Learning a POS tagger for AAVE-like language[*]

Anna Jørgensen
University of Amsterdam
Science Park 107
1098 XG Amsterdam, NL
jorgensen@uva.nl

Dirk Hovy
University of Copenhagen
Njalsgade 140
2300 Copenhagen S, DK
dirk.hovy@hum.ku.dk

Anders Søgaard
University of Copenhagen
Njalsgade 140
2300 Copenhagen S, DK
soegaard@hum.ku.dk

Abstract

Part-of-speech (POS) taggers trained on newswire perform much worse on domains such as subtitles, lyrics, or tweets. In addition, these domains are also heterogeneous, e.g., with respect to registers and dialects. In this paper, we consider the problem of learning a POS tagger for subtitles, lyrics, and tweets associated with African-American Vernacular English (AAVE). We learn from a mixture of randomly sampled and manually annotated Twitter data and unlabeled data, which we automatically and partially label using mined tag dictionaries. Our POS tagger obtains a tagging accuracy of 89% on subtitles, 85% on lyrics, and 83% on tweets, with up to 55% error reductions over a state-of-the-art newswire POS tagger, and 15-25% error reductions over a state-of-the-art Twitter POS tagger.

1 Introduction

Modern part-of-speech (POS) taggers perform well on what some consider canonical language, as found in domains such as newswire, for which sufficient manually-annotated data is available. For many domains, such as subtitles, lyrics, and tweets, however, labeled data is scarce, if existing, and the performance of off-the-shelf POS taggers is prohibitive of downstream applications.

Furthermore, subtitles, lyrics and tweets are very heterogeneous. Subtitles span from Shakespeare to The Wire, and the lyrics of Elvis Costello are very different from those of Tupac Shakur. Twitter can be anything from teenagers discussing where to go tonight, to researchers discussed the implications of new findings. All three sources of data exhibit a very high degree of linguistic variation, some of which is due to the dialects of the speakers or authors.

In this paper, we use a **corpus of POS-annotated tweets** recently released by CMU,[1] consisting of semi-randomly sampled US tweets. We want to use this corpus to learn a POS tagger for **subtitles, lyrics, and tweets**, which are typically associated with **African-American Vernacular English** (AAVE). We believe our POS tagger can broaden the coverage of NLP tools, and serve as an important tool for large-scale sociolinguistic analyses of language use associated with AAVE (Jørgensen et al., 2015; Stewart, 2014), which relies on the accuracy of these NLP tools.

We combine several recent trends in domain adaptation, namely word embeddings, clusters, sampling, and the use of type constraints. Word representations learned from representative unlabeled data, such as word clusters or embeddings, have been proven useful for increasing the accuracy of NLP tools for low-resource languages and domains (Owoputi et al., 2013; Aldarmaki and Diab, 2015; Gouws and Søgaard, 2015). Since similar words receive similar labels, this can give the model support for words not in the training data. In this paper, we use word clusters and word embeddings in both our baseline and system models.

Using unlabeled data to estimate a target distribution for importance sampling, or for semi-supervised

[*] This work was supported by ERC Starting Grant No. 313695.

[1] https://github.com/brendano/ark-tweet-nlp/tree/master/data/twpos-data-v0.3

Proceedings of NAACL-HLT 2016, pages 1115–1120,
San Diego, California, June 12-17, 2016. ©2016 Association for Computational Linguistics

learning (Søgaard, 2013), as well as wide-coverage, crowd-sourced tag dictionaries to obtain more robust predictions for out-of-domain data have been succesfully used for domain adaptation (Das and Petrov, 2011; Hovy et al., 2015a; Li et al., 2012). In this paper, we use automatically-harvested tag dictionaries for the target variety(/-ies) in two different settings: for labeling the unlabeled data using a technique elaborating on previous work (Li et al., 2012; Wisniewski et al., 2014; Hovy et al., 2015a), and for imposing type constraints at test time in a semi-supervised setting (Garrette and Baldridge, 2013; Plank et al., 2014a). Our best models are obtained using partially labeled training data created using tag dictionaries.

Our contributions We present a POS tagger for AAVE-like language, mining tag dictionaries from various websites and using them to create partially labeled data. Our contributions include: (i) a POS tagger that performs significantly better than existing tools on three datasets containing AAVE markers, (ii) a new domain adaptation algorithm combining ambiguous and cost-sensitive learning, and (iii) an annotated corpus and trained POS tagger made publicly available at `https://bitbucket.org/soegaard/aave-pos16`.

2 Data

For historical reasons, most of the manually annotated corpora available today are newswire corpora. In contrast, very little data is available for domains such as subtitles, lyrics and tweets — especially for language varieties such as AAVE. Learning robust models for AAVE-like language and other language varieties is often further complicated by the absence of standard writing systems (Boujelbane et al., 2013; Bernhard and Ligozat, 2013; Duh and Kirchhoff, 2005).

In this paper, we use three manually annotated data sets, consisting of **subtitles** from the television series *The Wire*, **hip-hop lyrics** from black American artists and **tweets** posted within the south-eastern corner of the United States. We do *not* use this data for training, but only for evaluation, so our experiments use unsupervised (or weakly supervised) domain adaptation.

Although the language use in the three domains vary, they have several things in common: the register is very informal, and the subtitles, lyrics and tweets contain **slang terms** such as *loc'd out*, *cheesing with* and *po'*, **spoken language features** such as *uh-hum, huh* and *oh*, **phonologically-motivated spelling variations** such as *dat mouf*, *missin'* and *niggas* and **contractions** such as *we'll* and *I'd*. These features are infrequent in or absent from most commonly used training corpora for NLP.

The data was annotated by two trained linguists with experience in analyzing AAVE, using the Universal Part-of-Speech tagset (Petrov et al., 2011). They obtained an inter-annotator agreement score of 93.6%. The test sections consist of 528 sentences (subtitles), 509 sentences (lyrics), and 374 sentences (tweets). In addition, we had 546 sentences of subtitles annotated for development data. Note that we only use one domain for development to avoid overly optimistic performance estimates.

For all experiments, we use a publicly available implementation of structured perceptron[2] and train on the 1827 tweets from the CMU Twitter Corpus (Gimpel et al., 2011). Note that despite the fact that the training data also comes from an informal domain, the distribution of POS tags in this data set is different from those of the test sets. For instance, the percentage of determiners in the CMU Twitter corpus is on average 4% lower than in our test domains, and there are 7% more pronouns in the test sets than in the CMU Twitter corpus.

We also create a large **unlabeled corpus** of data that is representative of our test sets. This corpus, consisting of 4.5M sentences, is created using subtitles from the TV series *The Wire* and *The Boondocks*, English hip-hop lyrics, and tweets from the south-eastern states of the US. None of the unlabeled data overlaps with our evaluation datasets. We use this corpus for two purposes: to induce word clusters and embeddings, and to partially annotate a portion of it automatically, which we include in the training data of our ambiguous supervision model (see Section 3 below).

3 Robust learning

Word representations To learn word embeddings from our unlabeled corpus, we use the Gensim im-

[2] `https://github.com/coastalcph/rungsted`

plementation of the word2vec algorithm (Mikolov et al., 2013b; Mikolov et al., 2013a). We also learn Brown clusters from a large corpus of tweets[3] (Owoputi et al., 2013), and add both as additional features to our training and test sets. The word representations capture latent similarities between words, but more importantly enable our tagging model to generalize to unseen words.

Partially labeled data Model performance generally benefits from additional data and constraints during training (Hovy and Hovy, 2012; Täckström et al., 2013). We therefore also use the unlabeled data and tag dictionaries as additional, partially labeled training data. For this purpose, we extract a tag dictionary for AAVE-like language from various crowdsourced online lexicons.

Partial constraints from tag dictionaries have previously been used to filter out incorrect label sequences from projected labels from parallel corpora (Wisniewski et al., 2014; Das and Petrov, 2011; Täckström et al., 2013). We use a combination of a publicly available dump of *Wiktionary*[4] (Li et al., 2012), entries from *Hepster's glossary of musical terms*[5], a list of African-American names[6] and *Urban Dictionary*[7] (UD). We augment our tag dictionary by scraping UD for all words in our unlabeled corpus and extracting the part-of-speech information where available. See an example entry for the word *hooch* below, which has five possible parts of speech in our tag dictionary: VERB, NOUN, ADJ, PRON, ADV.

> *Hooch*: ”Chewing tobacco commonly placed in the lower lip region. Hooch can be used as a verb, noun, adjective, pronoun, or an adverb.”

We use the tag dictionary to label the unlabeled corpus. E.g., when we see the word *hooch*, we assign it the label VERB/NOUN/ADJ/PRON/ADV. We present two ways of using this data for learning better POS models: one where the tag dictionaries are used in an ambiguously supervised setting, and one where they are used as type constraints at prediction time in a self-training setup.

Ambiguous supervision Our algorithm is related to work in cross-lingual transfer (Wisniewski et al., 2014; Das and Petrov, 2011; Täckström et al., 2013) and domain adaptation (Hovy et al., 2015a; Plank et al., 2014a), where tag dictionaries are used to filter projected annotation. We use the tag dictionaries to obtain partial labeling of in-domain training data.

Our baseline sequence labeling algorithm is the structured perceptron (Collins, 2002). This algorithm performs additive updates passing over labeled data, comparing predicted sequences to gold standard sequences. If the predicted sequence is identical to the gold standard, no update is performed. We use a cost-sensitive structured perceptron (Plank et al., 2014b) to learn from the partially labeled data.

Each update for a sequence can be broken down into a series of transition and emission updates, passing over the sequence item-by-item from left to right. For a word like *hooch* labeled VERB/NOUN/ADJ/PRON/ADV, we perform an update proportional to the cost associated with the predicted label. If the predicted label is not in the mined label set, e.g., PRT, we update with a cost of 1.0 (multiplied by the learning rate α); if the predicted label is in the mined label set, we do not update our model. This means that the POS model is not penalized for predicting any of the five supplied labels. We did consider distributing a small cost between the candidates in the mined label sets, but this led to slightly worse performance on our development data.

In the experiments below, we also filter the partially labeled data by the amount of ambiguity observed in our labels. At one extreme, we require *all* words to have a single label, as in fully labeled data. Hovy et al. (2015b) also used a tag dictionary to obtain fully labeled data for domain adaptation. At the other end of the scale, we use all the partially labeled data, allowing up to 12 tags per words. Finally, we also experiment with using only sentences from our unlabeled data such that the tag dictionary assigns at most two (2) or three (3) labels to each word.

We also experimented with using different

[3] http://www.cs.cmu.edu/~ark/TweetNLP/

[4] https://code.google.com/p/wikily-supervised-pos-tagger/

[5] http://www.dinosaurgardens.com/wp-content/uploads/2007/12/hepsters.html

[6] http://www.behindthename.com/submit/names/usage/african-american/3

[7] http://www.urbandictionary.com

	Baselines			Ambiguous	Self-train	Stanford	GATE	CMU
Test set	Baseline	+Cluster	+Clust+Emb					
Lyrics	83.9	85.0	**85.2**	**85.2**	85.0	77.7	83.0	81.5
Subtitles	87.8	88.4	**89.0**	**89.0**	88.8	83.7	87.5	85.6
Tweets	75.0	79.0	78.8	**83.0**	80.0	61.4	77.1	80.0
Average	82.2	84.1	84.3	**85.7**	84.6	74.3	82.5	82.4

Table 1: Main results

amounts of ambiguously labeled data. The best performing system on development data uses both Wiktionary and the tag dictionaries associated with AAVE, only 100 ambiguously labeled data points for training, a cost of 0.0 for predicting labels in the mined label sets, no threshold on ambiguity levels (but leaving only sentences covered by our tag dictionaries), the CMU Brown clusters, and 20-dimensional word2vec embeddings with a sliding window of nine (9). The results of this system are shown in Table 1 as **Ambiguous**.

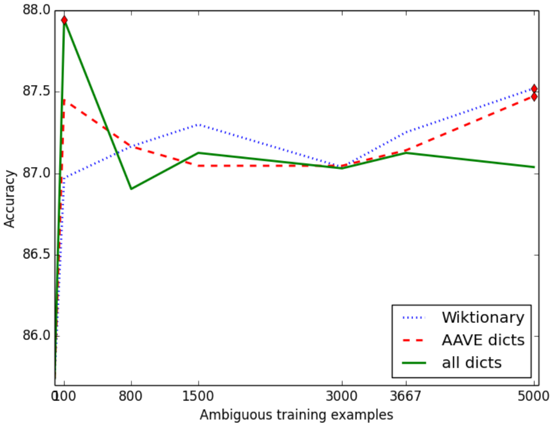

Figure 1: Learning curve ambiguous learning

Self-training with type constraints Our second system uses the harvested tag dictionary for type constraints when making predictions on the unlabeled data for self-training. The search space of possible labels for each word is simply restricted to the tags provided for that word by the tag dictionary.

For our self-training experiments, we experiment with pool size, but heuristically set the stopping criterion to be when the development set accuracy of the tagger decreases over three consecutive iterations. we obtained the best performance on development data using the tag dictionary without Wikipedia, using all entries for type constraints, the CMU Brown clusters, and 10-dimensional embeddings with a window size of five (5). The results of this model are listed in Table 1 as **Self-training**.

Pre-Normalization We also experimented with test-time pre-normalization of the input, using the normalization dictionary of Han et al. (2011), but this led to worse performance on development data.

4 Results and error analysis

Table 1 shows the baseline accuracies, with and without clusters and embeddings, as well as the performance of the two developed systems described above. All results for both ambiguous supervision and self-training with type constraints significantly outperform the simple baseline with $p < 0.01$ (Wilcoxon). The system using ambiguous supervision is also significantly better than the baseline with clusters and word embeddings on the Twitter data. The fact that we generally see worse performance on Twitter data than on the two other data set (even though the systems were trained on Twitter data) can be attributed to a higher type-token ratio.

We also provide the accuracies of three publicly available POS taggers in Table 1. The three POS systems are the bidirectional Stanford Log-linear POS Tagger[8] , the GATE Twitter POS tagger[9] , and the CMU POS Tagger.[10] We observe that our ambiguous learning system outperforms all three systems on all test sets.

[8] http://nlp.stanford.edu/software/tagger.shtml

[9] https://gate.ac.uk/wiki/twitter-postagger.html

[10] https://github.com/brendano/ark-tweet-nlp/

Test set	Lyrics	Subtitles	Tweets	Av.
Baseline	64%	78%	48%	63%
Ambiguous	**71%**	**83%**	**78%**	**77%**
Self-train	70%	82%	61%	71%

Table 2: Accuracies on unseen words

Our improvements are primarily due to better performance on unseen words. Both systems improve the accuracy on OOV items for all three test sets, with the ambiguous learning system reducing the error by an average of 14%, and the self-training system reducing it by 7.7% on average. However, we also see an average increase in performance on known words of 1% for both systems. This increase is highest for tweets (2%) and around 0.5% for the subtitles and hip-hop lyrics test sets. The main reason for the increased overall performances of our systems is therefore the improved accuracy on OOV words. Table 2 shows that the accuracy on OOVs increases on all three test sets for both developed systems over baseline.

The OOV words learned in these two test sets are mainly verbs such as *sittin'*, *gettin'* and *feelin'* (g-dropped spellings), and words that are infrequent in canonical written language such as *'em* and *ho*.

We observe that our systems improve performance on traditionally closed word classes such as pronouns, adpositions, determiners and conjunctions. These increases can be ascribed to the systems having learned from the additional information provided on spelling variations such as *'cause*, *fo'* and *ya* and unknown entities such as *dis*, *dat*, *sum*.

Finally, we note that increasing the number of training examples for ambiguous learning seems to come with diminishing returns. The learning curve is presented in Figure 1.

5 Conclusions

We explore several techniques to learn better POS models for AAVE-like subtitles, lyrics, and tweets from a manually annotated Twitter corpus. Our systems perform significantly better than three state-of-the-art POS taggers for English, with error reductions up to 55%. The improvements were shown to be primarily due to better handling of OOV words.

References

Hanan Aldarmaki and Mona Diab. 2015. Robust part-of-speech tagging of Arabic text. In *Proceedings of the Second Workshop on Arabic Natural Language Processing*, Beijing, China.

Delphine Bernhard and Anne-Laure Ligozat. 2013. Hassle-free pos-tagging for the Alsatian dialects. In Marcos Zampieri and Sascha Diwersy, editors, *Non-Standard Data Sources in Corpus Based-Research*, pages 85–92. ZSM Studien.

Rahma Boujelbane, Meriem Ellouze Khemekhem, and Lamia Hadrich Belguith. 2013. Mapping rules for building a Tunisian dialect lexicon and generating corpora. In *International Joint Conference on Natural Language Processing*, pages 419–428.

Michael Collins. 2002. Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. In *EMNLP*.

Dipanjan Das and Slav Petrov. 2011. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *ACL*, pages 256–263.

Kevin Duh and Katrin Kirchhoff. 2005. Pos tagging of dialectal Arabic: A minimally supervised approach. In *Proceedings in the ACL Workshop on Computational Approaches to Semitic Languages*.

Dan Garrette and Jason Baldridge. 2013. Learning a part-of-speech tagger from two hours of annotation. In *Proceedings of NAACL-HLT*, pages 138–147.

Kevin Gimpel, Nathan Schneider, Brendan OConnor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. 2011. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *ACL*.

Stephen Gouws and Anders Søgaard. 2015. Simple task-specific bilingual word embeddings. In *Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL*, pages 1386–1390.

Bo Han and Timothy Baldwin. 2011. Lexical Normalisation of Short Text Messages: Makn Sens a #twitter. In *ACL*.

Dirk Hovy and Eduard Hovy. 2012. Exploiting partial annotations with em training. In *Proceedings of the NAACL-HLT Workshop on the Induction of Linguistic Structure*, pages 31–38. Association for Computational Linguistics.

Dirk Hovy, Barbara Plank, Héctor Martínez Alonso, and Anders Søgaard. 2015a. Mining for unambiguous instances to adapt part-of-speech taggers to new domains. In *Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL*, pages 1256–1261.

Dirk Hovy, Barbara Plank, Héctor Martínez Alonso, and Anders Søgaard. 2015b. Mining for unambiguous instances to adapt pos taggers to new domains. In *NAACL-HLT*.

Anna Jørgensen, Dirk Hovy, and Anders Søgaard. 2015. Challenges of studying and processing dialects in social media. In *Proceedings of the ACL Workshop on Noisy User-generated Text*.

Shen Li, João V. Graça, and Ben Taskar. 2012. Wiki-ly supervised part-of-speech tagging. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*.

T. Mikolov, K. Chen, G Corrado, and J. Dean. 2013a. Efficient estimation of word representations in vector space. *ArXiv e-prints*.

T. Mikolov, W.T. Yih, and G. Zweig. 2013b. Linguistic regularities in continuous space word representations. In *Proceedings of NAACL-HLT*.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A. Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of NAACL-HLT*.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2011. A universal part-of-speech tagset. In *Proceedings of LREC*.

Barbara Plank, Dirk Hovy, Ryan McDonald, and Anders Søgaard. 2014a. Adapting taggers to twitter with not-so-distant supervision. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1783–1792.

Barbara Plank, Dirk Hovy, and Anders Søgaard. 2014b. Learning part-of-speech taggers with inter-annotator agreement loss. In *EACL*.

Anders Søgaard. 2013. *Semi-supervised learning and domain adaptation for NLP*. Morgan & Claypool.

Ian Stewart. 2014. Now We Stronger Than Ever: African-American syntax on Twitter. In *Proceedings of the Student Research Workshop to the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 26–30, Gothenburg, Sweden, April.

Oscar Täckström, Dipanjan Das, Slav Petrov, Ryan McDonald, and Joakim Nivre, 2013. *Token and Type Constraints for Cross-Lingual Part-of-Speech Tagging*, pages 1–12. Association for Computational Linguistics.

Guillaume Wisniewski, Nicolas Pécheux, Sophir Gahbiche-Braham, and François Yvon. 2014. Cross-lingual part-of-speech tagging through ambiguous learning. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*.