Adapting taggers to Twitter with not-so-distant supervision

Barbara Plank[1], **Dirk Hovy**[1], **Ryan McDonald**[2] **and Anders Søgaard**[1]
Center for Language Technology, University of Copenhagen[1]
Google Inc.[2]
{bplank,dirkh}@cst.dk,ryanmcd@google.com,soegaard@hum.ku.dk

Abstract

We experiment with using different sources of distant supervision to guide unsupervised and semi-supervised adaptation of part-of-speech (POS) and named entity taggers (NER) to Twitter. We show that a particularly good source of not-so-distant supervision is linked websites. Specifically, with this source of supervision we are able to improve over the state-of-the-art for Twitter POS tagging (89.76% accuracy, 8% error reduction) and NER (F1=79.4%, 10% error reduction).

1 Introduction

Twitter contains a vast amount of information, including first stories and breaking news (Petrovic et al., 2010), fingerprints of public opinions (Jiang et al., 2011) and recommendations of relevance to potentially very small target groups (Benson et al., 2011). In order to automatically extract this information, we need to be able to analyze tweets, e.g., determine the part-of-speech (POS) of words and recognize named entities. Tweets, however, are notoriously hard to analyze (Foster et al., 2011; Eisenstein, 2013; Baldwin et al., 2013). The challenges include dealing with variations in spelling, specific conventions for commenting and retweeting, frequent use of abbreviations and emoticons, non-standard syntax, fragmented or mixed language, etc.

Gimpel et al. (2011) showed that we can induce POS tagging models with high accuracy on in-sample Twitter data with relatively little annotation effort. Learning taggers for Twitter data from small amounts of labeled data has also been explored by others (Ritter et al., 2011; Owoputi et al., 2013; Derczynski et al., 2013). Hovy et al. (2014), on the other hand, showed that these models overfit their respective samples and suffer severe drops when evaluated on out-of-sample Twitter data, sometimes performing even worse than newswire models. This may be due to drift on Twitter (Eisenstein, 2013) or simply due to the heterogeneous nature of Twitter, which makes small samples biased. So while existing systems perform well on their own (in-sample) data sets, they over-fit the samples they were induced from, and suffer on other (out-of-sample) Twitter data sets. This bias can, at least in theory, be corrected by learning from additional unlabeled tweets. This is the hypothesis we explore in this paper.

We present a semi-supervised learning method that does not require additional labeled in-domain data to correct sample bias, but rather leverages pools of unlabeled Twitter data. However, since taggers trained on newswire perform poorly on Twitter data, we need additional guidance when utilizing the unlabeled data. This paper proposes distant supervision to help our models learn from unlabeled data. Distant supervision is a weakly supervised learning paradigm, where a knowledge resource is exploited to gather (possible noisy) training instances (Mintz et al., 2009). Our basic idea is to can use linguistic analysis of linked websites as a novel kind of distant supervision for learning how to analyze tweets. We explore standard sources of distant supervision, such as Wiktionary for POS tagging, but we also propose to use the linked websites of tweets with URLs as supervision. The intuition is that we can use websites to provide a richer linguistic context for our tagging decisions. We exploit the fact that tweets with URLs provide a one-to-one map between an unlabeled instance and the source of supervision, making this

Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 1783–1792, Dublin, Ireland, August 23-29 2014.

```
 1: $X = \{\langle \mathbf{x}_i, y_i \rangle\}_{i=1}^{N}$ labeled tweets
 2: $U = \{\langle \mathbf{x}_i, w_i \rangle\}_{i=1}^{M}$ unlabeled tweet-website pairs
 3: $I$ iterations
 4: $k = 1000$ pool size
 5: v=train($X$) base model
 6: for $i \in I$ do
 7:     for $\langle \mathbf{x}, w \rangle \in \mathbf{pool}_k(U)$ do
 8:         $\hat{y}$=predict($\langle \mathbf{x}, w \rangle; \mathbf{v}$)
 9:         $X \leftarrow X \cup \{\langle \hat{y}, \mathbf{x} \rangle\}$
10:     end for
11:     v=train($X$)
12: end for
13: return v
```

Figure 1: Semi-supervised learning with not-so-distant supervision, i.e. tweet-website pairs $\{\langle \mathbf{x}_i, w_i \rangle\}$. SELF-TRAINING, WEB, DICT, DICT≺WEB and WEB≺DICT differ only in how **predict()** (line 8) is implemented (cf. Section 2).

less distant supervision. Note that we use linked websites only for semi-supervised learning, but do *not* require them at test time.

Our semi-supervised learning method enables us to learn POS tagging and NER models that perform more robustly across different samples of tweets than existing approaches. We consider both the scenario where a small sample of labeled Twitter data is available, and the scenario where only newswire data is available. Training on a mixture of out-of-domain (WSJ) and in-domain (Twitter) data as well as unlabeled data, we get the best reported results in the literature for both POS tagging and NER on Twitter. Our tagging models are publicly available at `https://bitbucket.org/lowlands/ttagger-nsd`

2 Tagging with not-so-distant supervision

We assume that our labeled data is highly biased by domain differences (Jiang and Zhai, 2007), population drift (Hand, 2006), or by our sample size simply being too small. To correct this bias, we want to use unlabeled Twitter data. It is well-known that semi-supervised learning algorithms such as self-training sometimes effectively correct model biases (McClosky et al., 2006; Huang et al., 2009). This paper presents an augmented self-training algorithm that corrects model bias by exploiting unlabeled data and not-so-distant supervision. More specifically, the idea is to use hyperlinks to condition tagging decisions in tweets on a richer linguistic context than what is available in the tweets. This semi-supervised approach gives state-of-the-art performance across available Twitter POS and NER data sets.

The overall semi-supervised learning algorithm is presented in Figure 1. The aim is to correct model bias by predicting tag sequences on small pools of unlabeled tweets, and re-training the model across several iterations to gradually correct model bias. Since information from hyperlinks will be important, the unlabeled data U is a corpus of tweets containing URLs. We present a baseline and four system proposals that only differ in their treatment of the **predict()** function.

In the SELF-TRAINING baseline, **predict()** corresponds to standard Viterbi inference on the unlabeled Twitter data. This means, the current model $\mathbf{v}$ is applied to the tweets by disregarding the websites in the tweet-website pairs, i.e., tagging $\mathbf{x}$ without considering w. Then the automatically tagged tweets are added to the current pool of labeled data and the procedure is iterated (line 7-11 in Figure 1).

In the WEB method, we additionally use the information from the websites. The current model $\mathbf{v}$ is used to predict tags for the pooled tweets *and* the website they linked to. For all the words that occur both in a tweet and on the corresponding website, we then project the tag most frequently assigned to those words on the website to their occurrences in the tweet. This enables us to basically condition the tag decision for each such word on its accumulated context on the website. The assumption of course being that the word in the tweet has the part-of-speech it most often has on the website linked to.

Example Here is an example of a tweet that contains a URL:

(1) #Localization #job: Supplier / Project Manager - Localisation Vendor - NY, NY, United States http://bit.ly/16KigBg #nlppeople

The words in the tweet are all common words, but they occur without linguistic context that could help a tagging model to infer whether these words are nouns, verbs, named entities, etc. However, on the website that the tweet refers to, all of these words occur in context:

(2) The Supplier/Project Manager performs the selection and maintenance ...

For illustration, the Urbana-Champaign POS tagger[1] incorrectly tags *Supplier* in (1) as an adjective. In (2), however, it gets the same word right and tags it as a noun. The tagging of (2) could potentially help us infer that *Supplier* is also a noun in (1).

Obviously, the superimposition of tags in the WEB method may change the tag of a tweet word such that it results in an unlikely tag sequence, as we will discuss later. Therefore we also implemented type-constrained decoding (Täckström et al., 2013), i.e., prune the lattice such that the tweet words observed on the website have *one of* the tags they were labeled with on the website (soft constraints), or, alternatively, were forced during decoding to have the most frequent tags they were labeled with (hard constraint decoding), thereby focusing on licensed sequences. However, none of these approaches performed significantly better than the simple WEB approach on held-out data. This suggests that sequential dependencies are less important for tagging Twitter data, which is of rather fragmented nature. Also, the WEB approach allows us to override transitional probabilities that are biased by the observations we made about the distribution of tags in our out-of-domain data.

Furthermore, we combine the not-so-distant supervision from linked websites (WEB) with supervision from dictionaries (DICT). The idea here is to exploit the fact that many word types in a dictionary are actually unambiguous, i.e., contain only a single tag. In particular, 93% of the word types in Wiktionary[2] are unambiguous. Wiktionary is a crowdsourced tag dictionary that has previously been used for minimally supervised POS tagging (Li et al., 2012; Täckström et al., 2013). In the case of NER, we use a gazetteer that combines information on PER, LOC and ORG from the KnownLists of the Illinois tagger.[3] For this gazetteer, 79% of the word types contained only a single named entity tag.

We experiment with a model that uses the dictionary only (DICT) and two ways to combine the two sources. In the former setup, the current model is first applied to tag the tweets, then any token that appears in the dictionary and is unambiguous is projected back to the tweet. The next two methods are combinations of WEB and DICT: either first project the predicted tags from the website and then, in case of conflicts, overrule predictions by the dictionary (WEB$\prec$DICT), or the other way around (DICT$\prec$WEB).

The intuition behind the idea of using linked websites as not-so-distant supervision is that while tweets are hard to analyze (even for humans) because of the limited context available in 140 character messages, tweets relate to real-world events, and Twitter users often use hyperlinks to websites to indicate what real-world events their comments address. In fact, we observed that about 20% of tweets contain URLs. The websites they link to are often newswire sites that provide more context and are written in a more canonical language, and are therefore easier to process. Our analysis of the websites can then potentially inform our analysis of the tweets. The tweets with the improved analyses can then be used to bootstrap our tagging models using a self-training mechanism. Note that our method *does not* require tweets to contain URLs at test time, but rather uses unlabeled tweets with URLs during training to build better tagging models for tweets in general. At test time, these models can be applied to any tweet.

[1] http://cogcomp.cs.illinois.edu/demo/pos/

[2] http://en.wiktionary.org/ - We used the Wiktionary version derived by Li et al. (2012).

[3] http://cogcomp.cs.illinois.edu/page/software_view/NETagger

3 Experiments

3.1 Model

In our experiments we use a publicly available implementation of conditional random fields (CRF) (Lafferty et al., 2001).[4] We use the features proposed by Gimpel et al. (2011), in particular features for word tokens, a set of features that check for the presence of hyphens, digits, single quotes, upper/lowercase, 3 character prefix and suffix information. Moreover, we add Brown word cluster features that use 2^i for $i \in 1, ..., 4$ bitstring prefixes estimated from a large Twitter corpus (Owoputi et al., 2013), which is publicly available.[5] We use a pool size of 1000 tweets. We experimented with other pool sizes {500,2000} showing similar performance. The number of iterations i is set on the development data.

For NER on websites, we use the Stanford NER system (Finkel et al., 2005)[6] with POS tags from the Lapos tagger (Tsuruoka et al., 2011).[7] For POS we found it to be superior to use the current POS model for re-tagging websites; for NER it was slightly better to use the Stanford NER tagger and thus off-line NER tagging rather than retagging the websites in every iteration.

3.2 Data

In our experiments, we consider two scenarios, sometimes referred to as unsupervised and semi-supervised domain adaptation (DA), respectively (Daumé et al., 2010; Plank, 2011). In unsupervised DA, we assume only (labeled) newswire data, in semi-supervised DA, we assume labeled data from both domains, besides unlabeled target data, but the amount of labeled target data is much smaller than the labeled source data. Most annotated corpora for English are newswire corpora. Some annotated Twitter data sets have been made available recently, described next.

	POS	**NER**
train	WSJ (700k) Gimpel-Train (Owoputi et al., 2013) (14k)	Reuter-CoNLL (Tjong Kim Sang and De Meulder, 2003) (200k) Finin-Train (Finin et al., 2010) (170k)
dev	Foster-Dev (Foster et al., 2011) (3k) Ritter-Dev (Ritter et al., 2011) (2k)	n/a n/a
test	Foster-Test (Foster et al., 2011) (2.8k) Gimpel-Test (Gimpel et al., 2011) (7k) Hovy-Test (Hovy et al., 2014)	Ritter-Test (Ritter et al., 2011) (46k) Finin-Test (Finin et al., 2010) (51k) Fromreide-Test (Fromreide et al., 2014) (20k)

Table 1: Overview of data sets. Number in parenthesis: size in number of tokens.

Training data. An overview of the different data sets is given in Table 3.2. In our experiments, we use the SANCL shared task[8] splits of the OntoNotes 4.0 distribution of the WSJ newswire annotations as newswire training data for POS tagging.[9] For NER, we use the CoNLL 2003 data sets of annotated newswire from the Reuters corpus.[10] The in-domain training POS data comes from Gimpel et al. (2011), and the in-domain NER data comes from Finin et al. (2010) (Finin-Train). These data sets are added to the newswire sets when doing semi-supervised DA. Note that for NER, we thus do not rely on expert-annotated Twitter data, but rely on crowdsourced annotations. We use MACE[11] (Hovy et al., 2013) to resolve inter-annotator conflicts between turkers (50 iterations, 10 restarts, no confidence threshold). We believe relying on crowdsourced annotations makes our set-up more robust across different samples of Twitter data.

Development and test data. We use several evaluation sets for both tasks to prevent overfitting to a specific sample. We use the (out-of-sample) development data sets from Ritter et al. (2011) and Foster

[4] http://www.chokkan.org/software/crfsuite/

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

[6] http://http://nlp.stanford.edu/software/CRF-NER.shtml

[7] http://www.logos.ic.i.u-tokyo.ac.jp/~tsuruoka/lapos/

[8] https://sites.google.com/site/sancl2012/home/shared-task

[9] LDC2011T03.

[10] http://www.clips.ua.ac.be/conll2003/ner/

[11] http://www.isi.edu/publications/licensed-sw/mace/

et al. (2011). For NER, we simply use the parameters from our POS tagging experiments and thus do not assume to have access to further development data. For both POS tagging and NER, we have three test sets. For POS tagging, the ones used in Foster et al. (2011) (FOSTER-TEST) and Ritter et al. (2011) (RITTER-TEST),[12] as well as the one presented in Hovy et al. (2014) (HOVY-TEST). For NER, we use the data set from Ritter et al. (2011) and the two data sets from Fromreide et al. (2014) as test sets. One is a manual correction of a held-out portion of FININ-TRAIN, named FININ-TEST; the other one is referred to as FROMREIDE-TEST. Since the different POS corpora use different tag sets, we map all of them corpora onto the universal POS tag set by Petrov et al. (2012). The data sets also differ in a few annotation conventions, e.g., some annotate URLs as NOUN, some as X. Moreover, our newswire tagger baselines tend to get Twitter-specific symbols such as URLs, hashtags and user accounts wrong. Instead of making annotations more consistent across data sets, we follow Ritter et al. (2011) in using a few post-processing rules to deterministically assign Twitter-specific symbols to their correct tags. The major difference between the NER data sets is whether Twitter user accounts are annotated as PER. We follow Finin et al. (2010) in doing so.

Unlabeled data We downloaded 200k tweet-website pairs from the Twitter search API over a period of one week in August 2013 by searching for tweets that contain the string *http* and downloading the content of the websites they linked to. We filter out duplicate tweets and restrict ourselves to websites that contain more than one sentence (after removing boilerplate text, scripts, HTML, etc).[13] We also require website and tweet to have at least one matching word that is not a stopword (as defined by the NLTK stopword list).[14] Finally we restrict ourselves to pairs where the website is a subsite, because website head pages tend to contain mixed content that is constantly updated. The resulting files are all tokenized using the Twokenize tool.[15] Tweets were treated as one sentence, similar to the approaches in Gimpel et al. (2011) and Owoputi et al. (2013); websites were processed by applying the Moses sentence splitter.[16]

The out-of-vocabulary (OOV) rates in Figure 2 show that in-domain training data reduces the number of unseen words considerably, especially in the NER data sets. They also suggest that some evaluation data sets share more vocabulary with our training data than others. In particular, we would expect better performance on FOSTER-TEST than on RITTER-TEST and HOVY-TEST in POS tagging, as well as better performance on FININ-TEST than on the other two NER test sets. In POS tagging, we actually do see better results with FOSTER-TEST across the board, but in NER, FININ-TEST actually turns out to be the hardest data set.

4 Results

4.1 POS results

Baselines We use three supervised CRF models as baselines (cf. the first part of Table 2). The first supervised model is trained only on WSJ. This model does very well on FOSTER-DEV and FOSTER-TEST, presumably because of the low OOV rates (Figure 2). The second supervised model is trained only on GIMPEL-TRAIN; the third on the concatenation of WSJ and GIMPEL-TRAIN. While the second baseline performs well on held-out data from its own sample (90.3% on GIMPEL-DEV), it performs poorly across our out-of-sample test and development sets. Thus, it seems to overfit the sample of tweets described in Gimpel et al. (2011). The third model trained on the concatenation of WSJ and GIMPEL-TRAIN achieves the overall best baseline performance (88.4% macro-average accuracy). We note that this is around one percentage point better than the best available off-the-shelf system for Twitter (Owoputi et al., 2013) with an average accuracy of 87.5%.

[12] Actually (Ritter et al., 2011) do cross-validation over this data, but we use the splits of Derczynski et al. (2013) for POS.

[13] Using https://github.com/miso-belica/jusText

[14] ftp://ftp.cs.cornell.edu/pub/smart/english.stop

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

[16] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/ems/support/split-sentences.perl

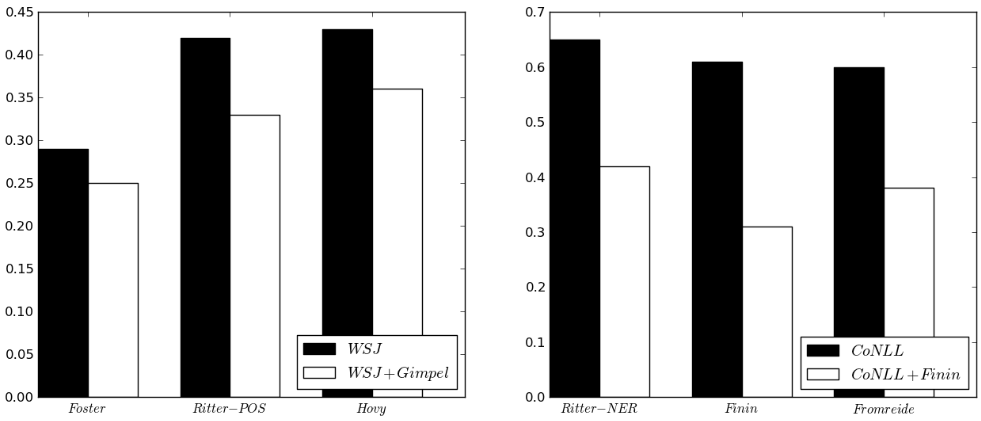

Figure 2: Test set (type-level) OOV rates for POS (left) and NER (right).

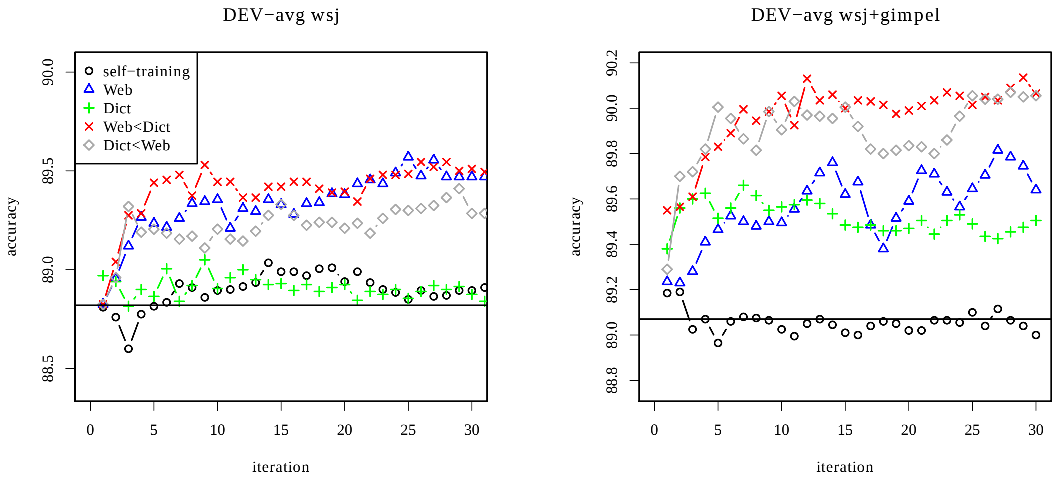

Figure 3: Learning curves on DEV-avg for systems trained on WSJ (left) and WSJ+GIMPEL (right) used to set the hyperparameter i.

Learning with URLs The results of our approaches are presented in Table 2. The hyperparameter i was set on the development data (cf. Figure 3). Note, again, that they do not require the test data to contain URLs. First of all, naive self-training does not work: accuracy declines or is just around baseline performance (Table 2 and Figure 3). In contrast, our augmented self-training methods with WEB or DICT reach large improvements. In case we assume no target training data (train on WSJ only, i.e. unsupervised DA), we obtain improvements of up to 9.1% error reduction. Overall the system improves from 88.42% to 89.07%. This also holds for the second scenario, i.e. training on WSJ+GIMPEL-TRAIN (semi-supervised DA, i.e., the case where we have some labeled target data, besides the pool of unlabeled tweets) where we reach error reductions of up to 10%. Our technique, in other words, improves the robustness of taggers, leading to much better performance on new samples of tweets.

4.2 NER results

For our NER results, cf. Table 3, we used the same feature models and parameter settings as those used for POS tagging, except conditioning also on POS information. It is conceivable that other parameter settings would have led to better results, but we did not want to assume the existence of in-domain development data for this task. Our baselines are again supervised systems, as well as off-the-shelf systems. Our in-

	Dev-avg	Test			Test-avg
		Foster	Hovy	Ritter	
Baselines trained on					
WSJ	88.82	91.87	87.01	86.38	88.42
Gimpel-Train	83.32	84.86	86.03	81.67	84.19
WSJ+Gimpel-Train	89.07	91.59	87.50	87.39	88.83
Systems trained on WSJ					
Self-training $i = 25$	85.52	91.80	86.72	85.90	88.14
Dict $i = 25$	85.61	92.08	87.63	85.68	88.46
Web $i = 25$	85.27	92.47	87.30	86.60	88.79
Dict$\prec$Web $i = 25$	86.11	**92.61**	87.70	**86.69**	89.00
Web$\prec$Dict $i = 25$	**86.15**	92.57	**88.12**	86.51	**89.07**
max err.red	4.7%	9.1%	8.6%	2.3%	4.2%
Systems trained on WSJ+Gimpel-Train					
Self-training $i = 27$	89.12	91.83	86.88	87.43	88.71
Dict $i = 27$	89.43	92.22	**88.38**	87.69	89.43
Web $i = 27$	89.82	**92.43**	87.43	88.21	89.36
Dict$\prec$Web $i = 27$	**90.04**	**92.43**	**88.38**	**88.48**	**89.76**
Web$\prec$Dict $i = 27$	**90.04**	92.40	87.99	88.39	89.59
max err.red	8.9%	10%	7.1%	8.6%	8.4%

Table 2: POS results.

house supervised baselines perform better than the available off-the-shelf systems, including the system provided by Ritter et al. (2011) (Test-avg of 54.2%). We report micro-average F$_1$-scores over entity types, computed using the publicly available evaluation script.[17] Our approaches again lead to substantial error reductions of 8–13% across our NER evaluation data sets.

	Test			Test-avg
	Ritter	Fromreide	Finin	
Baseline trained on				
CoNLL+Finin-Train	77.44	82.13	74.02	77.86
Systems trained on CoNLL+Finin-Train				
Self-training $i = 27$	**78.63**	82.88	74.89	78.80
Dict $i = 27$	65.24	69.1	65.45	66.60
Web $i = 27$	78.29	83.82	**74.99**	79.03
Dict$\prec$Web $i = 27$	78.53	**83.91**	75.83	**79.42**
Web$\prec$Dict $i = 27$	65.97	69.92	65.86	67.25
err.red	9.1%	13.3%	8.0%	9.8%

Table 3: NER results.

5 Error analysis

The majority of cases where our taggers improve on the ARK tagger (Owoputi et al., 2013) seem to relate to richer linguistic context. The ARK tagger incorrectly tags the sequence *Man Utd* as PRT-NOUN, whereas our taggers correctly predict NOUN-NOUN. In a similar vein, our taggers correctly predict the tag sequence NOUN-NOUN for *Radio Edit*, while the ARK tagger predicts NOUN-VERB.

However, some differences seem arbitrary. For example, the ARK tagger tags the sequence *Nokia*

[17] `http://www.cnts.ua.ac.be/conll2000/chunking/`

D5000 in FOSTER-TEST as NOUN-NUM. Our systems correctly predict NOUN-NOUN, but it is not clear which analysis is better in linguistic terms. Our systems predict a sequence such as *Love his version* to be VERB-PRON-NOUN, whereas the ARK tagger predicts VERB-DET-NOUN. Both choices seem linguistically motivated.

Finally, some errors are made by all systems. For example, the word *please* in *please, do that*, for example, is tagged as VERB by all systems. In FOSTER-TEST, this is annotated as X (which in the PTB style was tagged as interjection UH). Obviously, *please* often acts as a verb, and while its part-of-speech in this case may be debatable, we see *please* annotated as a verb in similar contexts in the PTB, e.g.:

(3) Please/VERB make/VERB me/PRON …

It is interesting to look at the tags that are projected from the websites to the tweets. Several of the observed projections support the intuition that coupling tweets and the websites they link to enables us to condition our tagging decisions on a richer linguistic context. Consider, for example *Salmon-Safe*, initially predicted to be a NOUN, but after projection correctly analyzed as an ADJ:

Word	Context	Initial tag	Projected tag
Salmon-Safe	… parks	NOUN	ADJ
Snohomish	… Bakery	ADJ	NOUN
toxic	ppl r …	NOUN	ADJ

One of the most frequent projections is analyzing *you're*, correctly, as a VERB rather than an ADV (if the string is not split by tokenization).

One obvious limitation of the WEB-based models is that the projections apply to all occurrences of a word. In rare cases, some words occur with different parts of speech in a single tweet, e.g., *wish* in:

(4) If I gave you one **wish** that will become true . What's your **wish** ?... ? i **wish** i'll get <num> wishes from you :p <url>

In this case, our models enforce all occurrences of *wish* to, incorrectly, be verbs.

6 Related work

Previous work on tagging tweets has assumed labeled training data (Ritter et al., 2011; Gimpel et al., 2011; Owoputi et al., 2013; Derczynski et al., 2013). Strictly supervised approaches to analyzing Twitter has the weakness that labeled data quickly becomes unrepresentative of what people write on Twitter. This paper presents results using no in-domain labeled data that are significantly better than several off-the-shelf systems, as well as results leveraging a mixture of out-of-domain and in-domain labeled data to reach new highs across several data sets.

Type-constrained POS tagging using tag dictionaries has been explored in weakly supervised settings (Li et al., 2012), as well as for cross-language learning (Das and Petrov, 2011; Täckström et al., 2013). Our type constraints in POS tagging come from tag dictionaries, but also from linked websites. The idea of using linked websites as distant supervision is similar in spirit to the idea presented in Ganchev et al. (2012) for search query tagging.

Ganchev et al. (2012), considering the problem of POS tagging search queries, tag search queries and the associated snippets provided by the search engine, projecting tags from the snippets to the queries, guided by click-through data. They do not incorporate tag dictionaries, but consider a slightly more advanced matching of snippets and search queries, giving priority to n-gram matches with larger n. Search queries contain limited contexts, like tweets, but are generally much shorter and exhibit less spelling variation than tweets.

In NER, it is common to use gazetteers, but also dictionaries as distant supervision (Kazama and Torisawa, 2007; Cucerzan, 2007). Rüd et al. (2011) consider using search engines for distant supervision of NER of search queries. Their set-up is very similar to Ganchev et al. (2012), except they do not use click-through data. They use the search engine snippets to generate feature representations rather than projections. Want et al. (2013) also use distant supervision for NER, i.e., Wikipedia page view counts, applying their model to Twitter data, but their results are considerably below the state of the art. Also, their source of supervision is not linked to the individual tweets in the way mentioned websites are.

In sum, our method is the first successful application of distant supervision to POS tagging and NER for Twitter. Moreover, it is, to the best of our knowledge, the first paper that addresses both problems using the same technique. Finally, our results are significantly better than state-of-the-art results in both POS tagging and NER.

7 Conclusion

We presented a semi-supervised approach to POS tagging and NER for Twitter data that uses dictionaries and linked websites as a source of not-so-distant (or linked) supervision to guide the bootstrapping. Our approach outperforms off-the-shelf taggers when evaluated across various data sets, achieving average error reductions across data sets of 5% on POS tagging and 10% on NER over state-of-the-art baselines.

References

Timothy Baldwin, Paul Cook, Marco Lui, Andrew MacKinlay, and Li Wang. 2013. How noisy social media text, how diffrnt social media sources? In *IJCNLP*.

Edward Benson, Aria Haghighi, and Regina Barzilay. 2011. Event discovery in social media feeds. In *ACL*.

Silvia Cucerzan. 2007. Large-scale named entity disambiguation based on wikipedia data. In *EMNLP-CoNLL*.

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

Hal Daumé, Abhishek Kumar, and Avishek Saha. 2010. Frustratingly easy semi-supervised domain adaptation. In *ACL Workshop on Domain Adaptation for NLP*.

Leon Derczynski, Alan Ritter, Sam Clark, and Kalina Bontcheva. 2013. Twitter part-of-speech tagging for all: overcoming sparse and noisy data. In *RANLP*.

Jacob Eisenstein. 2013. What to do about bad language on the internet. In *NAACL*.

Tim Finin, Will Murnane, Anand Karandikar, Nicholas Keller, Justin Martineau, and Mark Dredze. 2010. Annotating named entities in Twitter data with crowdsourcing. In *NAACL-HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*.

Jenny Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *ACL*.

Jennifer Foster, Ozlem Cetinoglu, Joachim Wagner, Josef Le Roux, Joakim Nivre, Deirde Hogan, and Josef van Genabith. 2011. From news to comments: Resources and benchmarks for parsing the language of Web 2.0. In *IJCNLP*.

Hege Fromreide, Dirk Hovy, and Anders Søgaard. 2014. Crowdsourcing and annotating ner for twitter #drift. In *LREC*.

Kuzman Ganchev, Keith Hall, Ryan McDonald, and Slav Petrov. 2012. Using search-logs to improve query tagging. In *ACL*.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, 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*.

David Hand. 2006. Classifier technology and illusion of progress. *Statistical Science*, 21(1):1–15.

Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard Hovy. 2013. Learning whom to trust with MACE. In *NAACL*.

Dirk Hovy, Barbara Plank, and Anders Søgaard. 2014. When pos datasets don't add up: Combatting sample bias. In *LREC*.

Zhongqiang Huang, Mary Harper, and Slav Petrov. 2009. Self-training with products of latent variable grammars. In *EMNLP*.

Jing Jiang and ChengXiang Zhai. 2007. Instance weighting for domain adaptation in NLP. In *ACL*.

Long Jiang, Mo Yo, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent Twitter sentiment classification. In *ACL*.

Jun'ichi Kazama and Kentaro Torisawa. 2007. Exploiting wikipedia as external knowledge for named entity recognition. In *EMNLP-CoNLL*.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *ICML*.

Shen Li, João Graça, and Ben Taskar. 2012. Wiki-ly supervised part-of-speech tagging. In *EMNLP*.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *HLT-NAACL*.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *ACL*.

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 *NAACL*.

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

Sasa Petrovic, Miles Osborne, and Victor Lavrenko. 2010. Streaming first story detection with application to Twitter. In *NAACL*.

Barbara Plank. 2011. *Domain Adaptation for Parsing*. Ph.D. thesis, University of Groningen.

Alan Ritter, Sam Clark, Oren Etzioni, et al. 2011. Named entity recognition in tweets: an experimental study. In *EMNLP*.

Stefan Rüd, Massimiliano Ciaramita, Jens Müller, and Hinrich Schütze. 2011. Piggyback: Using search engines for robust cross-domain named entity recognition. In *ACL*.

Oscar Täckström, Dipanjan Das, Slav Petrov, Ryan McDonald, and Joakim Nivre. 2013. Token and type constraints for cross-lingual part-of-speech tagging. *TACL*, 1:1–12.

Erik F Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *In CoNLL*.

Yoshimasa Tsuruoka, Yusuke Miyao, and Jun'ichi Kazama. 2011. Learning with lookahead: can history-based models rival globally optimized models? In *CoNLL*.

Chun-Kai Wang, Bo-June Hsu, Ming-Wei Chang, and Emre Kiciman. 2013. Simple and knowledge-intensive generative model for named entity recognition. Technical report, Microsoft Research.