Named Entity Recognition in Tweets: An Experimental Study

Alan Ritter, Sam Clark, Mausam and Oren Etzioni
Computer Science and Engineering
University of Washington
Seattle, WA 98125, USA
{aritter,ssclark,mausam,etzioni}@cs.washington.edu

Abstract

People tweet more than 100 Million times daily, yielding a noisy, informal, but sometimes informative corpus of 140-character messages that mirrors the zeitgeist in an unprecedented manner. The performance of standard NLP tools is severely degraded on tweets. This paper addresses this issue by re-building the NLP pipeline beginning with part-of-speech tagging, through chunking, to named-entity recognition. Our novel T-NER system doubles F_1 score compared with the Stanford NER system. T-NER leverages the redundancy inherent in tweets to achieve this performance, using LabeledLDA to exploit Freebase dictionaries as a source of distant supervision. LabeledLDA outperforms co-training, increasing F_1 by 25% over ten common entity types.

Our NLP tools are available at: http://github.com/aritter/twitter_nlp

1 Introduction

Status Messages posted on Social Media websites such as Facebook and Twitter present a new and challenging style of text for language technology due to their noisy and informal nature. Like SMS (Kobus et al., 2008), tweets are particularly terse and difficult (See Table 1). Yet tweets provide a unique compilation of information that is more up-to-date and inclusive than news articles, due to the low-barrier to tweeting, and the proliferation of mobile devices.[1] The corpus of tweets already exceeds the size of the Library of Congress (Hachman, 2011) and is growing far more rapidly. Due to the volume of tweets, it is natural to consider named-entity recognition, information extraction, and text mining over tweets. Not surprisingly, the performance of "off the shelf" NLP tools, which were trained on news corpora, is weak on tweet corpora.

In response, we report on a re-trained "NLP pipeline" that leverages previously-tagged out-of-domain text, [2] tagged tweets, and unlabeled tweets to achieve more effective part-of-speech tagging, chunking, and named-entity recognition.

1	The Hobbit has FINALLY started filming! I cannot wait!
2	Yess! Yess! Its official Nintendo announced today that they Will release the Nintendo 3DS in north America march 27 for $250
3	Government confirms blast n nuclear plants n japan...don't knw wht s gona happen nw...

Table 1: Examples of noisy text in tweets.

We find that classifying named entities in tweets is a difficult task for two reasons. First, tweets contain a plethora of distinctive named entity types (Companies, Products, Bands, Movies, and more). Almost all these types (except for People and Locations) are relatively infrequent, so even a large sample of manually annotated tweets will contain few training examples. Secondly, due to Twitter's 140 character limit, tweets often lack sufficient context to determine an entity's type without the aid of background

[1] See the "trending topics" displayed on twitter.com

[2] Although tweets can be written on any subject, following convention we use the term "domain" to include text styles or genres such as Twitter, News or IRC Chat.

knowledge.

To address these issues we propose a distantly supervised approach which applies LabeledLDA (Ramage et al., 2009) to leverage large amounts of unlabeled data in addition to large dictionaries of entities gathered from Freebase, and combines information about an entity's context across its mentions.

We make the following contributions:

1. We experimentally evaluate the performance of off-the-shelf news trained NLP tools when applied to Twitter. For example POS tagging accuracy drops from about 0.97 on news to 0.80 on tweets. By utilizing in-domain, out-of-domain, and unlabeled data we are able to substantially boost performance, for example obtaining a 52% increase in F_1 score on segmenting named entities.

2. We introduce a novel approach to *distant supervision* (Mintz et al., 2009) using Topic Models. LabeledLDA is applied, utilizing constraints based on an open-domain database (Freebase) as a source of supervision. This approach increases F_1 score by 25% relative to co-training (Blum and Mitchell, 1998; Yarowsky, 1995) on the task of classifying named entities in Tweets.

The rest of the paper is organized as follows. We successively build the NLP pipeline for Twitter feeds in Sections 2 and 3. We first present our approaches to shallow syntax – part of speech tagging (§2.1), and shallow parsing (§2.2). §2.3 describes a novel classifier that predicts the informativeness of capitalization in a tweet. All tools in §2 are used as features for named entity segmentation in §3.1. Next, we present our algorithms and evaluation for entity classification (§3.2). We describe related work in §4 and conclude in §5.

2 Shallow Syntax in Tweets

We first study two fundamental NLP tasks – POS tagging and noun-phrase chunking. We also discuss a novel capitalization classifier in §2.3. The outputs of all these classifiers are used in feature generation for named entity recognition in the next section.

For all experiments in this section we use a dataset of 800 randomly sampled tweets. All results (Tables 2, 4 and 5) represent 4-fold cross-validation experiments on the respective tasks.[3]

	Accuracy	Error Reduction
Majority Baseline (NN)	0.189	-
Word's Most Frequent Tag	0.760	-
Stanford POS Tagger	0.801	-
T-POS(PTB)	0.813	6%
T-POS(Twitter)	0.853	26%
T-POS(IRC + PTB)	0.869	34%
T-POS(IRC + Twitter)	0.870	35%
T-POS(PTB + Twitter)	0.873	36%
T-POS(PTB + IRC + Twitter)	0.883	41%

Table 2: POS tagging performance on tweets. By training on in-domain labeled data, in addition to annotated IRC chat data, we obtain a 41% reduction in error over the Stanford POS tagger.

2.1 Part of Speech Tagging

Part of speech tagging is applicable to a wide range of NLP tasks including named entity segmentation and information extraction.

Prior experiments have suggested that POS tagging has a very strong baseline: assign each word to its most frequent tag and assign each Out of Vocabulary (OOV) word the most common POS tag. This baseline obtained a 0.9 accuracy on the Brown corpus (Charniak et al., 1993). However, the application of a similar baseline on tweets (see Table 2) obtains a much weaker 0.76, exposing the challenging nature of Twitter data.

A key reason for this drop in accuracy is that Twitter contains far more OOV words than grammatical text. Many of these OOV words come from spelling variation, *e.g.*, the use of the word "n" for "in" in Table 1 example 3. Although NNP is the most frequent tag for OOV words, only about 1/3 are NNPs.

The performance of off-the-shelf news-trained POS taggers also suffers on Twitter data. The state-of-the-art Stanford POS tagger (Toutanova et al., 2003) improves on the baseline, obtaining an accuracy of 0.8. This performance is impressive given that its training data, the Penn Treebank WSJ (PTB), is so different in style from Twitter, however it is a huge drop from the 97% accuracy reported on the

[3] We used Brendan O'Connor's Twitter tokenizer

Gold	Predicted	Stanford Error	T-POS Error	Error Reduction
NN	NNP	0.102	0.072	29%
UH	NN	0.387	0.047	88%
VB	NN	0.071	0.032	55%
NNP	NN	0.130	0.125	4%
UH	NNP	0.200	0.036	82%

Table 3: Most common errors made by the Stanford POS Tagger on tweets. For each case we list the fraction of times the gold tag is misclassified as the predicted for both our system and the Stanford POS tagger. All verbs are collapsed into VB for compactness.

PTB. There are several reasons for this drop in performance. Table 3 lists common errors made by the Stanford tagger. First, due to unreliable capitalization, common nouns are often misclassified as proper nouns, and vice versa. Also, interjections and verbs are frequently misclassified as nouns. In addition to differences in vocabulary, the grammar of tweets is quite different from edited news text. For instance, tweets often start with a verb (where the subject 'I' is implied), as in: "watchng american dad."

To overcome these differences in style and vocabulary, we manually annotated a set of 800 tweets (16K tokens) with tags from the Penn TreeBank tag set for use as in-domain training data for our POS tagging system, T-POS.[4] We add new tags for the Twitter specific phenomena: retweets, @usernames, #hashtags, and urls. Note that words in these categories can be tagged with 100% accuracy using simple regular expressions. To ensure fair comparison in Table 2, we include a postprocessing step which tags these words appropriately for all systems.

To help address the issue of OOV words and lexical variations, we perform clustering to group together words which are distributionally similar (Brown et al., 1992; Turian et al., 2010). In particular, we perform hierarchical clustering using Jcluster (Goodman, 2001) on 52 million tweets; each word is uniquely represented by a bit string based on the path from the root of the resulting hierarchy to the word's leaf. We use the Brown clusters resulting from prefixes of 4, 8, and 12 bits. These clusters are often effective in capturing lexical variations, for example, following are lexical variations on the word "tomorrow" from one cluster after filtering out other words (most of which refer to days):

> '2m', '2ma', '2mar', '2mara', '2maro', '2marrow', '2mor', '2mora', '2moro', '2morow', '2morr', '2morro', '2morrow', '2moz', '2mr', '2mro', '2mrrw', '2mrw', '2mw', 'tmmrw', 'tmo', 'tmoro', 'tmorrow', 'tmoz', 'tmr', 'tmro', 'tmrow', 'tmrrow', 'tmrrw', 'tmrw', 'tmrww', 'tmw', 'tomaro', 'tomarow', 'tomarro', 'tomarrow', 'tomm', 'tommarow', 'tommarrow', 'tommoro', 'tommorow', 'tommorrow', 'tommorw', 'tommrow', 'tomo', 'tomolo', 'tomoro', 'tomorow', 'tomorro', 'tomorrw', 'tomoz', 'tomrw', 'tomz'

T-POS uses Conditional Random Fields[5] (Lafferty et al., 2001), both because of their ability to model strong dependencies between adjacent POS tags, and also to make use of highly correlated features (for example a word's identity in addition to prefixes and suffixes). Besides employing the Brown clusters computed above, we use a fairly standard set of features that include POS dictionaries, spelling and contextual features.

On a 4-fold cross validation over 800 tweets, T-POS outperforms the Stanford tagger, obtaining a 26% reduction in error. In addition we include 40K tokens of annotated IRC chat data (Forsythand and Martell, 2007), which is similar in style. Like Twitter, IRC data contains many misspelled/abbreviated words, and also more pronouns, and interjections, but fewer determiners than news. Finally, we also leverage 50K POS-labeled tokens from the Penn Treebank (Marcus et al., 1994).

Overall T-POS trained on 102K tokens (12K from Twitter, 40K from IRC and 50K from PTB) results in a 41% error reduction over the Stanford tagger, obtaining an accuracy of 0.883. Table 3 lists gains on some of the most common error types, for example, T-POS dramatically reduces error on interjections and verbs that are incorrectly classified as nouns by the Stanford tagger.

2.2 Shallow Parsing

Shallow parsing, or chunking is the task of identifying non-recursive phrases, such as noun phrases,

[4] Using MMAX2 (Müller and Strube, 2006) for annotation.

[5] We use MALLET (McCallum, 2002).

	Accuracy	Error Reduction
Majority Baseline (B-NP)	0.266	-
OpenNLP	0.839	-
T-CHUNK(CoNLL)	0.854	9%
T-CHUNK(Twitter)	0.867	17%
T-CHUNK(CoNLL + Twitter)	0.875	22%

Table 4: Token-Level accuracy at shallow parsing tweets. We compare against the OpenNLP chunker as a baseline.

verb phrases, and prepositional phrases in text. Accurate shallow parsing of tweets could benefit several applications such as Information Extraction and Named Entity Recognition.

Off the shelf shallow parsers perform noticeably worse on tweets, motivating us again to annotate in-domain training data. We annotate the same set of 800 tweets mentioned previously with tags from the CoNLL shared task (Tjong Kim Sang and Buchholz, 2000). We use the set of shallow parsing features described by Sha and Pereira (2003), in addition to the Brown clusters mentioned above. Part-of-speech tag features are extracted based on cross-validation output predicted by T-POS. For inference and learning, again we use Conditional Random Fields. We utilize 16K tokens of in-domain training data (using cross validation), in addition to 210K tokens of newswire text from the CoNLL dataset.

Table 4 reports T-CHUNK's performance at shallow parsing of tweets. We compare against the off-the shelf OpenNLP chunker[6], obtaining a 22% reduction in error.

2.3 Capitalization

A key orthographic feature for recognizing named entities is capitalization (Florian, 2002; Downey et al., 2007). Unfortunately in tweets, capitalization is much less reliable than in edited texts. In addition, there is a wide variety in the styles of capitalization. In some tweets capitalization is informative, whereas in other cases, non-entity words are capitalized simply for emphasis. Some tweets contain all lowercase words (8%), whereas others are in ALL CAPS (0.6%).

To address this issue, it is helpful to incorporate information based on the entire content of the message to determine whether or not its capitalization is informative. To this end, we build a capitalization classifier, T-CAP, which predicts whether or not a tweet is informatively capitalized. Its output is used as a feature for Named Entity Recognition. We manually labeled our 800 tweet corpus as having either "informative" or "uninformative" capitalization. The criteria we use for labeling is as follows: if a tweet contains any non-entity words which are capitalized, but do not begin a sentence, or it contains any entities which are not capitalized, then its capitalization is "uninformative", otherwise it is "informative".

	P	R	F_1
Majority Baseline	0.70	1.00	0.82
T-CAP	0.77	0.98	0.86

Table 5: Performance at predicting reliable capitalization.

For learning , we use Support Vector Machines.[7] The features used include: the fraction of words in the tweet which are capitalized, the fraction which appear in a dictionary of frequently lowercase/capitalized words but are not lowercase/capitalized in the tweet, the number of times the word 'I' appears lowercase and whether or not the first word in the tweet is capitalized. Results comparing against the majority baseline, which predicts capitalization is always informative, are shown in Table 5. Additionally, in §3 we show that features based on our capitalization classifier improve performance at named entity segmentation.

3 Named Entity Recognition

We now discuss our approach to named entity recognition on Twitter data. As with POS tagging and shallow parsing, off the shelf named-entity recognizers perform poorly on tweets. For example, applying the Stanford Named Entity Recognizer to one of the examples from Table 1 results in the following output:

> $[\text{Yess}]_{ORG}$! $[\text{Yess}]_{ORG}$! Its official $[\text{Nintendo}]_{LOC}$ announced today that they Will release the $[\text{Nintendo}]_{ORG}$ 3DS in north $[\text{America}]_{LOC}$ march 27 for $250

[6] http://incubator.apache.org/opennlp/

[7] http://www.chasen.org/~taku/software/TinySVM/

The OOV word 'Yess' is mistaken as a named entity. In addition, although the first occurrence of 'Nintendo' is correctly segmented, it is misclassified, whereas the second occurrence is improperly segmented – it should be the product "Nintendo 3DS". Finally "north America" should be segmented as a *LOCATION*, rather than just 'America'. In general, news-trained Named Entity Recognizers seem to rely heavily on capitalization, which we know to be unreliable in tweets.

Following Collins and Singer (1999), Downey et al. (2007) and Elsner et al. (2009), we treat classification and segmentation of named entities as separate tasks. This allows us to more easily apply techniques better suited towards each task. For example, we are able to use discriminative methods for named entity segmentation and distantly supervised approaches for classification. While it might be beneficial to jointly model segmentation and (distantly supervised) classification using a joint sequence labeling and topic model similar to that proposed by Sauper et al. (2010), we leave this for potential future work.

Because most words found in tweets are not part of an entity, we need a larger annotated dataset to effectively learn a model of named entities. We therefore use a randomly sampled set of 2,400 tweets for NER. All experiments (Tables 6, 8-10) report results using 4-fold cross validation.

3.1 Segmenting Named Entities

Because capitalization in Twitter is less informative than news, in-domain data is needed to train models which rely less heavily on capitalization, and also are able to utilize features provided by T-CAP.

We exhaustively annotated our set of 2,400 tweets (34K tokens) with named entities.[8] A convention on Twitter is to refer to other users using the @ symbol followed by their unique username. We deliberately choose not to annotate @usernames as entities in our data set because they are both unambiguous, and trivial to identify with 100% accuracy using a simple regular expression, and would only serve to inflate our performance statistics. While there is ambiguity as to the type of @usernames (for example, they can refer to people or companies), we believe they could be more easily classified using features of their associated user's profile than contextual features of the text.

	P	R	F_1	F_1 inc.
Stanford NER	0.62	0.35	0.44	-
T-SEG(None)	0.71	0.57	0.63	43%
T-SEG(T-POS)	0.70	0.60	0.65	48%
T-SEG(T-POS, T-CHUNK)	0.71	0.61	0.66	50%
T-SEG(All Features)	0.73	0.61	0.67	52%

Table 6: Performance at segmenting entities varying the features used. "None" removes POS, Chunk, and capitalization features. Overall we obtain a 52% improvement in F_1 score over the Stanford Named Entity Recognizer.

T-SEG models Named Entity Segmentation as a sequence-labeling task using IOB encoding for representing segmentations (each word either begins, is inside, or is outside of a named entity), and uses Conditional Random Fields for learning and inference. Again we include orthographic, contextual and dictionary features; our dictionaries included a set of type lists gathered from Freebase. In addition, we use the Brown clusters and outputs of T-POS, T-CHUNK and T-CAP in generating features.

We report results at segmenting named entities in Table 6. Compared with the state-of-the-art news-trained Stanford Named Entity Recognizer (Finkel et al., 2005), T-SEG obtains a 52% increase in F_1 score.

3.2 Classifying Named Entities

Because Twitter contains many distinctive, and infrequent entity types, gathering sufficient training data for named entity classification is a difficult task. In any random sample of tweets, many types will only occur a few times. Moreover, due to their terse nature, individual tweets often do not contain enough context to determine the type of the entities they contain. For example, consider following tweet:

KKTNY in 45min..........

without any prior knowledge, there is not enough context to determine what type of entity "KKTNY" refers to, however by exploiting redundancy in the data (Downey et al., 2010), we can determine it is likely a reference to a television show since it of-

[8] We found that including out-of-domain training data from the MUC competitions lowered performance at this task.

ten co-occurs with words such as *watching* and *premieres* in other contexts.[9]

In order to handle the problem of many infrequent types, we leverage large lists of entities and their types gathered from an open-domain ontology (Freebase) as a source of distant supervision, allowing use of large amounts of unlabeled data in learning.

Freebase Baseline: Although Freebase has very broad coverage, simply looking up entities and their types is inadequate for classifying named entities in context (0.38 F-score, §3.2.1). For example, according to Freebase, the mention 'China' could refer to a country, a band, a person, or a film. This problem is very common: 35% of the entities in our data appear in more than one of our (mutually exclusive) Freebase dictionaries. Additionally, 30% of entities mentioned on Twitter do not appear in any Freebase dictionary, as they are either too new (for example a newly released videogame), or are misspelled or abbreviated (for example 'mbp' is often used to refer to the "mac book pro").

Distant Supervision with Topic Models: To model unlabeled entities and their possible types, we apply LabeledLDA (Ramage et al., 2009), constraining each entity's distribution over topics based on its set of possible types according to Freebase. In contrast to previous weakly supervised approaches to Named Entity Classification, for example the Co-Training and Naïve Bayes (EM) models of Collins and Singer (1999), LabeledLDA models each entity string as a mixture of types rather than using a single hidden variable to represent the type of each mention. This allows information about an entity's distribution over types to be shared across mentions, naturally handling ambiguous entity strings whose mentions could refer to different types.

Each entity string in our data is associated with a bag of words found within a context window around all of its mentions, and also within the entity itself. As in standard LDA (Blei et al., 2003), each bag of words is associated with a distribution over topics, Multinomial(θ_e), and each topic is associated with a distribution over words, Multinomial(β_t). In addition, there is a one-to-one mapping between topics and Freebase type dictionaries. These dictionaries constrain θ_e, the distribution over topics for each entity string, based on its set of possible types, $FB[e]$. For example, θ_{Amazon} could correspond to a distribution over two types: *COMPANY*, and *LOCATION*, whereas θ_{Apple} might represent a distribution over *COMPANY*, and *FOOD*. For entities which aren't found in any of the Freebase dictionaries, we leave their topic distributions θ_e unconstrained. Note that in absence of any constraints LabeledLDA reduces to standard LDA, and a fully unsupervised setting similar to that presented by Elsner et. al. (2009).

In detail, the generative process that models our data for Named Entity Classification is as follows:

```
for each type: $t = 1 \ldots T$ do
    Generate $\beta_t$ according to symmetric Dirichlet distribution Dir($\eta$).
end for
for each entity string $e = 1 \ldots |E|$ do
    Generate $\theta_e$ over $FB[e]$ according to Dirichlet distribution Dir($\alpha_{FB[e]}$).
    for each word position $i = 1 \ldots N_e$ do
        Generate $z_{e,i}$ from Mult($\theta_e$).
        Generate the word $w_{e,i}$ from Mult($\beta_{z_{e,i}}$).
    end for
end for
```

To infer values for the hidden variables, we apply Collapsed Gibbs sampling (Griffiths and Steyvers, 2004), where parameters are integrated out, and the $z_{e,i}$s are sampled directly.

In making predictions, we found it beneficial to consider θ_e^{train} as a prior distribution over types for entities which were encountered during training. In practice this sharing of information across contexts is very beneficial as there is often insufficient evidence in an isolated tweet to determine an entity's type. For entities which weren't encountered during training, we instead use a prior based on the distribution of types across all entities. One approach to classifying entities in context is to assume that θ_e^{train} is fixed, and that all of the words inside the entity mention and context, $\mathbf{w}$, are drawn based on a single topic, z, that is they are all drawn from Multinomial(β_z). We can then compute the posterior distribution over types in closed form with a simple application of Bayes rule:

$$P(z|\mathbf{w}) \quad \propto \quad \prod_{w \in \mathbf{w}} P(w|z:\beta)P(z:\theta_e^{\text{train}})$$

During development, however, we found that rather than making these assumptions, using Gibbs Sam-

[9] Kourtney & Kim Take New York.

Type	Top 20 Entities not found in Freebase dictionaries
PRODUCT	nintendo ds lite, apple ipod, generation black, ipod nano, apple iphone, gb black, xperia, ipods, verizon media, mac app store, kde, hd video, nokia n8, ipads, iphone/ipod, galaxy tab, samsung galaxy, playstation portable, nintendo ds, vpn
TV-SHOW	pretty little, american skins, nof, order svu, greys, kktny, rhobh, parks & recreation, parks & rec, dawson 's creek, big fat gypsy weddings, big fat gypsy wedding, winter wipeout, jersey shores, idiot abroad, royle, jerseyshore, mr . sunshine, hawaii five-0, new jersey shore
FACILITY	voodoo lounge, grand ballroom, crash mansion, sullivan hall, memorial union, rogers arena, rockwood music hall, amway center, el mocambo, madison square, bridgestone arena, cat club, le poisson rouge, bryant park, mandalay bay, broadway bar, ritz carlton, mgm grand, olympia theatre, consol energy center

Table 7: Example type lists produced by LabeledLDA. No entities which are shown were found in Freebase; these are typically either too new to have been added, or are misspelled/abbreviated (for example rhobh="Real Housewives of Beverly Hills"). In a few cases there are segmentation errors.

pling to estimate the posterior distribution over types performs slightly better. In order to make predictions, for each entity we use an informative Dirichlet prior based on θ_e^{train} and perform 100 iterations of Gibbs Sampling holding the hidden topic variables in the training data fixed (Yao et al., 2009). Fewer iterations are needed than in training since the type-word distributions, β have already been inferred.

3.2.1 Classification Experiments

To evaluate T-CLASS's ability to classify entity mentions in context, we annotated the 2,400 tweets with 10 types which are both popular on Twitter, and have good coverage in Freebase: *PERSON*, *GEO-LOCATION*, *COMPANY*, *PRODUCT*, *FACILITY*, *TV-SHOW*, *MOVIE*, *SPORTSTEAM*, *BAND*, and *OTHER*. Note that these type annotations are only used for evaluation purposes, and not used during training T-CLASS, which relies only on distant supervision. In some cases, we combine multiple Freebase types to create a dictionary of entities representing a single type (for example the *COMPANY* dictionary contains Freebase types */business/consumer_company* and */business/brand*). Because our approach does not rely on any manually labeled examples, it is straightforward to extend it for a different sets of types based on the needs of downstream applications.

Training: To gather unlabeled data for inference, we run T-SEG, our entity segmenter (from §3.1), on 60M tweets, and keep the entities which appear 100 or more times. This results in a set of 23,651 distinct entity strings. For each entity string, we collect words occurring in a context window of 3 words from all mentions in our data, and use a vocabulary of the 100K most frequent words. We run Gibbs sampling for 1,000 iterations, using the last sample to estimate entity-type distributions θ_e, in addition to type-word distributions β_t. Table 7 displays the 20 entities (not found in Freebase) whose posterior distribution θ_e assigns highest probability to selected types.

Results: Table 8 presents the classification results of T-CLASS compared against a majority baseline which simply picks the most frequent class (*PERSON*), in addition to the Freebase baseline, which only makes predictions if an entity appears in exactly one dictionary (*i.e.*, appears unambiguous). T-CLASS also outperforms a simple supervised baseline which applies a MaxEnt classifier using 4-fold cross validation over the 1,450 entities which were annotated for testing. Additionally we compare against the co-training algorithm of Collins and Singer (1999) which also leverages unlabeled data and uses our Freebase type lists; for seed rules we use the "unambiguous" Freebase entities. Our results demonstrate that T-CLASS outperforms the baselines and achieves a 25% increase in F_1 score over co-training.

Tables 9 and 10 present a breakdown of F_1 scores by type, both collapsing types into the standard classes used in the MUC competitions (*PERSON*, *LOCATION*, *ORGANIZATION*), and using the 10 popular Twitter types described earlier.

Entity Strings *vs.* Entity Mentions: DL-Cotrain and LabeledLDA use two different representations for the unlabeled data during learning. LabeledLDA groups together words across all mentions of an en-

System	P	R	F_1
Majority Baseline	0.30	0.30	0.30
Freebase Baseline	0.85	0.24	0.38
Supervised Baseline	0.45	0.44	0.45
DL-Cotrain	0.54	0.51	0.53
LabeledLDA	0.72	0.60	0.66

Table 8: Named Entity Classification performance on the 10 types. Assumes segmentation is given as in (Collins and Singer, 1999), and (Elsner et al., 2009).

Type	LL	FB	CT	SP	N
PERSON	0.82	0.48	0.65	0.83	436
LOCATION	0.74	0.21	0.55	0.67	372
ORGANIZATION	0.66	0.52	0.55	0.31	319
overall	0.75	0.39	0.59	0.49	1127

Table 9: F_1 classification scores for the 3 MUC types *PERSON*, *LOCATION*, *ORGANIZATION*. Results are shown using LabeledLDA (LL), Freebase Baseline (FB), DL-Cotrain (CT) and Supervised Baseline (SP). N is the number of entities in the test set.

Type	LL	FB	CT	SP	N
PERSON	0.82	0.48	0.65	0.86	436
GEO-LOC	0.77	0.23	0.60	0.51	269
COMPANY	0.71	0.66	0.50	0.29	162
FACILITY	0.37	0.07	0.14	0.34	103
PRODUCT	0.53	0.34	0.40	0.07	91
BAND	0.44	0.40	0.42	0.01	54
SPORTSTEAM	0.53	0.11	0.27	0.06	51
MOVIE	0.54	0.65	0.54	0.05	34
TV-SHOW	0.59	0.31	0.43	0.01	31
OTHER	0.52	0.14	0.40	0.23	219
overall	0.66	0.38	0.53	0.45	1450

Table 10: F_1 scores for classification broken down by type for LabeledLDA (LL), Freebase Baseline (FB), DL-Cotrain (CT) and Supervised Baseline (SP). N is the number of entities in the test set.

	P	R	F_1
DL-Cotrain-entity	0.47	0.45	0.46
DL-Cotrain-mention	0.54	0.51	**0.53**
LabeledLDA-entity	0.73	0.60	**0.66**
LabeledLDA-mention	0.57	0.52	0.54

Table 11: Comparing LabeledLDA and DL-Cotrain grouping unlabeled data by entities vs. mentions.

System	P	R	F_1
COTRAIN-NER (10 types)	0.55	0.33	0.41
T-NER(10 types)	0.65	0.42	**0.51**
COTRAIN-NER (PLO)	0.57	0.42	0.49
T-NER(PLO)	0.73	0.49	**0.59**
Stanford NER (PLO)	0.30	0.27	0.29

Table 12: Performance at predicting both segmentation and classification. Systems labeled with PLO are evaluated on the 3 MUC types *PERSON*, *LOCATION*, *ORGANIZATION*.

tity string, and infers a distribution over its possible types, whereas DL-Cotrain considers the entity mentions separately as unlabeled examples and predicts a type independently for each. In order to ensure that the difference in performance between LabeledLDA and DL-Cotrain is not simply due to this difference in representation, we compare both DL-Cotrain and LabeledLDA using both unlabeled datasets (grouping words by all mentions vs. keeping mentions separate) in Table 11. As expected, DL-Cotrain performs poorly when the unlabeled examples group mentions; this makes sense, since Co-Training uses a discriminative learning algorithm, so when trained on entities and tested on individual mentions, the performance decreases. Additionally, LabeledLDA's performance is poorer when considering mentions as "documents". This is likely due to the fact that there isn't enough context to effectively learn topics when the "documents" are very short (typically fewer than 10 words).

End to End System: Finally we present the end to end performance on segmentation and classification (T-NER) in Table 12. We observe that T-NER again outperforms co-training. Moreover, comparing against the Stanford Named Entity Recognizer on the 3 MUC types, T-NER doubles F_1 score.

4 Related Work

There has been relatively little previous work on building NLP tools for Twitter or similar text styles. Locke and Martin (2009) train a classifier to recognize named entities based on annotated Twitter data, handling the types *PERSON*, *LOCATION*, and *ORGANIZATION*. Developed in parallel to our work, Liu et al. (2011) investigate NER on the same 3 types, in addition to *PRODUCT*s and present a semi-supervised approach using k-nearest neighbor. Also developed in parallel, Gimpell et al. (2011) build a POS tagger for tweets using 20 coarse-grained tags. Benson et. al. (2011) present a system which extracts artists and venues associated with musical performances. Recent work (Han and Baldwin, 2011; Gouws et al., 2011) has proposed lexical normalization of tweets which may be useful as a preprocessing step for the upstream tasks like POS tagging and NER. In addition Finin et. al. (2010) investigate the use of Amazon's Mechanical Turk for annotating Named Entities in Twitter, Minkov et. al. (2005) investigate person name recognizers in email, and Singh et. al. (2010) apply a minimally supervised approach to extracting entities from text advertisements.

In contrast to previous work, we have demonstrated the utility of features based on Twitter-specific POS taggers and Shallow Parsers in segmenting Named Entities. In addition we take a distantly supervised approach to Named Entity Classification which exploits large dictionaries of entities gathered from Freebase, requires no manually annotated data, and as a result is able to handle a larger number of types than previous work. Although we found manually annotated data to be very beneficial for named entity segmentation, we were motivated to explore approaches that don't rely on manual labels for classification due to Twitter's wide range of named entity types. Additionally, unlike previous work on NER in informal text, our approach allows the sharing of information across an entity's mentions which is quite beneficial due to Twitter's terse nature.

Previous work on Semantic Bootstrapping has taken a weakly-supervised approach to classifying named entities based on large amounts of unlabeled text (Etzioni et al., 2005; Carlson et al., 2010; Kozareva and Hovy, 2010; Talukdar and Pereira, 2010; McIntosh, 2010). In contrast, rather than predicting which classes an entity belongs to (e.g. a multi-label classification task), LabeledLDA estimates a *distribution* over its types, which is then useful as a prior when classifying mentions in context.

In addition there has been been work on Skip-Chain CRFs (Sutton, 2004; Finkel et al., 2005) which enforce consistency when classifying multiple occurrences of an entity within a document. Using topic models (e.g. LabeledLDA) for classifying named entities has a similar effect, in that information about an entity's distribution of possible types is shared across its mentions.

5 Conclusions

We have demonstrated that existing tools for POS tagging, Chunking and Named Entity Recognition perform quite poorly when applied to Tweets. To address this challenge we have annotated tweets and built tools trained on unlabeled, in-domain and out-of-domain data, showing substantial improvement over their state-of-the art news-trained counterparts, for example, T-POS outperforms the Stanford POS Tagger, reducing error by 41%. Additionally we have shown the benefits of features generated from T-POS and T-CHUNK in segmenting Named Entities.

We identified named entity classification as a particularly challenging task on Twitter. Due to their terse nature, tweets often lack enough context to identify the types of the entities they contain. In addition, a plethora of distinctive named entity types are present, necessitating large amounts of training data. To address both these issues we have presented and evaluated a distantly supervised approach based on LabeledLDA, which obtains a 25% increase in F_1 score over the co-training approach to Named Entity Classification suggested by Collins and Singer (1999) when applied to Twitter.

Our POS tagger, Chunker Named Entity Recognizer are available for use by the research community: `http://github.com/aritter/twitter_nlp`

Acknowledgments

We would like to thank Stephen Soderland, Dan Weld and Luke Zettlemoyer, in addition to the anonymous reviewers for helpful comments on a previous draft. This research was supported in part by NSF grant IIS-0803481, ONR grant N00014-11-1-0294, Navy STTR contract N00014-10-M-0304, a National Defense Science and Engineering Graduate (NDSEG) Fellowship 32 CFR 168a and carried out at the University of Washington's Turing Center.

References

Edward Benson, Aria Haghighi, and Regina Barzilay. 2011. Event discovery in social media feeds. In *The 49th Annual Meeting of the Association for Computational Linguistics*, Portland, Oregon, USA. To appear.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.*

Avrim Blum and Tom M. Mitchell. 1998. Combining labeled and unlabeled sata with co-training. In *COLT*, pages 92–100.

Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. 1992. Class-based n-gram models of natural language. *Comput. Linguist.*

Andrew Carlson, Justin Betteridge, Richard C. Wang, Estevam R. Hruschka, Jr., and Tom M. Mitchell. 2010. Coupled semi-supervised learning for information extraction. In *Proceedings of the third ACM international conference on Web search and data mining*, WSDM '10.

Eugene Charniak, Curtis Hendrickson, Neil Jacobson, and Mike Perkowitz. 1993. Equations for part-of-speech tagging. In *AAAI*, pages 784–789.

Michael Collins and Yoram Singer. 1999. Unsupervised models for named entity classification. In *Empirical Methods in Natural Language Processing*.

Doug Downey, Matthew Broadhead, and Oren Etzioni. 2007. Locating complex named entities in web text. In *Proceedings of the 20th international joint conference on Artifical intelligence*.

Doug Downey, Oren Etzioni, and Stephen Soderland. 2010. Analysis of a probabilistic model of redundancy in unsupervised information extraction. *Artif. Intell.*, 174(11):726–748.

Micha Elsner, Eugene Charniak, and Mark Johnson. 2009. Structured generative models for unsupervised named-entity clustering. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09.

Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. 2005. Unsupervised named-entity extraction from the web: an experimental study. *Artif. Intell.*

Tim Finin, Will Murnane, Anand Karandikar, Nicholas Keller, Justin Martineau, and Mark Dredze. 2010. Annotating named entities in Twitter data with crowdsourcing. In *Proceedings of the NAACL Workshop on Creating Speech and Text Language Data With Amazon's Mechanical Turk*. Association for Computational Linguistics, June.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05.

Radu Florian. 2002. Named entity recognition as a house of cards: classifier stacking. In *Proceedings of the 6th conference on Natural language learning - Volume 20*, COLING-02.

Eric N. Forsythand and Craig H. Martell. 2007. Lexical and discourse analysis of online chat dialog. In *Proceedings of the International Conference on Semantic Computing*.

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*.

Joshua T. Goodman. 2001. A bit of progress in language modeling. Technical report, Microsoft Research.

Stephan Gouws, Donald Metzler, Congxing Cai, and Eduard Hovy. 2011. Contextual bearing on linguistic variation in social media. In *ACL Workshop on Language in Social Media*, Portland, Oregon, USA. To appear.

T. L. Griffiths and M. Steyvers. 2004. Finding scientific topics. *Proceedings of the National Academy of Sciences*, April.

Mark Hachman. 2011. Humanity's tweets: Just 20 terabytes. In *PCMAG.COM*.

Bo Han and Timothy Baldwin. 2011. Lexical normalisation of short text messages: Makn sens a #twitter. In *The 49th Annual Meeting of the Association for Computational Linguistics*, Portland, Oregon, USA. To appear.

Catherine Kobus, François Yvon, and Géraldine Damnati. 2008. Normalizing sms: are two metaphors better than one ? In *COLING*, pages 441–448.

Zornitsa Kozareva and Eduard H. Hovy. 2010. Not all seeds are equal: Measuring the quality of text mining seeds. In *HLT-NAACL*.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Xiaohua Liu, Shaodian Zhang, Furu Wei, and Ming Zhou. 2011. Recognizing named entities in tweets. In *ACL*.

Brian Locke and James Martin. 2009. Named entity recognition: Adapting to microblogging. In *Senior Thesis, University of Colorado*.

Mitchell P. Marcus, Beatrice Santorini, and Mary A. Marcinkiewicz. 1994. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*.

Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. In `http://mallet.cs.umass.edu`.

Tara McIntosh. 2010. Unsupervised discovery of negative categories in lexicon bootstrapping. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10.

Einat Minkov, Richard C. Wang, and William W. Cohen. 2005. Extracting personal names from email: applying named entity recognition to informal text. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 443–450, Morristown, NJ, USA. Association for Computational Linguistics.

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

Christoph Müller and Michael Strube. 2006. Multi-level annotation of linguistic data with MMAX2. In Sabine Braun, Kurt Kohn, and Joybrato Mukherjee, editors, *Corpus Technology and Language Pedagogy: New Resources, New Tools, New Methods*, pages 197–214. Peter Lang, Frankfurt a.M., Germany.

Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D. Manning. 2009. Labeled lda: a supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 248–256, Morristown, NJ, USA. Association for Computational Linguistics.

Christina Sauper, Aria Haghighi, and Regina Barzilay. 2010. Incorporating content structure into text analysis applications. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 377–387, Morristown, NJ, USA. Association for Computational Linguistics.

Fei Sha and Fernando Pereira. 2003. Shallow parsing with conditional random fields. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03.

Sameer Singh, Dustin Hillard, and Chris Leggetter. 2010. Minimally-supervised extraction of entities from text advertisements. In *Human Language Technologies: Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT)*.

Charles Sutton. 2004. Collective segmentation and labeling of distant entities in information extraction.

Partha Pratim Talukdar and Fernando Pereira. 2010. Experiments in graph-based semi-supervised learning methods for class-instance acquisition. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1473–1481. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the conll-2000 shared task: chunking. In *Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning - Volume 7*, ConLL '00.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10.

Limin Yao, David Mimno, and Andrew McCallum. 2009. Efficient methods for topic model inference on streaming document collections. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, ACL '95.