Twitter Part-of-Speech Tagging for All: Overcoming Sparse and Noisy Data

Leon Derczynski
University of Sheffield
leon@dcs.shef.ac.uk

Alan Ritter
University of Washington
aritter@cs.washington.edu

Sam Clark
University of Washington
ssclark@cs.washington.edu

Kalina Bontcheva
University of Sheffield
kalina@dcs.shef.ac.uk

Abstract

Part-of-speech information is a pre-requisite in many NLP algorithms. However, Twitter text is difficult to part-of-speech tag: it is noisy, with linguistic errors and idiosyncratic style. We present a detailed error analysis of existing taggers, motivating a series of tagger augmentations which are demonstrated to improve performance. We identify and evaluate techniques for improving English part-of-speech tagging performance in this genre.

Further, we present a novel approach to system combination for the case where available taggers use different tagsets, based on vote-constrained bootstrapping with unlabeled data. Coupled with assigning prior probabilities to some tokens and handling of unknown words and slang, we reach 88.7% tagging accuracy (90.5% on development data). This is a new high in PTB-compatible tweet part-of-speech tagging, reducing token error by 26.8% and sentence error by 12.2%. The model, training data and tools are made available.

1 Introduction

Twitter provides a wealth of uncurated text. The site has over 200 million users active each month (O'Carroll, 2012) generating messages at a peak rate over 230 000 per minute (Ashtari, 2013). Information found on Twitter has already been shown to be useful for a variety of applications (e.g. monitoring earthquakes (Sakaki et al., 2010) and predicting flu (Culotta, 2010)). However, the lack of quality part-of-speech taggers tailored specifically to this emerging genre impairs the accuracy of key downstream NLP techniques (e.g. named entity recognition, term extraction), and by extension, overall application results.

Microblog text (from e.g. Twitter) is characterised by: short messages; inclusion of URIs; username mentions; topic markers; and threaded conversations. It often presents colloquial content containing abbreviations and errors. Some of these phenomena comprise linguistic noise, which when coupled with message brevity (140 characters for "tweets") and the lack of labeled corpora, make microblog part-of-speech tagging very challenging. Alongside the genre's informal nature, such limits encourage "compressed" utterances, with authors omitting not only needless words but also those with grammatical or contextualising function.

Part-of-speech tagging is a central problem in natural language processing, and a key step early in manly NLP pipelines. Machine learning-based part-of-speech (PoS) taggers can exploit labeled training data to adapt to new genres or even languages, through supervised learning. Algorithm sophistication apart, the performance of these taggers is reliant upon the quantity and quality of available training data. Consequently, lacking large PoS-annotated resources and faced with prevalent noise, state-of-the-art PoS taggers perform poorly on microblog text (Derczynski et al., 2013), with error rates up to ten times higher than on newswire (see Section 3).

To address these issues, we propose a data-intensive approach to microblog part-of-speech tagging for English, which overcomes data sparsity by using the thousands of unlabeled tweets created every minute, coupled with techniques to smooth out genre-specific noise. To reduce the impact of data sparsity, we introduce a new method for *vote-constrained* bootstrapping, evaluated in the context of PoS tagging. Further, we introduce methods for handling the genre's characteristic errors and slang, and evaluate the performance impact of adjusting prior tag probabilities of unambiguous tokens.

1. A comprehensive comparative evaluation of existing POS taggers on tweet datasets is carried out (Section 3), followed by a detailed analysis and classification of common errors (Section 4), including errors due to tokenisation, slang, out-of-vocabulary, and spelling.
2. Address tweet noisiness through handling of rare words (Section 5.1) and adjusting prior tag probabilities of unambiguous tokens, using external knowledge (Section 5.2).
3. Investigate vote-constrained bootstrapping on a large corpus of unlabeled tweets, to create needed tweet-genre training data (Section 5.3).
4. Demonstrate that these techniques reduce token-level error by 26.8% and sentence-level error by 12.2% (Section 6).

Proceedings of Recent Advances in Natural Language Processing, pages 198–206,
Hissar, Bulgaria, 7-13 September 2013.

Tagger	Known	Unknown	Overall	Sentence
TnT	96.76%	85.86%	96.46%	-
SVMTool	97.39%	89.01%	97.16%	-
TBL	-	-	93.67%	-
Stanford	-	90.46%	**97.28%**	56.79%

Table 1: Token-level labeling accuracy for four off-the-shelf PoS taggers on newswire. Not all these performance measures are supplied in the literature.

Tagger	T-dev		D-dev	
	Token	Sentence	Token	Sentence
TnT	71.50%	1.69%	77.52%	14.87%
SVMTool	**74.84%**	**4.24%**	82.92%	**22.68%**
TBL	70.52%	2.54%	76.22%	11.52%
Stanford	73.37%	1.67%	**83.29%**	22.22%

Table 2: Token tagging performance of WSJ-trained taggers (sections 0-18) on Twitter data. Figures listed are the proportion of tokens labeled with the correct part-of-speech tag, and the proportion of sentences in which all tokens were correctly labeled.

2 Related Work

Regarding Twitter part-of-speech tagging, the two most similar earlier papers introduce the ARK tagger (Gimpel et al., 2011) and T-Pos (Ritter et al., 2011). Both these approaches adopt clustering to handle linguistic noise, and train from a mixture of hand-annotated tweets and existing PoS-labeled data. The ARK tagger[1] reaches 92.8% accuracy at token level but uses a coarser, custom tagset. T-Pos[2] is based on the Penn Treebank set and, in its evaluation, achieves an 88.4% token tagging accuracy. Neither report sentence/whole-tweet accuracy rates. Foster et al. (2011) introduce results for both PoS tagging and parsing, but do not present a tool, and focus more on the parsing aspect.

Previous work on part-of-speech tagging in noisy environments has focused on either dealing with noisy tokens either by using a lexicon that can handle partial matches through e.g. topic models (Darling et al., 2012) or Brown clustering (Clark, 2003), or by applying extra processing steps to correct/bias tagger performance, e.g., post-/pre-processing respectively (Gadde et al., 2011). Finally, classic work on bootstrapped PoS tagging is that of Clark et al. (2003), who use a co-training approach to improve tagger performance using unlabeled data.

3 Comparing taggers on Twitter data

In order to evaluate a new tagging approach, we must first have a good idea of the current performance of state-of-the art tools, and a common basis (e.g. corpus and tagset) for comparison.

3.1 Conventional Part-of-speech Taggers

To quantify the disadvantage conventional PoS taggers have when faced with microblog text, we evaluate state-of-the-art taggers against Twitter data. We used the same training and evaluation data for each tagger, re-training taggers where required.

When measuring the performance of taggers, as per popular convention we report the overall proportion of tags that are accurately assigned. Where possible we report performance on "unknown" words – those that do not occur in the training data. Further, as per Manning (2011) we report the rate of getting whole sentences right, since "a single bad mistake in a sentence can greatly throw off the usefulness of a tagger to downstream tasks".[3]

We evaluated four state-of-the-art trainable and publicly available PoS taggers that used the Penn Treebank tagsettrereetagger: SVMTool (Giménez and Marquez, 2004), the Stanford Tagger (Toutanova et al., 2003), TnT (Brants, 2000) and a transformation-based learning (TBL) tagger (Brill, 1995) supported by sequential n-gram backoff. The NLTK implementations of TnT and TBL were used (Bird et al., 2009). The 'left3words' model was used with the Stanford tagger, and 'M0' with SVMTool. For initial comparison, taggers were tested on standard newswire text from the Penn Treebank (Marcus et al., 1993),[4] training with Wall Street Journal (WSJ) sections 0-18 and evaluating on sections 19-21. The base performance for each tagger is given in Table 1.

3.2 Labeled Tweet Corpora

Three PoS-labeled microblog datasets are currently available. The T-Pos corpus of 15K tokens introduced by Ritter et al. (2011) uses a tagset based on the Penn Treebank tagset,plus four new tags for URLs (`URL`), hashtags (`HT`), username mentions (`USR`) and retweet signifiers (`RT`). The DCU dataset of 14K tokens (Foster et al., 2011) is also based on the Penn Treebank (**PTB**) set, but does not have the same new tags as T-Pos, and uses slightly different tokenisation. The ARK corpus of 39K tokens (Gimpel et al., 2011) uses a novel tagset, which, while suitable for the microblog genre, is somewhat less descriptive than the PTB sets on many points. For example, its `V` tag corresponds to any verb, conflating PTB's `VB`, `VBD`, `VBG`, `VBN`, `VBP`, `VBZ`, and `MD` tags. Intuitively, this seems to be a simpler tagging task, and performance using it reaches 92.8% (Owoputi et al., 2012).

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

[2] https://github.com/aritter/twitter_nlp

[3] In fact, as sentence boundaries are at best unclear in many tweets, we use a slightly stricter interpretation of "sentence" and only count entire tweets that are labeled correctly.

[4] LDC corpus reference LDC99T42

Tagger	T-dev		D-dev	
	Token	Sentence	Token	Sentence
TnT	79.17%	5.08%	80.05%	16.73%
SVMTool	77.70%	4.24%	78.22%	11.15%
TBL	78.64%	**8.47%**	79.02%	13.75%
Stanford	**83.14%**	6.78%	**84.19%**	**24.07%**
T-Pos	83.85%	10.17%	84.96%	27.88%

Table 3: Performance of taggers trained on a WSJ/IRC/Twitter (T-train) corpus. T-Pos is the only tagger with Twitter-specific customisations.

Although it is possible to transduce data labeled using the T-Pos or PTB tagsets to the ARK tagset, the reverse is not true. We built a tagger using the T-Pos tagset. This choice was motivated by the tagset's PTB compatibility, the volume of existing tools which rely on a PTB-like tagging schema, and the fact that labeling microtext using this more complex tagset is not vastly more difficult than with the ARK tagset (e.g. Ritter et al. (2011))

The following datasets were used in our study. We shuffled and then split the T-Pos data 70:15:15 into training, development and evaluation sets named T-train, T-dev and T-eval. Splits are made at whole-tweet level. For comparability, we mapped the DCU development and evaluation datasets (D-dev and D-eval) into the T-Pos tokenisation and tagset schema.

Some near-genre corpora are available. For example, resources are available of IRCtext and SMS text (Almeida et al., 2011). Of these, only one is annotated for part-of-speech tags – the NPS IRC corpus (Forsyth and Martell, 2007) – which we use.

3.3 Performance Comparison

For training data composition, we approximate Ritter's approach. We use 50K tokens from the Wall Street Journal part of the Penn Treebank (**WSJ**), 32K tokens from the NPS IRC corpus, and T-train (2.3K tokens). We vary in that we have a fixed split of Twitter data, where earlier work did four-way cross-validation.

The first experiment was to evaluate the performance of the news-trained taggers described in Section 3.1 on two tweet corpora: T-dev and D-dev. As shown in Table 2, performance on tweets is poor and, in some cases, absolute token accuracy is 20% lower than with newswire (Table 1). This comparison is somewhat unfair as not all labels in the test set are seen in the training data. Combining training data of 10K tokens of tweets, 10K tokens of a genre similar to tweets (IRC) and 50K tokens of non-tweets (newswire) is fairer; performance of taggers trained on this dataset is given in Table 3. All taggers performed better against T-dev after having T-train and the IRC data included in their training data (e.g. from 73.37% to 83.14% for the Stanford tagger), showing the impact of tweet-genre training data.

However, the improvements are much less impressive on D-dev, which is a completely different corpus. There, e.g. Stanford improves only from 83.29% on WSJ to 84.19%. Candid analysis suggests that the DCU corpus contains less noisy utterances, with better grammatical consistency and fewer orthographic errors.

Training data	Token	Sentence	No. tokens
WSJ	73.37%	1.67%	50K
IRC	70.03%	2.54%	36K
WSJ+IRC	78.37%	5.08%	86K
Twitter (T-train)	78.19%	6.78%	10K
IRC+Twitter	79.75%	**8.47%**	46K
WSJ+Twitter	82.11%	**8.47%**	60K
All three	**83.14%**	6.78%	96K

Table 4: Performance of Stanford tagger over the development dataset T-dev using a combination of three genres of training data.

Category	Count	Proportion
GS error	6	6.7%
IV	24	27.0%
Pre-taggable	7	9.0%
Proper noun	10	11.2%
Slang	24	27.0%
Tokenisation	8	9.0%
Twitter-specific	2	2.2%
Typo	7	7.9%
Total Result	89	

Table 5: Categorisation of mis-tagged unknown words.

Based on its strong performance, we concentrate on the Stanford tagger for the remainder of this paper. Using this, we measured the impact that tweet and tweet-like training data have on PoS tagging accuracy. As shown in Table 4, the newswire-only trained Stanford tagger performed worst, with IRC (a tweet-like genre) training data yielding some improvement and tweet-genre data having greatest effect.

4 Error analysis

We investigated errors made on words not in the training lexicon (**unknown** words). For the basic Stanford tagger model trained using WSJ+IRC+Twitter (T-train), the tagging accuracy on known tokens (e.g. those in the training lexicon) is 83.14%, and 38.56% on unknown words. One approach for improving overall accuracy is to better handle unknown words.

Tagging of unknown words forces the tagger to rely on contextual clues. Errors on these words make up a large part of the mis-tagged tokens. One can see the effect that improving accuracy on unknown words has on overall performance by comparing, for example, the Stanford tagger when trained on non-tweet vs. tweet data in Table 4. We identified the unknown words that were tagged incorrectly and categorised them into eight groups.

Gold standard error – Where the ground truth data is wrong. For example, the Dutch *dank je* should in an English corpus be tagged as foreign words (`FW`), but in our dataset is marked *dank/URL je/IN*. These are not tagger errors but rather evaluation errors, avoided by

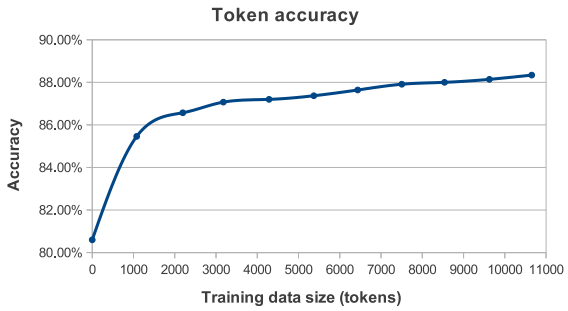

Figure 1: Stanford tagger token-level accuracy on T-dev with increasing amounts of microblog training text.

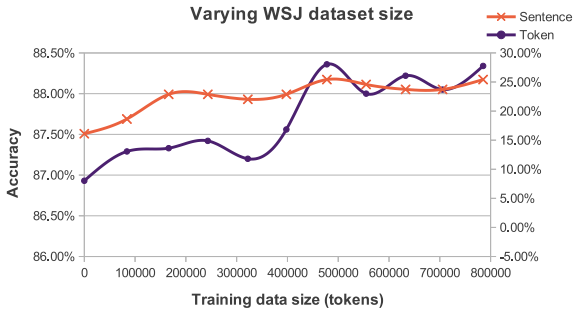

Figure 2: Token-level performance on T-dev with varying amounts of WSJ text, in addition to T-train and IRC data.

repairing the ground truth.

In-vocabulary – Tokens that are common in general, but do not occur in the training data. For example, *Internet* and *bake* are unknown words and mis-tagged in the evaluation corpus. This kind of error may be fixed by a larger training set or the use of a lexicon, especially for monosemous words.

Pre-taggable – Words to which a label may be reliably assigned automatically. This group includes well-formed URLs, hash tags and smileys.

Proper noun – Proper nouns not in the training data. Most of these should be tagged NNP, and are often useful for later named entity recognition. Incorrectly tagged proper nouns often had incorrect capitalisation; for example, *derek* and *birmingham*. Gazetteer approaches may help annotate these, in cases of words that can only occur as proper nouns.

Slang – An abundance of slang is a characteristic feature of microblog text, and these words are often incorrectly tagged, as well as being rarely seen due to a proliferation of spelling variations (all incorrect). Examples include *LUVZ*, *HELLA* and *2night*. Some kind of automatic correction or expanded lexicon could be employed to either map these back to dictionary words or to include previously-seen spelling variations.

Tokenisation error – Occasionally the tokeniser or original author makes tokenisation errors. Examples include *ass**sneezes*, which should have been split into more than one token as indicated by special/punctuation characters, and *eventhough*, where the author has missed a space. These are hard to correct. Specific subtypes of error, such as the joined words in the example, could be checked for and forcibly fixed, though this requires distinguishing intentional from unintentional word usage.

Genre-specific – Words that are unique to specific sites, often created for microblog usage, such as *unfollowing*. Extra tweet-genre-specific training data may to reduce genre-specific word errors.

Orthographic error – Finally, although it is difficult to detect the intent of the user, some content seems likely to have been accidentally mis-spelled. Examples include *Handle]* and *suprising*. Automatic spelling correction may improve performance in these cases.

We also examined the impact the volume of training data had on performance. Figure 1 shows a continuing performance increase as ground-truth tweets are added, suggesting more tweet-genre training data will yield improvements. Conversely, there is already enough newswire-type training data and adding more is unlikely to greatly increase performance (Figure 2). Consequently, subsequent experiments do not include more newswire beyond the 50K-token WSJ corpus excerpt also used in T-Pos.

5 Addressing Noise and Data Sparseness

Our examination of frequent PoS tagging errors identified some readily rectifiable classes of problem. These were: slang, jargon and common mis-spellings; genre-related phrases; smileys; and unambiguous named entities. In addition, observations suggested that more tweet training data would help. Thus, we augmented our approach in three ways: improved handling of unknown and slang words; conversion of unambiguous tags into token prior probabilities; and addition of semi-supervised training data.

5.1 Normalisation for Unknown Words

Tagging accuracy on tokens not seen in the training data (out-of-vocabulary, or **OOV** tokens) is lower than that on those previously encountered (see Table 1). Consequently, reducing the proportion of unknown words is likely to improve performance. Informal error analysis suggested that slang makes up a notable proportion of the unknown word set. To provide in-vocabulary (**IV**) versions of slang words (i.e. to normalise them), we created a set of mappings from OOV words to their IV equivalents, using slang dictionaries and manual examination of the training data. The mapping is applied to text before it is tagged, and the original token is labeled with a PoS tag based on the mapped (normalised) word.

Many texts contain erroneous or slang tokens, which can be mapped to in-lexicon versions of themselves via *normalisation*. A critical normalisation subtask is

Features	Token	Sent.
Baseline[7]	83.14%	6.78%
Word shape features[8]	87.91%	22.88%
As above, excl. company suffixes	88.34%	**25.42%**
Low common word threshold[9]	88.36%	**25.42%**
Low common & rare word thresh.[10]	**88.49%**	**25.42%**

Table 6: Impact of introduction of word shape features, as token accuracy on T-dev.

distinguishing previously-unseen but correctly spelled words (such as proper nouns) from those with orthographic anomalies. Anomalous tokens are those with unusual orthography, either intentional (e.g. slang) or unintentional (e.g. typos). Slang words account for a large proportion of mislabeled unknowns (Table 5).

Normalisation is a difficult task and current approaches are complex (Kaufmann and Kalita, 2010; Han and Baldwin, 2011; Liu et al., 2012). Rather than apply sophisticated word clustering or multi-stage normalisation, we took a data-driven approach to investigating and then handling problematic tokens.

Setup In our data, a small subset of orthographic errors and otherwise-unusual words account for a large part of the total anomalous words. We use a lookup list (derived from unknown words in the training corpus) to map these to more common forms, e.g. *luv→love* and *hella→very*.[5] This lookup list is based upon both external slang gazetteers and observations over T-train.

To supplement this knowledge-based approach, we enable and fine-tune unknown-word handling features of the Stanford tagger. The tagger contains highly-configurable feature generation options for handling unknown words. These extra **rare word features** accounted for information such as word shape, word length and so on.[6] Their inclusion should increase the amount of unknown word handling information in the final model. Results are given in Table 6.

We also tuned the rare word thresholds for our corpus, changing the threshold for inclusion of a token's rare word features. We tried values from zero to 20 in steps of 1; per-token performance peaked at 88.49% for $rarewordthreshold = 3$. It slowly declined for higher values up to 700 (tested in larger steps). This modest improvement indicates value in optimising the rare word threshold.

Unknown Handling Results Thus, we were able to increase part-of-speech tagging performance in three ways: by adapting the idea of normalisation and implementing it with both fixed word-lists (repairing all but 20% of problem tokens), with extra features encoding word shapes to handle OOV terms, and with a more sensitive threshold to inclusion of rare words in the model.

[5] An intensifier, from the original *"one hell of a ..."*.

[6] http://nlp.stanford.edu/nlp/javadoc/javanlp/edu/stanford/nlp/tagger/maxent/ExtractorFramesRare.html

[8] The tagger's `naacl2003unk` feature set

[9] veryCommonWordThresh = 40

[10] veryCommonWordThresh = 40, rarewordthreshold = 3

Entity pre-labeled	Token
Baseline	88.49%
Slang	88.76%
Named Entities	88.71%
Smileys	88.54%
Genre-specific	88.58%
All	**89.07%**
Error reduction	5.03%

Table 7: Impact of prior labeling and mapping slang to IV terms on T-dev; rare word threshold is 3.

5.2 Tagging from External Knowledge

It is possible to constrain the possible set of sentence labelings by pre-assigning probability distributions to tokens for which there is an unambiguous tag. In these cases, the distribution is just $P(t_{correct}) = 1.0$. This strategy not only improves accuracy on these tokens, but also reduces uncertainty regarding the set of potential sentence taggings.

For example, in a simplified HMM bigram tagging scenario, one has a sequence of words $w_0, w_1..w_n$ having corresponding tags $t_0, t_1..t_n$, and is concerned with emission distributions $P(w_i|t_i)$ and tag transition probabilities $P(t_i|t_{i-1})$. Knowing $P(t_i)$ for one word affects all subsequent tag distributions. As the tagger is typically used in a bidirectional mode (effectively adding reverse transition probabilities $P(t_i|t_{i+1})$), using prior knowledge to inform labels reduces tagging uncertainty over the whole sentence.

Setup In the above error analysis, off-the-shelf taggers made errors on some Twitter-specific phenomena. Some errors on tokens where the four tweet-specific labels `URL`, `USR`, `RT` and `HT` apply can be reliably and automatically prevented by using regular expression patterns to detect pertinent tokens.

A second category of mistakes was smileys (aka emoticons), of which the most frequent can be labeled `UH` unambiguously using a look-up list. Some flexibility is required to capture smiley variations, e.g. −__− vs. −____− (Park et al., 2013), which was implemented again with high-accuracy regular expressions.

Proper noun errors (`NN`/`NNP`) were relatively common – an observation also made by Ritter et al. (2011). It is possible to recognise unambiguous named entities (i.e. words that only ever occur as `NNP`) using external knowledge sources, such as a gazetteer list or an entity database. In this case, we used GATE's ANNIE gazetteer lists of personal first-names and cities (Cunningham et al., 2002) and, in addition, a manually constructed list of corporation and website names frequently mentioned in the training data (e.g. *YouTube*, *Toyota*). Terms were excluded from the latter list if their PoS tag is ambiguous (e.g. *google* may occur as a proper noun or verb and so is not included).

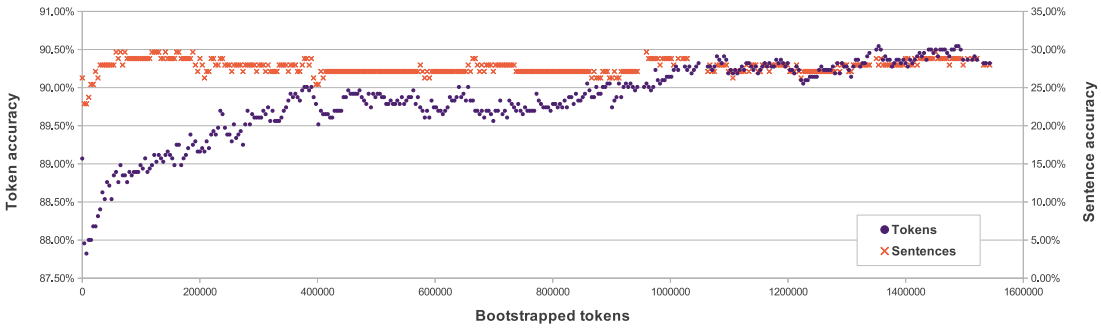

Figure 3: Bootstrapping the tagger using data with vote-constrained labelings.

Tagging with Priors Results In our experiments, the tagger was adapted to take prior probabilities into account, and experiments run using a model trained on WSJ+IRC+T-train that includes the noise-handling augmentations described in Section 5.1. Table 7 shows the performance difference on each of the four categories of token discussed above. Each has an effect, combining to yield a 5.0% error reduction ($P<0.005$, McNemar's test). When the original model detailed in Table 3 is used, token performance improves from 83.14% accuracy to 86.93%. Assignment of priors affords 22.5% error reduction in this scenario. We compare fixing the tag *before* tagging the rest of the sentence, with tagging the whole sentence and overwriting such tokens' tags. While the latter only affects unambiguous tokens, the former affects the other tags in the sentence during tagging, via e.g. transmission probabilities and window features. This is a novel adaptation of this tagger. To compare, when correcting this model's labels *post*-tagging, error reduction is only 19.0% (to 86.34%).

5.3 Vote-constrained Bootstrapping

Having seen the impact that tweet data has on performance, one choice is to increase the amount of labeled training tweets. We have only a small amount of ground-truth, labeled data. However, large amounts of unlabeled data are readily accessible; a day's discourse on Twitter comprises 500 million tweets of unlabeled data (Terdiman, 2012). In this scenario, one option is bootstrapping (Goldman and Zhou, 2000; Cucerzan and Yarowsky, 2002).

In bootstrapping, the training data is bolstered using semi-supervised data, a "pool" of examples not human curated but labeled automatically. To maintain high data quality, one should only admit to the pool instances in which there is a high confidence. We propose vote-constrained bootstrapping as bootstrapping where not all participating systems (or classifiers) use the same class label inventory. This allows different approaches to the same task to be combined into an ensemble. It is less strict than classic voting, because although both approaches constrain the set of labels that are seen in agreement with each other, classic voting constrains this maximally, to a 1:1 mapping.

In this scenario, equivalence classes are determined for class labels assigned by systems. Matches occur when all outputs are in the same class, thus only constraining the set of agreeing votes. This permits the constraint of valid responses through voting. The caveat is that at least one voting classifier must use the same class inventory as the eventual trained classifier. Given unlabeled data, the method is for each system to perform feature extraction and then classifications of instances. For instances where all classifiers assign a label in the same equivalence class, the instance may be admitted to the pool, using whichever class label is that belonging to the eventual output system.

In this instances, our approach is to use T-Pos and the ARK tagger to create semi-supervised data. We used a single tokeniser based on the T-Pos tokenisation scheme (PTB but catering for Twitter specific phenomena such as hashtags). To label the unlabeled data with maximum accuracy, we combined the two taggers, which are trained on different data with different features and different tagsets. The ARK tagger uses a tagset that is generally more coarse-grained than that of T-Pos, and so instead of requiring direct matches between the two taggers' output, the ARK labelings *constrain* the set of tags that could be considered a match.

To increase fidelity of data added to the pool, for PoS-tagging, we add a further criterion to the vote-constraint requirement. We define high-confidence instances as those from the tweets where the T-Pos labelings fit within the ARK tagger output's constraints on every token.

Setup We gathered unlabeled data directly from Twitter using the "garden hose" (a streaming 10% sample of global messages). Tweets were collected, automatically filtered to remove non-English tweets using the language identification of Preotiuc-Pietro et al. (2012), tokenised, and then labeled using both taggers. The labelings were compared using manually-predefined equivalence classes, and if consistent for the whole tweet, the tweet-specific tags re-labeled using regular expressions (see Section 5.2) and the T-Pos tagset labeled tweet added to the pool.

Tagger	T-eval		D-eval	
	Token	Sentence	Token	Sentence
T-Pos (Ritter et al., 2011)	84.55%	9.32%	84.83%	28.00%
Our Augmented tagger	**88.69%**	**20.34%**	**89.37%**	**36.80%**
Error reduction	26.80%	12.15%	29.93%	12.22%

Table 8: Performance of our augmented tagger on the held-out evaluation data. ER is error reduction.

Vote-constraint results We set out to From our unlabeled data, taggers reached agreement on 19.2% of tweets. This reduced an initial capture of 832 135 English tweets (9 523 514 tokens) to 159 492 tweets with agreed PoS labelings (1 542 942 tokens). To see how confident we can be in taggings generated with this method, we checked accuracy of agreed tweets on T-dev. When tested on the T-dev dataset, the taggers agreed on 17.8% of tweets (accounting for 15.2% of tokens). Of the labelings agreed upon over T-dev, these were correct for 97.4% of tokens (71.3% of sentences).

After an initial dip, adding bootstrapped training data gave a performance increase. Figure 3 shows the benefit of using vote-constrained bootsrapping, giving **90.54% token accuracy** (28.81% for sentences) on T-dev after seeing 1.5M training tokens. The shape of the curve suggests potential benefit from even more bootstrapping data.

6 Results

We set out to improve part-of-speech tagging on tweets, using the full, rich Penn Treebank set. We made a series of improvements based on observed difficulties with microblog tagging, including the introduction of a bootstrapping technique using labelers that have different tag sets.

Based on our augmentations, we evaluated against the held-out evaluation sets T-eval and D-eval. Results are in Table 8, comparing with T-Pos (the other taggers are far behind as to not warrant direct comparison). Significance is at $P<0.01$ using the McNemar (1947) test with Yates' continuity correction.

Note that we use different evaluation splits in this paper compared to that used in the original T-Pos work. In this paper, training data and evaluation data are always the same across compared systems.

The augmentations offered significant improvements, which can be both extended (in terms of bootstrapping data, prior-probability lists and slang lists) as well as readily distributed independent of platform. The performance on the development set is even higher, reaching over 90.5% tagging accuracy. Both these tagging accuracies are significantly above anything previously reached on the Penn Treebank tagset. Critically, the large gains in sentence-level accuracy offer significant improvements for real world applications.

Regarding limits to this particular approach, the technique is likely sensitive to annotator errors given the size of the initial data, and probably limited by inter-annotator agreement. We have partially quantified the linguistic noise this genre presents, but it is still a significant problem – unknown word tagging does not reach nearly as high performance as on e.g. newswire. Finally, the wide variation in forms of expression (possibly encouraged by message length limits) may reduce the frequency of otherwise common phrases, making data harder to generalise over.

7 Conclusion

Twitter is a text source that offers much, but is difficult to process, partially due to linguistic noise. Additionally, existing approaches suffer from insufficient labeled training data. We introduced approaches for overcoming this noise, for taking advantage of genre-specific structure in tweets, and for generating data through heterogeneous taggers. These combined to provide a readily-distributable and improved part of speech tagger for twitter. Our techniques led to significant reductions in error rate, not only at the token but also at sentence level, and the creation of a 1.5 million token corpus of high-confidence PoS-labeled tweets.

Resources Presented – Our twitter part-of-speech tagger is available in four forms. First, as a standalone Java program, including handling of slang and prior probabilities. Second, a plugin for the popular language processing framework, GATE (Cunningham et al., 2013). Third, a model for the Stanford tagger, distributed as a single file, for use in existing applications. Finally, a high-speed model that trades about 2% accuracy for doubled pace. We also provide the bootstrapped corpus and its vote-constraint based creation tool, allowing replication of our results and the construction of new taggers with this large, high-confidence dataset.

This tagger is now part of the GATE TwitIE toolkit for processing social media text (Bontcheva et al., 2013). The tagger and datasets are also distributed via the GATE wiki, at:

`http://gate.ac.uk/wiki/twitter-postagger.html`

Acknowledgments

This work was partially supported by funding from UK EPSRC grants EP/K017896/1 the CHIST-ERA uComp project (www.ucomp.eu) and EP/I004327/1. The authors would also like to thank John Bauer of Stanford University for his kind assistance with the Stanford Tagger, and Jennifer Foster of Dublin City University for her generous help with extra labeled data. Finally, the first author thanks Aarhus University for their facilities support.

References

T. Almeida, J. Hidalgo, and A. Yamakami. 2011. Contributions to the study of SMS spam filtering: new collection and results. *Proceedings of the 11th ACM symposium on Document engineering*, pages 259–62.

O. Ashtari. 2013. The super tweets of #sb47. http://blog.twitter.com/2013/02/the-super-tweets-of-sb47.html.

S. Bird, E. Loper, and E. Klein. 2009. *Natural Language Processing with Python*. O'Reilly Media Inc.

K. Bontcheva, L. Derczynski, A. Funk, M. A. Greenwood, D. Maynard, and N. Aswani. 2013. TwitIE: A Fully-featured Information Extraction Pipeline for Microblog Text. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*. Association for Computational Linguistics.

T. Brants. 2000. TnT: a statistical part-of-speech tagger. In *Proceedings of the sixth conference on Applied Natural Language Processing*, pages 224–231. ACL.

E. Brill. 1995. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4):543–565.

S. Clark, J. Curran, and M. Osborne. 2003. Bootstrapping PoS taggers using unlabelled data. In *Proceedings of the seventh Conference on Natural Language Learning*, pages 49–55. ACL.

A. Clark. 2003. Combining distributional and morphological information for part of speech induction. In *Proceedings of the tenth conference of the European chapter of the Association for Computational Linguistics*, pages 59–66. ACL.

S. Cucerzan and D. Yarowsky. 2002. Bootstrapping a multilingual part-of-speech tagger in one person-day. In *Proceedings of the 6th Conference on Natural Language Learning*, pages 1–7. ACL.

A. Culotta. 2010. Towards detecting influenza epidemics by analyzing twitter messages. In *Proceedings of the First Workshop on Social Media Analytics*, pages 115–122. ACM.

H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. 2002. GATE: an Architecture for Development of Robust HLT Applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*. ACL.

H. Cunningham, V. Tablan, A. Roberts, and K. Bontcheva. 2013. Getting more out of biomedical documents with gate's full lifecycle open source text analytics. *PLoS Computational Biology*, 9(2):e1002854, 02.

W. Darling, M. Paul, and F. Song. 2012. Unsupervised part-of-speech tagging in noisy and esoteric domains with a syntactic-semantic Bayesian HMM. In *Proceedings of the conference of the European chapter of the Association for Computational Linguistics*, pages 1–9. ACL.

L. Derczynski, D. Maynard, N. Aswani, and K. Bontcheva. 2013. Microblog-Genre Noise and Impact on Semantic Annotation Accuracy. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media*. ACM.

E. Forsyth and C. Martell. 2007. Lexical and discourse analysis of online chat dialog. In *International Conference on Semantic Computing*, pages 19–26. IEEE.

J. Foster, O. Cetinoglu, J. Wagner, J. Le Roux, S. Hogan, J. Nivre, D. Hogan, and J. van Genabith. 2011. #hardtoparse: POS Tagging and Parsing the Twitterverse. In *Proceedings of the AAAI Workshop on Analyzing Microtext*.

P. Gadde, L. Subramaniam, and T. Faruquie. 2011. Adapting a wsj trained part-of-speech tagger to noisy text: preliminary results. In *Proceedings of the Joint Workshop on Multilingual OCR and Analytics for Noisy Unstructured Text Data*, pages 5:1–5:8. ACM.

J. Giménez and L. Marquez. 2004. SVMTool: A general POS tagger generator based on Support Vector Machines. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*.

K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, and N. Smith. 2011. Part-of-speech tagging for twitter: annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 42–47. ACL.

S. Goldman and Y. Zhou. 2000. Enhancing supervised learning with unlabeled data. In *Proceedings of the International Conference on Machine Learning*, pages 327–334.

B. Han and T. Baldwin. 2011. Lexical normalisation of short text messages: Makn sens a# twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 368–378.

M. Kaufmann and J. Kalita. 2010. Syntactic normalization of twitter messages. In *International Conference on Natural Language Processing*.

F. Liu, F. Weng, and X. Jiang. 2012. A broadcoverage normalization system for social media language. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*.

C. Manning. 2011. Part-of-speech tagging from 97% to 100%: is it time for some linguistics? *Computational Linguistics and Intelligent Text Processing*, pages 171–189.

M. Marcus, M. Marcinkiewicz, and B. Santorini. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Q. McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.

L. O'Carroll. 2012. Twitter active users pass 200 million. http://www.guardian.co.uk/technology/2012/dec/18/twitter-users-pass-200-million.

O. Owoputi, B. O'Connor, C. Dyer, K. Gimpel, and N. Schneider. 2012. Part-of-Speech Tagging for Twitter: Word Clusters and Other Advances. Technical Report CMU-ML-12-107, Machine Learning Department, Carnegie Mellon University.

J. Park, V. Barash, C. Fink, and M. Cha. 2013. Emoticon Style: Interpreting Differences in Emoticons Across Cultures. In *Proceedings of the Seventh International AAAI Conference on Weblogs and Social Media*, pages 466–475. AAAI Press.

D. Preotiuc-Pietro, S. Samangooei, T. Cohn, N. Gibbins, and M. Niranjan. 2012. Trendminer: An architecture for real time analysis of social media text. In *Proceedings of the workshop on Real-Time Analysis and Mining of Social Streams*.

A. Ritter, S. Clark, O. Etzioni, et al. 2011. Named entity recognition in tweets: an experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534. ACL.

T. Sakaki, M. Okazaki, and Y. Matsuo. 2010. Earthquake shakes Twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World Wide Web (WWW)*, pages 851–860. ACM.

D. Terdiman. 2012. Report: Twitter hits half a billion tweets a day. http://news.cnet.com/8301-1023_3-57541566-93/report-twitter-hits-half-a-billion-tweets-a-day/.

K. Toutanova, D. Klein, C. Manning, and Y. Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 173–180. ACL.