian and Italian), and the Ukwabelana corpus (Spiegler et al., 2010) (for Zulu). For evaluation, we obtain the corresponding gold-standard POS tags by deterministically mapping the language-specific POS tags in the aforementioned corpora to the corresponding universal POS tag set (Petrov et al., 2012). This is the same set up we used in (Ammar et al., 2014).

Setup. In this section, we used skip-gram (i.e., `word2vec`) embeddings with a context window size = 1 and with dimensionality $d = 100$, trained with the largest corpora for each language in (Quasthoff et al., 2006), in addition to the plain text used to train the POS induction models.[8] In the proposed models, we only show results for estimating $\boldsymbol{\mu}_t$, assuming a diagonal covariance matrix $\boldsymbol{\Sigma}_t(k,k) = 0.45 \forall k \in \{1, \ldots, d\}$.[9] While the CRF autoencoder with multinomial reconstructions were carefully initialized as

[7] We use the configuration with best performance which reconstructs Brown clusters.

[8] We used the `corpus/tokenize-anything.sh` script in the cdec decoder (Dyer et al., 2010) to tokenize the corpora from (Quasthoff et al., 2006). The other corpora were already tokenized. In Arabic and Italian, we found a lot of discrepancies between the tokenization used for inducing word embeddings and the tokenization used for evaluation. We expect our results to improve with consistent tokenization.

[9] Surprisingly, we found that estimating $\boldsymbol{\Sigma}_t$ significantly degrades the performance. This may be due to overfitting (Shinozaki and Kawahara, 2007). Possible remedies include using a prior (Gauvain and Lee, 1994).

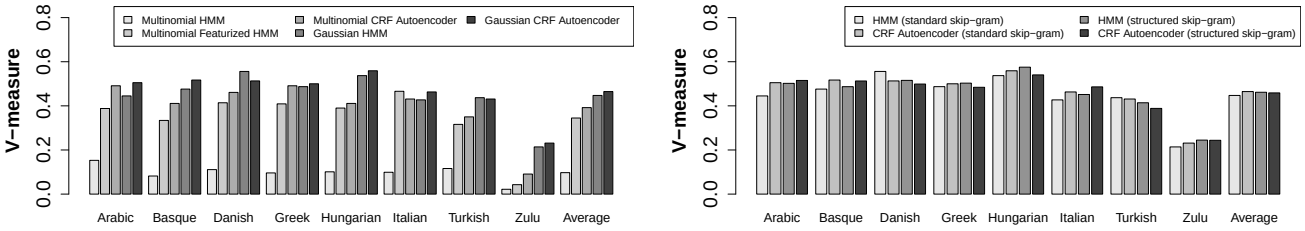

Figure 1: POS induction results. (V-measure, higher is better.) Window size is 1 for all word embeddings. **Left**: Models which use standard skip-gram word embeddings (i.e., Gaussian HMM and Gaussian CRF Autoencoder) outperform all baselines on average across languages. **Right**: comparison between standard and structured skip-grams on Gaussian HMM and CRF Autoencoder.

discussed in (Ammar et al., 2014), CRF autoencoder with Gaussian reconstructions were initialized uniformly at random in $[-1, 1]$. All HMM models were also randomly initialized. We tuned all hyperparameters on the English PTB corpus, then fixed them for all languages.

Evaluation. We use the V-measure evaluation metric (Rosenberg and Hirschberg, 2007) to evaluate the predicted syntactic classes at the token level.[10]

Results. The results in Fig. 1 clearly suggest that we can use word embeddings to improve POS induction. Surprisingly, the feature-less Gaussian HMM model outperforms the strong feature-rich baselines: Multinomial Featurized HMM and Multinomial CRF Autoencoder. One explanation is that our word embeddings were induced using larger unlabeled corpora than those used to train the POS induction models. The best results are obtained using both word embeddings and feature-rich models using the Gaussian CRF autoencoder model. This set of results suggest that word embeddings and hand-engineered features play complementary roles in POS induction. It is worth noting that the CRF autoencoder model with Gaussian reconstructions did not require careful initialization.[11]

[10]We found the V-measure results to be consistent with the many-to-one evaluation metric (Johnson, 2007). We only show one set of results for brevity.

[11]In (Ammar et al., 2014), we found that careful initialization for the CRF autoencoder model with multinomial reconstructions is necessary.

4.2 Choice of Embeddings

Standard skip-gram vs. structured skip-gram. On Gaussian HMMs, structured skip-gram embeddings score moderately higher than standard skip-grams. And as context window size gets larger, the gap widens (as shown in Fig. 2.) The reason may be that structured skip-gram embeddings give each position within the context window its own project matrix, so the smearing effect is not as pronounced as the window grows when compared to the standard embeddings. However the best performance is still obtained when window size is small.[12]

Dimensions = 20 vs. 200. We also varied the number of dimensions in the word vectors ($d \in \{20, 50, 100, 200\}$). The best V-measure we obtain is 0.504 ($d = 20$) and the worst is 0.460 ($d = 100$). However, we did not observe a consistent pattern as shown in Fig. 3.

Window size = 1 vs. 16. Finally, we varied the window size for the context surrounding target words ($w \in \{1, 2, 4, 8, 16\}$). $w = 1$ yields the best average V-measure across the eight languages as shown in Fig. 2. This is true for both standard and structured

[12]In preliminary experiments, we also compared standard skip-gram embeddings to SENNA embeddings (Collobert et al., 2011) (which are trained in a semi-supervised multi-task learning setup, with one task being POS tagging) on a subset of the English PTB corpus. As expected, the induced POS tags are much better when using SENNA embeddings, yielding a V-measure score of 0.57 compared to 0.51 for skip-gram embeddings. Since SENNA embeddings are only available in English, we did not include it in the comparison in Fig. 1.

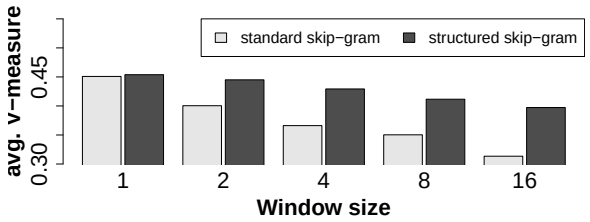

Figure 2: Effect of window size and embeddings type on POS induction over the languages in Fig. 1. d = 100. The model is HMM with Gaussian emissions.

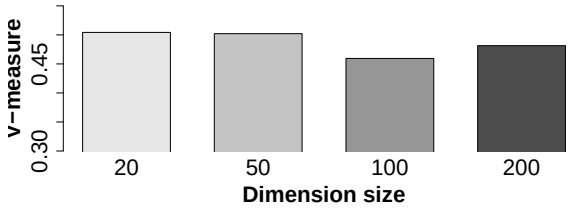

Figure 3: Effect of dimension size on POS induction on a subset of the English PTB corpus. w = 1. The model is HMM with Gaussian emissions.

skip-gram models. Notably, larger window sizes appear to produce word embeddings with less syntactic information. This result confirms the observations of Bansal et al. (2014).

4.3 Discussion

We have shown that (re)generating word embeddings does much better than generating opaque word types in unsupervised POS induction. At a high level, this confirms prior findings that unsupervised word embeddings capture syntactic properties of words, and shows that different embeddings capture more syntactically salient information than others. As such, we contend that unsupervised POS induction can be seen as a diagnostic metric for assessing the syntactic quality of embeddings.

To get a better understanding of what the multivariate Gaussian models have learned, we conduct a hill-climbing experiment on our English dataset. We seed each POS category with the average vector of 10 randomly sampled words from that category and train the model. Seeding unsurprisingly improves tagging performance. We also find words that are the nearest to the centroids generally agree with the correct category label, which validate our assumption that syntactically similar words tend to cluster in the high-dimensional embedding space. It also shows that careful initialization of model parameters can bring further improvements.

However we also find that words that are close to the centroid are not necessarily representative of what linguists consider to be prototypical. For example, Hopper and Thompson (1983) show that physical, telic, past tense verbs are more prototypical with respect to case marking, agreement, and other syntactic behavior. However, the verbs nearest our centroid all seem rather abstract. In English, the nearest 5 words in the verb category are *entails, aspires, attaches, foresees, deems*. This may be because these words seldom serve functions other than verbs; and placing the centroid around them incurs less penalty (in contrast to physical verbs, e.g. *bite*, which often also act as nouns). Therefore one should be cautious in interpreting what is prototypical about them.

5 Conclusion

We propose using a multivariate Gaussian model to generate vector space representations of observed words in generative or hybrid models for POS induction, as a superior alternative to using multinomial distributions to generate categorical word types. We find the performance from a simple Gaussian HMM competitive with strong feature-rich baselines. We further show that substituting the emission part of the CRF autoencoder can bring further improvements. We also confirm previous findings which suggest that smaller context windows in skip-gram models result in word embeddings which encode more syntactic information. It would be interesting to see if we can apply this approach to other tasks which require generative modeling of textual observations such as language modeling and grammar induction.

Acknowledgements

This work was sponsored by the U.S. Army Research Laboratory and the U.S. Army Research Office under contract/grant numbers W911NF-11-2-0042 and W911NF-10-1-0533. The statements made herein are solely the responsibility of the authors.

References

Waleed Ammar, Chris Dyer, and Noah A. Smith. 2014. Conditional random field autoencoders for unsupervised structured prediction. In *NIPS*.

Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2014. Tailoring continuous word representations for dependency parsing. In *Proc. of ACL*.

Taylor Berg-Kirkpatrick, Alexandre Bouchard-Côté, John DeNero, and Dan Klein. 2010. Painless unsupervised learning with features. In *Proc. of NAACL*.

David M. Blei and Peter I. Frazier. 2011. Distance dependent Chinese restaurant processes. *JMLR*.

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. *Computational Linguistics*.

Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *CoNLL-X*.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proc. of ICML*.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *JMLR*, 12:2493–2537, November.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

Paramveer S. Dhillon, Dean Foster, and Lyle Ungar. 2011. Multi-view learning of word embeddings via CCA. In *NIPS*, volume 24.

Chris Dyer, Adam Lopez, Juri Ganitkevitch, Johnathan Weese, Ferhan Ture, Phil Blunsom, Hendra Setiawan, Vladimir Eidelman, and Philip Resnik. 2010. cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *Proc. of ACL*.

J. Gauvain and Chin-Hui Lee. 1994. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *Speech and Audio Processing, IEEE Transactions on*, 2(2):291–298, Apr.

Paul Hopper and Sandra Thompson. 1983. The iconicity of the universal categories "noun" and "verb". In John Haiman, editor, *Iconicity in Syntax: Proceedings of a symposium on iconicity in syntax*.

Mark Johnson. 2007. Why doesn't EM find good HMM POS-taggers? In *Proc. of EMNLP*.

Julian Kupiec. 1992. Robust part-of-speech tagging using a hidden Markov model. *Computer Speech and Language*, 6:225–242.

Wang Ling, Chris Dyer, Alan Black, and Isabel Trancoso. 2015. Two/too simple adaptations of word2vec for syntax problems. In *Proc. of NAACL*.

Tomas Mikolov, Stefan Kombrink, Anoop Deoras, Lukar Burget, and J Cernocky. 2011. RNNLM — recurrent neural network language modeling toolkit. In *Proc. of the 2011 ASRU Workshop*, pages 196–201.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *ArXiv e-prints*, January.

Joakim Nivre, Johan Hall, Sandra Kubler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007. The CoNLL 2007 shared task on dependency parsing. In *Proc. of CoNLL*.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proc. of LREC*, May.

Uwe Quasthoff, Matthias Richter, and Christian Biemann. 2006. Corpus portal for search in monolingual corpora. In *Proc. of LREC*, pages 1799–1802.

Andrew Rosenberg and Julia Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *EMNLP-CoNLL*.

T. Shinozaki and T. Kawahara. 2007. HMM training based on CV-EM and CV gaussian mixture optimization. In *Proc. of the 2007 ASRU Workshop*, pages 318–322, Dec.

Kairit Sirts, Jacob Eisenstein, Micha Elsner, and Sharon Goldwater. 2014. POS induction with distributional and morphological information using a distance-dependent Chinese restaurant process. In *Proc. of ACL*.

Sebastian Spiegler, Andrew van der Spuy, and Peter A. Flach. 2010. Ukwabelana: An open-source morphological Zulu corpus. In *Proc. of COLING*, pages 1020–1028.

Mehmet Ali Yatbaz, Enis Rıfat Sert, and Deniz Yuret. 2014. Unsupervised instance-based part of speech induction using probable substitutes. In *Proc. of COLING*.