Semi-supervised Named Entity Recognition in noisy-text

Shubhanshu Mishra
School of Information Sciences
University of Illinois at Urbana-Champaign
Champaign, IL – 61820, USA
smishra8@illinois.edu

Jana Diesner
School of Information Sciences
University of Illinois at Urbana-Champaign
Champaign, IL – 61820, USA
jdiesner@illinois.edu

Abstract

Many of the existing Named Entity Recognition (NER) solutions are built based on news corpus data with proper syntax. These solutions might not lead to highly accurate results when being applied to noisy, user generated data, e.g., tweets, which can feature sloppy spelling, concept drift, and limited contextualization of terms and concepts due to length constraints. The models described in this paper are based on linear chain conditional random fields (CRFs), use the BIEOU encoding scheme, and leverage random feature dropout for up-sampling the training data. The considered features include word clusters and pre-trained distributed word representations, updated gazetteer features, and global context predictions. The latter feature allows for ingesting the meaning of new or rare tokens into the system via unsupervised learning and for alleviating the need to learn lexicon based features, which usually tend to be high dimensional. In this paper, we report on the solution [ST] we submitted to the WNUT 2016 NER shared task. We also present an improvement over our original submission [SI], which we built by using semi-supervised learning on labelled training data and pre-trained resourced constructed from unlabelled tweet data. Our ST solution achieved an F1 score of 1.2% higher than the baseline (35.1% F1) for the task of extracting 10 entity types. The SI resulted in an increase of 8.2% in F1 score over the baseline (7.08% over ST). Finally, the SI model's evaluation on the test data achieved a F1 score of 47.3% (~1.15% increase over the 2nd best submitted solution). Our experimental setup and results are available as a standalone twitter NER tool at https://github.com/napsternxg/TwitterNER.

1 Introduction

A common task in information extraction is the identification of named entities from free text, also referred to as Named Entity Recognition (NER) (Sarawagi, 2008). In the machine learning and data mining literature, NER is typically formulated as a sequence prediction problem, where for a given sequence of tokens, an algorithm or model need to predict the correct sequence of labels. Additionally, most of the NER systems are designed or trained based on monolingual newswire corpora, which are written with proper linguistic syntax. However, noisy and user generated text data, which are common on social media, pose several challenges for generic NER systems, such as shorter and multilingual texts, ever evolving word forms and vocabulary, improper grammar, and shortened or incorrectly spelled words. Let us consider a fictional tweet: "*r u guyz goin to c da* ***#coldplay*** *show* ***@madisonsqrgrdn***☺?". This tweet contains two named entities, namely: "Coldplay", a music band, and "Madison Square Garden, NYC, USA", a geolocation, which references the place at which the band is playing. Many of the terms present in the exemplary tweet would be considered as out of vocabulary (OOV) terms by traditional NER systems. Furthermore, using a large set of such OOV tokens for training a classifier is likely to result in a sparse and high dimensional feature space, thereby increasing computing time. The phenomenon of concept-drift, i.e., the meaning of terms shifting over time, has also been found to affect the accuracy of NER systems over time, resulting in poor performance of a classifier trained on older data (Cherry & Guo, 2015; Derczynski, Maynard, et al., 2015; Fromreide et al., 2014; Hulten et al., 2001; Masud et al., 2010).

Proceedings of the 2nd Workshop on Noisy User-generated Text,
pages 203–212, Osaka, Japan, December 11 2016.

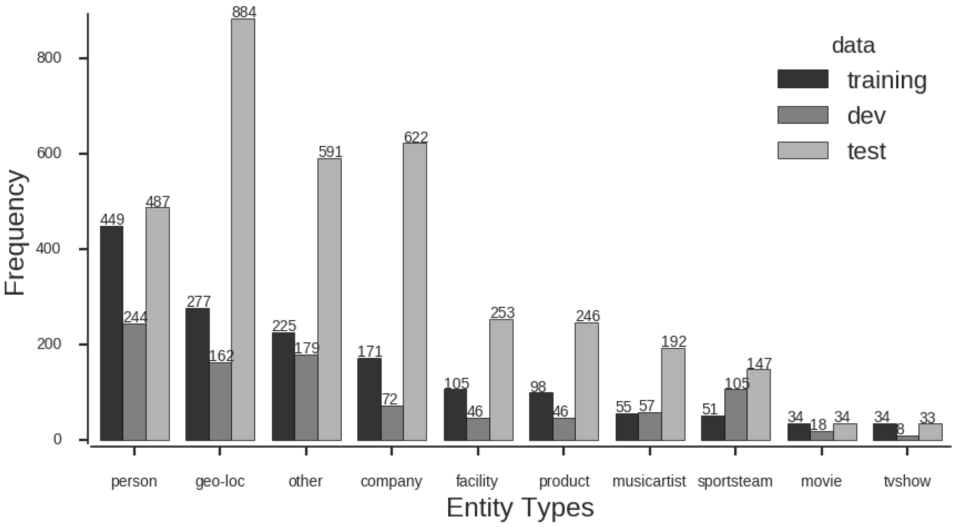

Figure 1: Frequency of named entity types in training, development, and test datasets.

The Workshop on "Noisy User-generated Text" (WNUT) continued its 2015 shared task on NER on tweets (Baldwin et al., 2015) in 2016. In 2016, the task was divided into two parts: (1) identification of named entities in tweets, and (2) NER on 10 types of entities, namely *person, geo-location, other, company, sports-team, facility, product, music-artist, movie, and tv-show*.

In this paper we introduce two solutions to perform NER on tweets. The first system, which we will refer to as the submitted solution [ST], was submitted as an entry to the WNUT 2016 NER shared task. It uses random feature [RF] dropout for up-sampling the dataset. This system was improved into a semi-supervised solution (our 2nd solution [SI]), which uses additional, unsupervised features. These features have been found to be useful in prior information extraction and NER tasks. The semi-supervised approach circumvents the need to include word n-gram features from any tweets, and builds upon the successful usage of word representations (Collobert et al., 2011), and word clusters (Lin & Wu, 2009; Miller et al., 2004; Ratinov & Roth, 2009; Turian et al., 2010) for NER by utilizing large amounts of unlabelled data or models pre-trained on a large vocabulary. The SI system was designed to mitigate the various issues mentioned above, and utilizes the unlabelled tokens from the all the available datasets (including unlabelled test data) to improve the prediction quality on the evaluation datasets, a form of transductive learning (Joachims, 2003). The SI system outperforms ST by ~7% (F1 score) when using the development set for evaluation, and by ~11% when using the test set (1% higher than the 2nd best team in the task). The SI model does not utilize any word n-gram lexical features. We believe that the approach taken for SI is useful for situations that require refinement or adaptation of an existing classifier to perform well on a new test set. We have released our experimental setup and code at https://github.com/napsternxg/TwitterNER.

2 Data

The training, development, and test dataset were provided by the task organizers. The training set consists of 2,394 tweets with a total of 1,499 named entities. The organizers provided two separate development datasets, which we merged to create a dataset of 1,420 tweets with 937 named entities. This merged dataset was used as the development dataset for all of our experiments. The test dataset comprises 3,856 tweets with 3,473 named entities. Most of the tweets in the provided data lack any entities mentions (42% in training, 59% in development, and 47% in test data), resulting in sparse training samples. Furthermore, certain types of entities, such as *movie* and *tvshow*, have only a few instances. The frequency distribution of the different types of named entities in the training, development, and test data are shown in Figure 1. Additionally, we found that the training, development, and test data have an average of 19.4 (±7.6), 16.2 (±6.8), and 16.1 (±6.6) tokens per sequence, respectively, and mostly contain less than 3 entities per tweet. This implies that the presence of certain entity types might be reflective of the category of the tweet, e.g. *movie* entities will be found in tweets about movies, and *sports-team* entities will be found in tweets about sports. Additionally, some types of entities are more likely to co-occur with each other than others. Using the provided data, we found that both *person* and *geo-location* entities were most likely to co-occur with entities of other 8 types, compared to the co-occurrence of the rest of the entities.

Although the original dataset was tagged using the **B**egin-**I**nside-**O**utside (BIO) encoding, we converted that into the **B**egin-**I**nside-**E**nd-**O**utside-**U**nigram (BIEOU) encoding, which has been found to be more efficient for sequence classification tasks (Ratinov & Roth, 2009). However, the predicted tags were converted back to the BIO encoding to make our submission compatible with the evaluation system.

3 Feature Engineering

We trained our system using multiple combinations of features. Features were chosen with the intent to increase the generalizability and scalability of our classifier. Some of the considered features can be updated with the availability of new unlabelled data, while other features capture the general token patterns in tweets. All features are described in detail in the following subsections.

3.1 Regex features [RF]

Regular expressions are rules describing regularities in data, and are typically empirically derived. For example, in regular English corpora, named entities usually being with capital letters. Although regex based approaches can be effective, they are likely to result in retrieving large amounts of false positives. Most NER systems use token level regex features (Baldwin et al., 2015; Ratinov & Roth, 2009). We extended these regex features by including features which detect syntax patterns of tokens commonly present in tweets. Our patterns return "true" if the regex pattern matches the token. A detailed list of our regex features is described in Table 1. These features were extracted per token, and every pair of the neighbouring tokens' regex features were multiplied to create pairwise features.

isHashtag - Identifies if token is a hashtag
isMention - Identifies if token is a user mention
isMoney - Identifies if token represents monetary values
isNumber - Identifies if token is a number
isDigits - Identifies if token only consists of digits
isAllCapitalWord - Identifies if token only consists of capital alphabets
isAllSmallCase - Identifies if token only consists of small alphabets
isWord - Identifies if token only consists of letters
isAlphaNumeric - Identifies if token only consists of digits and letters
isSingleCapLetter - Identifies if token only consists of single capital letter
isSpecialCharacter - Identifies if token only consists of special characters such as: *#;:-/<>'"()&*

endsWithDot- Identifies if token only consists of alphanumeric and ends with a `.`, e.g. *Dr*
containsDashes - Identifies if token only contains dashes
containsDigits - Identifies if token only contains digits
singlePunctuation - Identifies if token is only single punctuation
repeatedPunctuation - Identifies if token only consists of repeated punctuations
singleDot - Identifies if token only consists of a single dot
singleComma- Identifies if token only consists of a single comma
fourDigits- Identifies if token only consists of four digits
singleQuote- Identifies if token only consists of a single quotation mark

Table 1: List of regex features

3.2 Gazetteers [GZ]

The task organizers provided a set of gazetteer lists. Although being helpful, these lists include some irregularities, such as words composed of or containing non-ascii characters, garbled strings, and missing names of important named entities in many categories. Furthermore, the provided gazetteers did not include names of movies or music artists. We increased the given set of gazetteers by including an additional 41K person names, 63K music artist names, 8K TV show titles, 2K sports team names, and 110K movie titles from WikiData (https://www.wikidata.org), additional 8.3M locations from GeoNames (http://www.geonames.org/), and 4.5M music artist names and their 1.4M name variants from the Discogs' public data dump (http://data.discogs.com/). Improved gazetteer features were also used as features in last year's shared task (Derczynski, Augenstein, et al., 2015). The gazetteer features were implemented on a per token level, where we look up a gazetteer phrase in a range of window sizes W (min=1 and max=6) both left and right of the current token. Additionally, we encode the window size and the identified gazetteer name. Finally, we include interaction terms computed as the product of all pairs of gazetteer features for each token.

3.3 Word representations [WR]

Distributed word representations have been shown to improve the accuracy of NER systems (Collobert et al., 2011; Turian et al., 2010). We used 200 dimensional GloVe word representations [WR_G] (Pennington et al., 2014), which were pre-trained on 6 billion tweets. Furthermore, we built a set of word clusters by performing an agglomerative clustering of word representations [WR_{FTC}] and fine tuning them on the training plus development dataset by running the word2vec model (Mikolov, Chen, et al., 2013; Mikolov, Sutskever, et al., 2013).

3.4 Word clusters [WC]

Word clusters are word groupings that get generated in an unsupervised fashion, and they have been successfully used as features for NER tasks (Lin & Wu, 2009; Miller et al., 2004; Ratinov & Roth, 2009; Turian et al., 2010). One algorithm for creating such sets is Brown clustering (Brown et al., 1992), which produces a hierarchical cluster of words in the corpus while optimizing the likelihood of a language model based on a Hidden Markov Model (HMM). We used pre-trained 1000 brown clusters [WC_{BPT}] that were prepared by using a large corpus of tweets (Gimpel et al., 2011; Owoputi et al., 2013). Additionally, we built another set of brown clusters [WC_{BD}] with a cluster size of 100 based on all of the available data by using the code provided by Liang (2005)[1]. Furthermore, we also used an implementation[2] of the algorithm proposed by Clark (2003) to create 32 (default option) additional word clusters from our training plus development data based on the regex and sequential features of the words. We choose to call these Clark clusters [WC_{CC}]. Additionally, for each token, we also included all word cluster features for their immediate neighbours along with interactive terms; with the latter capturing the product of the token cluster with the neighbouring cluster.

3.5 Additional features

Even though the strength of our system lies in its semi-supervised nature and its non-reliance on data specific features such as lexical tokens [LT], we still included lexical tokens for comparison. Additionally, we used certain global features [GF] for helping with the prediction. Global features capture the overall composition of the sequence. We constructed the GF using the average values of the word representations and the binary presence of cluster and dictionary features. Additionally, another feature was constructed, which approximates the probability of the sequence being of a certain type. This feature adds an additional context to the token level prediction task, e.g. a tweet about sports is more likely to mention a sports team, and similarly, a tweet about a company is more likely to mention a product and vice-versa. To use this global feature, we first trained a Logistic regression classifier to predict if a tweet is about any of each of the 10 types of entities. The predicted probability per type is used as a feature for each of the tokens in the sequence.

3.6 Random up-sampling with feature dropout [RS_{FD}]

Since the training dataset is comparatively small and its features are sparse, we create synthetic examples by dropping interaction and lexical features with probability *p*. These features were chosen for random dropout because our earlier experiments had shown that the classifier identifies large weights for these features. We further scaled the training data size by a factor of *k*. This technique is inspired by the success of the dropout technique (Srivastava et al., 2014), which serves as a regularization function for deep neural networks. However, our technique is slightly different in that we use dropout to create a larger number of noisy samples from our data. Also, in contrast to the basic dropout technique, we did not re-weight the feature weights using the dropout probability (Srivastava et al., 2014) during evaluation.

[1] https://github.com/percyliang/brown-cluster
[2] https://github.com/ninjin/clark_pos_induction

4 NER classification algorithm

We used a linear chain CRF (Lafferty et al., 2001; McCallum & Li, 2003) as implemented in CRFSuite (Okazaki, 2007) package for training all our models. The models were trained using stochastic gradient decent (SGD) with an L2 norm (C=1e-3). We also tested some of the recently popular deep learning based approaches, such as word embedding based and character based recurrent neural networks, for our prediction task. However, these techniques did not yield competitive results and were too slow to converge on CPU. Furthermore, training the CRF model was faster (average training time of the CRF algorithms was ~3 mins on CPU, compared to >15 minutes for the character/word based 3-layer deep recurrent neural network solution), and gave interpretable results while beating the baseline model provided by the task organizers. In the following sections, we will first describe the model we used in our submission to the shared task, and then our improvement over the initial model and results.

4.1 Shared task submission solution [ST] based on random feature dropout up-sampling

Our original submission to the shared task [ST] was based on a system that uses the lexical, regex, and dictionary based features with random feature dropout based up-sampling. All the interaction terms were randomly dropped out with $p=0.5$, and the scaling factor k was chosen to be 5. The dictionary based features were created using a context window of size 2 to the left and right of the token. Additional interaction features were included by calculating the product of the dictionary features of the token and the neighbouring tokens. Finally, ST was based on a classifier trained only on the training dataset, and was corpus specific in that it used the vocabulary created from the training data.

4.2 Semi-supervised word clusters and representation based solution [SI]

The described lexicon based solution [ST] had one major drawback: The most highly weighted features were mainly tokens descriptive of entity types as occurring in the training data. For example, the highest weighted feature for the label *U-person* was *word_normed:pope*. Similarly, for many of the other entity types, the highest weighted features were the names or labels of popular entities. Although these features help to achieve a decent evaluation score on the development dataset, they can lead to overfitting of the classifier to the vocabulary of the training corpus. In order to circumvent this issue, a semi-supervised (Blum, 1998; Blum & Mitchell, 1998) solution builds on the general recent success of using word representations and word clusters in NER tasks, while disregarding lexical vocabulary based features. The intuition behind our approach to the 2nd solution [SI] was to ensure that the classifier learns higher level representations of the observed tokens. All the features used for our second solution augment the tokens present in the given tweets. This allows us to scale-up the underlying resources, such as gazetteers, and improve word representations and clusters using the new unlabelled test data, while still being able to update the classifier from the initially provided, limited training data. We replicate this behaviour in our classifiers by training our clusters on all of the unlabelled data generated by merging tweet texts from the training, development, and test data (only un-labelled) [TDT_E] (Blum & Mitchell, 1998), and comparing the resulting performance to that obtained with unsupervised training that does not consider the test data [TD]. Although it might appear that our classifier has access to the unlabelled test data sequences while learning, it rather is the case that we resemble an online setting where we continuously update our unsupervised features using the new batch of unlabelled test data, and then retrain our model on the original training data (Blum, 1998; Blum & Mitchell, 1998; Carlson et al., 2010; Chapelle et al., 2009; Liang, 2005; Turian et al., 2010; Zhu & Goldberg, 2009). In this case, the unlabelled data prevent the classifier from overfitting to the training data by acting as a regularization factor. An alternative approach would be to train these clusters on a large number of unlabelled tweets that match the time range and search domain of the test tweets.

5 Results

In the following sections, we describe the evaluation of the accuracy of both the ST and SI system in comparison to BL and against each other. All evaluations were done by using the evaluation script provided by the organizers. We use the classifier provided by the organizers as the baseline (BL) system. The baseline system uses lexical, gazetteer, and regex features.

5.1 Performance in WNUT NER shared task

Using BL as a point of comparison, ST scored 1.1% (F1 score) higher for the 10-types task (based on the development set), and 1.2% (F1) lower for the no-types task. Our ST is based on random feature dropout based sampling.

Among the 10 participating teams, our solution placed 7th for the 10-types category with an overall F1 score of 36.95%, and 6th in the no-type category with an overall F1 score of 51.38%. The top team on both tasks (same team in both cases) achieved F1 scores of 52.41% and 65.89%, respectively. Overall, we found that ST performed best on the geo-location type (F1 score of 64.72%), and behind the top two teams (score of 72.61% and 68.36%, respectively) for this category. We placed 3rd in terms of F1 (37%) in the facility category shown Table 2.

Rank	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**TD**	**TDT$_{\text{E}}$**
10-types overall	52.4	46.2	44.8	40.1	39.0	37.2	**37.0**	36.2	29.8	19.3	**46.4**	**47.3**
No-types	65.9	63.2	60.2	59.1	55.2	51.4	**47.8**	46.7	44.3	40.7	57.3	59.0
company	57.2	46.9	43.8	31.3	38.9	34.5	**25.8**	42.6	24.3	10.2	42.1	46.2
facility	42.4	31.6	36.1	36.5	20.3	30.4	**37.0**	40.5	26.3	26.1	37.5	34.8
geo-loc	72.6	68.4	63.3	61.1	61.1	57.0	**64.7**	60.9	47.4	37.0	70.1	71.0
movie	10.9	5.1	4.6	15.8	2.9	0.0	**4.0**	5.0	0.0	5.4	0.0	0.0
musicartist	9.5	8.5	7.0	17.4	5.7	37.2	**1.8**	0.0	2.8	0.0	7.6	5.8
other	31.7	27.1	29.2	26.3	21.1	22.5	**16.2**	13.0	22.6	8.4	31.7	32.4
person	59.0	51.8	52.8	48.8	52.0	42.6	**40.5**	52.3	34.1	20.6	51.3	52.2
product	20.1	11.5	18.3	3.8	10.0	7.3	**5.7**	15.4	6.3	0.8	10.0	9.3
sportsteam	52.4	34.2	38.5	18.5	34.6	15.9	**9.1**	19.7	11.0	0.0	31.3	32.0
tvshow	5.9	0.0	4.7	5.4	7.3	9.8	**4.8**	0.0	5.1	0.0	5.7	5.7
Rank	1	2	3	4	5	6	**7**	8	9	10	**~2**	**~2**

Table 2: Results of the WNUT NER 2016 shared task. Rank denotes the rank of the winning team, which we use as an ID to identify the evaluation performance of each of the participating teams in the shared task. Our solution was ranked 7th (in bold) and (6th not shown) in the 10-types and no-types categories, respectively. Columns with TD and TDT$_{\text{E}}$ show the performance of the improved model on the test data, and their ranks denote the best rank in the competition which they beat.

5.2 Improved model performance [SI]

In this section we describe the evaluation of our improved system SI, which was developed after the release of the shared task results. Since we received the gold standard labels for the test-set late in the process, we evaluated most of the improved models based on the development set. We present the additive effect of a series of features to the model in Table 3. Additionally, that table also shows the performance of ST and BL. We do not include any lexical features in SI, however, lexical features were part of the ST and BL models. We found that the addition of the gazetteer [GZ] features improved the classification accuracy considerably. The next two big jumps in accuracy increase in SI came from using brown clusters [WC$_{\text{BTP}}$] and fine-tuned word representations based clusters [WC$_{\text{FTC}}$]. From all of the improved models that we trained, we selected the 10-types category model with the highest overall F1 score, namely RF+GZ+WR$_{\text{G}}$+WC$_{\text{BPT}}$+WC$_{\text{CC}}$+WR$_{\text{FTC}}$ model, also referred to as SI herein. Only the SI model was also evaluated on the test data with [TDT$_{\text{E}}$] as well as without [TD], using the test data for enriching the unsupervised features. Although the model with the global features [See +GF in Table 3] is not the top one in terms of the F1 score, it achieved considerably high scores for *movie* and *tvshow* class, which have very few training instances. Similarly, the random dropout up-sampling based solution showed improvements by 15% and 6% F1 score in terms of predicting named entities of the types movie

Additive Features →	**RF**	**+GZ**	**+WR$_G$**	**+WC$_{BPT}$**	**+WC$_{CC}$**	**+WR$_{FTC}$**	**+GF**	**+RS$_{FD}$**	**ST**	**BL**	**TD**	**TDT$_E$**
10-types	5.3	34.8	36.7	41.6	41.0	43.3	40.9	40.0	36.2	35.1	46.4	**47.3**
company	0.0	30.0	34.5	33.3	35.2	33.3	32.0	33.3	27.7	26.2	42.1	**46.2**
facility	0.0	12.4	9.6	20.8	18.6	17.9	14.5	16.7	30.4	19.2	**37.5**	34.8
geo-loc	5.2	47.2	48.1	53.8	54.4	55.9	56.7	56.1	49.7	48.4	70.1	**71.0**
movie	8.0	7.4	6.5	8.3	7.7	9.5	23.5	**28.6**	8.3	0.0	0.0	0.0
musicartist	0.0	6.6	8.5	9.1	9.5	12.7	6.5	**14.7**	0.0	0.0	7.6	5.8
other	5.8	18.6	18.7	22.5	20.9	26.6	22.1	17.7	24.2	27.7	31.7	**32.4**
person	11.4	55.1	58.5	63.4	63.8	64.8	**65.0**	60.2	53.4	50.2	51.3	52.2
product	2.9	12.7	**20.0**	16.7	18.2	15.4	10.8	11.9	9.0	11.9	10.0	9.3
Sportsteam	0.0	12.9	27.9	30.5	29.0	28.1	27.7	25.4	12.8	13.1	31.3	**32.0**
tvshow	0.0	0.0	0.0	16.7	16.7	16.7	**18.2**	13.3	0.0	14.3	5.7	5.7
No-types	13.1	48.3	52.5	56.7	56.4	57.4	53.7	52.9	50.5	51.7	57.3	**59.0**

Table 3: Change in F1 score for the NER classifier on the development dataset on incremental addition of different types of features (from left to right). ST refers to submitted solution, BL refers to baseline solution provided by the organizers. Bolded values are the best scores across classifiers.

and music-artist, respectively. Finally, these models were trained in almost half the time as the ST models.

5.3 Features learned by the model

We extracted the learned features from the top performing model on the 10-types category (the RF+GZ+WR$_G$+WC$_{BPT}$+WC$_{CC}$+WR$_{FTC}$ model). The features with the highest positive and negative weights for each of the category labels are presented in Table 4. The table also shows that for *person, product, movie*, and *tvshow* the top features were specific dimensions of the pre-trained word embedding. Furthermore, the brown cluster ids of the token word are more informative for the named entities of *geo-location*, *other*, and *company* types, while the brown cluster id of neighbouring tokens is likely to indicate if a named entity is of type *musicartist*, *sportsteam*, or *facility*. Additionally, if the token belongs to a phrase in a gazetteer of music artist names, then it is less likely to be a *geo-loc*, *company*, or *product*.

Type	**Most positive weight**		**Most negative weight**	
Entity	**feature**	**w**	**feature**	**w**
person	WV_90→U	1.27	WV_46→U	-1.02
other	BC_2_:1001000→U	1.28	isAllSmallCase|isAlphaNumeric[+1] →U	-0.88
geo-loc	BC_0:11100110101->U	2.33	DICT=musicartist_names→U	-0.88
facility	BC_2[-1]:1001111110→B	1.63	WV_185->B	-0.68
company	BC_0:111001100001→U	1.30	DICT=musicartist_namevars->U	-0.77
product	WV_199→U	1.07	DICT=musicartist_namevars→U	-0.97
musicartist	BC_2[-1]:11110010→U	1.21	DICT=geonames→U	-0.80
movie	WV_75→B	0.76	isAlphaNumeric[+1] →E	-0.50
sportsteam	BC_0[+1]:1111011010→B	1.29	WV_30→U	-0.86
tvshow	WV_154→U	0.76	isInitCapitalWord|singlePunctuation[+1] →E	-0.40

Table 4: Feature weights (**w**) in the SI model for each of the 10 entity types. WV is word vector; BC is brown cluster. [-/+ num] denotes features for left or right neighbour, respectively. → BIEOU denotes which boundary type for the entity type the feature belongs to.

We also investigated the transition features of the linear chain CRF model. The transition matrix (based on transition weights) is presented in Figure 2, and coloured as red for negative weights and black for positive weights. Some trends become obvious from the transition matrix: For most entity types, the model is able to find high transition weights for going from *B* to *I* to *E*, while penalizing transitions between the other states. The choice of using *BIEOU* tagging is supported by the results shown in the transition matrix since for most entity types, there is a high negative weight for going from the *B* or *I* tag to the *O* tag. However, a transition from the *U* tag to *O* tag is usually supported. Our earliest experiments (not reported here) revealed that there was a considerable improvement from using the BIEOU tagging scheme. This finding aligns with the existing research which argues for the usage of this tagging scheme for NER tasks.

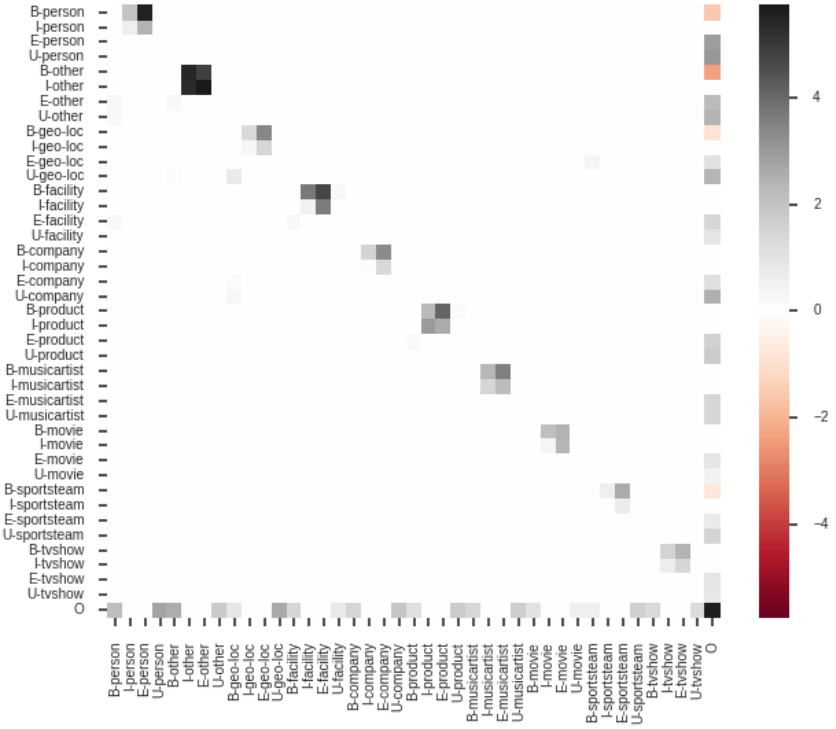

Figure 2: Transition weights learned by the SI model.

6 Discussion and conclusion

Prior work has shown that semi-supervised algorithms can perform decently for NER tasks with sparse labelled data (Blum, 1998; Carlson et al., 2010; Chapelle et al., 2009; Liang, 2005; Turian et al., 2010; Zhu & Goldberg, 2009). We leverage this fact in our SI model via the use of unsupervised word clusters, word representations, and refined gazetteers; all of which contributed to a cumulative increase in accuracy over our initial submission [ST] by ~11% when using the test data for evaluation. Furthermore, the transition features learned by our model are reflective of correct learning of NER sequences and demonstrate the strength of using the BIEUO encoding scheme. Additionally, the supervised training of our classifier on features extracted from the unlabelled data, as opposed to lexical token features, reduces the dimensionality of the training data for the classifier and results in increased performance in terms of both accuracy and training time. Furthermore, our model can be adjusted on the arrival of new unlabelled data by updating the underlying learned word clusters and representations, and retraining the model on the existing labelled data. As identified by Turian et al. (2010), the importance of word representations and word clusters increases as the availability of unlabelled data increases. We can add additional entity names to the gazetteers. Retraining the model on the same training data would then allow for accommodating to the new feature representations. Finally, the random feature dropout based up-sampling can help to increase the amount of training data available, and can also be improved by random swapping of entity types in the training data with their nearest neighbours in the word representations and clusters, or by choosing entities from the most correlated gazetteers. We believe that our described models can help in improving NER on noisy-text, and our open source implementation can be further extended.

7 Acknowledgements

We would like to acknowledge the three anonymous reviewers for their useful feedback.

Reference

Baldwin, Timothy, de Marneffe, Marie-Catherine, Han, Bo, Kim, Young-Bum, Ritter, Alan, & Xu, Wei. (2015). Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition. In *First Workshop on Noisy User-generated Text at ACL '15*, (pp. 126-135), Association for Computational Linguistics, Beijing, China.

Blum, Avrim. (1998). On-line algorithms in machine learning. In A. Fiat & G. J. Woeginger (Eds.), *Online algorithms: The state of the art* (pp. 306-325): Springer Berlin Heidelberg.

Blum, Avrim, & Mitchell, Tom. (1998). Combining labeled and unlabeled data with co-training. In *Eleventh Annual Conference on Computational Learning Theory (CoLT '98)*, (pp. 92-100), ACM, Madison, Wisconsin, USA.

Brown, Peter F., deSouza, Peter V., Mercer, Robert L., Watson, T. J., Della Pietra, Vincent J., & Lai, Jenifer C. (1992). Class-based n-gram models of natural language. *Computational Linguistics*, *18*(4).

Carlson, Andrew, Betteridge, Justin, Wang, Richard C., Estevam R. Hruschka, Jr., & Mitchell, Tom M. (2010). Coupled semi-supervised learning for information extraction. In *Third ACM International Conference on Web Search and Data Mining (WSDM '10)*, (pp. 101-110), ACM, New York, New York, USA.

Chapelle, O., Scholkopf, B., & Eds, A. Zien. (2009). Semi-supervised learning (Vol. 20). *The MIT Press*.

Cherry, Colin, & Guo, Hongyu. (2015). The unreasonable effectiveness of word representations for twitter named entity recognition. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (HLT-NAACL '15)*, (pp. 735-745), Association for Computational Linguistics, Denver, Colorado

Clark, Alexander. (2003). Combining distributional and morphological information for part of speech induction. In *Tenth Conference on European Chapter of the Association for Computational Linguistics (EACL '03)*, (pp. 59-66), (Vol. 1), Budapest, Hungary.

Collobert, Ronan, Weston, Jason, Bottou, Léon, Karlen, Michael, Kavukcuoglu, Koray, & Kuksa, Pavel. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research (JMLR), 12*, 2493-2537.

Derczynski, Leon, Augenstein, Isabelle, & Bontcheva, Kalina. (2015). Usfd: Twitter ner with drift compensation and linked data. In *First Workshop on Noisy User-generated Text at ACL '15*, Beijing, China.

Derczynski, Leon, Maynard, Diana, Rizzo, Giuseppe, van Erp, Marieke, Gorrell, Genevieve, Troncy, Raphaël, Petrak, Johann, & Bontcheva, Kalina. (2015). Analysis of named entity recognition and linking for tweets. *Information Processing & Management*, *51*(2), 32-49.

Fromreide, Hege, Hovy, Dirk, & Søgaard, Anders. (2014). Crowdsourcing and annotating ner for twitter #drift. In *Ninth International Conference on Language Resources and Evaluation (LREC '14)*, European Language Resources Association (ELRA), Reykjavik, Iceland.

Gimpel, Kevin, Schneider, Nathan, O'Connor, Brendan, Das, Dipanjan, Mills, Daniel, Eisenstein, Jacob, Heilman, Michael, Yogatama, Dani, Flanigan, Jeffrey, & Smith, A. Noah. (2011). Part-of-speech tagging for twitter: Annotation, features, and experiments. In *49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (HLT-ACL '11)*, (pp. 42-47), Association for Computational Linguistics, Stroudsburg, PA, USA.

Hulten, Geoff, Spencer, Laurie, & Domingos, Pedro. (2001). Mining time-changing data streams. In *Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '01)*, (pp. 97-106), ACM, San Francisco, California, USA.

Joachims, Thorsten. (2003). Transductive learning via spectral graph partitioning. In *Twentieth International Conference on Machine Learning (ICML '03)*, (pp. 290-297), (Vol. 3), Washington, DC, USA.

Lafferty, John, McCallum, Andrew, & Pereira, Fernando. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Eighteenth International Conference on Machine Learning (ICML '01)*, (pp. 282--289), Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Liang, Percy. (2005). *Semi-supervised learning for natural language.* (PhD Thesis), Massachusetts Institute of Technology.

Lin, Dekang, & Wu, Xiaoyun. (2009). Phrase clustering for discriminative learning. In *Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP (ACL '09)*, (pp. 1030-1038), Association for Computational Linguistics, Suntec, Singapore.

Masud, Mohammad M., Chen, Qing, Khan, Latifur, Aggarwal, Charu, Gao, Jing, Han, Jiawei, & Thuraisingham, Bhavani. (2010). Addressing concept-evolution in concept-drifting data streams. In *Tenth IEEE International Conference on Data Mining (ICDM '10)*, (pp. 929-934), IEEE Computer Society, Sydney, Australia.

McCallum, Andrew, & Li, Wei. (2003). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Seventh Conference on Natural Language Learning at North American Chapter of the Association for Computational Linguistics: Human Language*

Technologies (HLT-NAACL '03), (pp. 188-191), (Vol. 4), Association for Computational Linguistics, Stroudsburg, PA, USA.

Mikolov, Tomas, Chen, Kai, Corrado, Greg, & Dean, Jeffrey. (2013). Efficient estimation of word representations in vector space. *Computing Research Repository (CoRR), abs/1301.3781*.

Mikolov, Tomas, Sutskever, Ilya, Chen, Kai, Corrado, Greg S, & Dean, Jeff. (2013). Distributed representations of words and phrases and their compositionality. In *Twenty Sixth Neural Information Processing Systems (NIPS '13)*, (pp. 3111-3119), (Vol. 26), Lake Tahoe, USA.

Miller, Scott, Guinness, Jethran, & Zamanian, Alex. (2004). Name tagging with word clusters and discriminative training. In *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL '04)*.

Okazaki, Naoaki. (2007). Crfsuite: A fast implementation of conditional random fields (crfs). Retrieved from http://www.chokkan.org/software/crfsuite/

Owoputi, Olutobi, O'Connor, Brendan, Dyer, Chris, Gimpel, Kevin, Schneider, Nathan, & Smith, A. Noah. (2013). Improved part-of-speech tagging for online conversational text with word clusters. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (HLT-NAACL '13)*, (pp. 380-390), Association for Computational Linguistics, Atlanta, GA, USA.

Pennington, Jeffrey, Socher, Richard, & Manning, Christopher D. (2014). Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP '14)*, (pp. 1532-1543), (Vol. 14), Doha, Qatar.

Ratinov, Lev, & Roth, Dan. (2009). Design challenges and misconceptions in named entity recognition. In *Thirteenth Conference on Computational Natural Language Learning (CoNLL '09)*, (pp. 147-155), Association for Computational Linguistics, Boulder, Colorado.

Sarawagi, Sunita. (2008). Information extraction. *Foundations and Trends in Databases, 1*(3), 261-377.

Srivastava, Nitish, Hinton, Geoffrey E, Krizhevsky, Alex, Sutskever, Ilya, & Salakhutdinov, Ruslan. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research (JMLR), 15*(1), 1929-1958.

Turian, Joseph , Ratinov, Lev, & Bengio, Yoshua. (2010). Word representations: A simple and general method for semi-supervised learning. In *48th Annual Meeting of the Association for Computational Linguistics (ACL '10)*, (pp. 384-394), Association for Computational Linguistics, Uppsala, Sweden.

Zhu, Xiaojin, & Goldberg, Andrew B. (2009). Introduction to semi-supervised learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning, 3*(1), 1-130.