Hallym: Named Entity Recognition on Twitter with Induced Word Representation

Eun-Suk Yang
Hallym University
esyang219@gmail.com

Yu-Seop Kim
Hallym University
yskim01@hallym.ac.kr

Abstract

Twitter is a type of social media that contains diverse user-generated texts. Traditional models are not applicable to tweet data because the text style is not as grammaticalized as that of newswire. In this paper, we construct word embeddings via canonical correlation analysis (CCA) on a considerable amount of tweet data and show the efficacy of word representation. Besides word embedding, we use part-of-speech (POS) tags, chunks, and brown clusters induced from Wikipedia as features. Here, we describe our system and present the final results along with their analysis. Our model achieves an F1 score of 37.21% with entity types and distinguishes 53.01% of the entity boundaries.

1 Introduction

Named entity recognition (NER) is a task of finding and classifying names of things, such as person, location, and organization, given a sequence of words. NER is a very important subtask of information extraction (IE).

With the development of the Internet, a huge amount of information has been generated by users. The information generated on the Internet, particularly on social media (e.g., Twitter and Facebook), includes very diverse and noisy texts. The volume of Twitter data has increased rapidly, and about 500 million tweets are sent per day[1]. In recent years, Twitter data have considered a new source in nature and researchers are paying increased attention to them (Bollen et al., 2011; Mathioudakis and Koudas, 2010).

Twitter is a type of microblogging service in which users are allowed to post contents such as small messages, individual images, or videos. There are a number of microblogging sites such as Twitter, Tumblr, Plurk and identi.ca. Each service has its own characteristics. For example, Plurk has a timeline view for videos and pictures, and Twitter has "status updates."

The characteristic of "status updates" is one of the features that makes the classification of named entities in Twitter difficult. In Twitter, there is a limit for the number of characters that people can post at once. People post their thoughts with a short sentence; this leads to the problem that tweets do not contain sufficient contextual information (Ritter et al., 2011).

The shared task of ACL W-NUT 2015 is to find named entities on Twitter. Here, we will focus on ten types of named entities: company, facility, geo-loc, movie, musicartist, other, person, product, sportsteam, and tvshow. We have the training and development data for Twitter and 53 gazetteers from the abovementioned shared task.

In this paper, we describe the datasets in Section 2 and present the model that we use in this study in Section 3. In Section 4, we discuss the features used and the methods used for generating these features. We present our final results along with their analysis in Section 5 and conclude this paper in Section 6.

2 Data and Labels

In this section, we introduce the considered datasets and describe the data format used. We also list the characteristics of each entity type with some examples.

2.1 Data

The datasets provided by shared task are raw tweets. Table 1 shows an overview of the sizes of these datasets. In a tweet, each line contains words and its label is separated by a tab and a blank line that forms a sentence boundary. All tokens follow the IOB format. The token with a B-prefix indi-

[1] See "http://www.internetlivestats.com/twitter-statistics/"

Proceedings of the ACL 2015 Workshop on Noisy User-generated Text, pages 72–77,
Beijing, China, July 31, 2015. ©2015 Association for Computational Linguistics

cates the beginning of a named entity and the token with an I-prefix indicates the inside of a named entity. An I-prefix only follows after a token with a B-prefix. An O tag indicates that a token does not belong to a specific named entity.

Data	Tweets	Tokens
train	1,795	37,899
test	1,000	16,261

Table 1: An overview of datasets.

2.2 Labels

In the system, we focus on the following ten types of named entities:

company The name of a company or a brand
e.g., Snapchat, Twitter, and Facebook

facility The name of an institution such as a museum, a center, or a restaurant
e.g., Iowa City schools and Disneyland

geo-loc The name of a city or country
e.g., Chicago and Russia

movie The title of a movie
e.g., Interstellar and Inception

musicartist The name of music groups or disc jockeys (DJs)
e.g., Taylor Swift and Lady Gaga

other A phrase that can be used generally such as the name of a ceremony or an anniversary, or the title of a song
e.g., X-mas and Murphy's law

person The name of a person; it can be the person's full name, last name, or first name
e.g., Steve King and Ellen

product The name of a product
e.g., Nokia 5800 and Coke

sportsteam The name of a sports team
e.g., Arsenal and West Ham

tvshow The title of a television (TV) show
e.g., The Persuaders and Pretty Little Liars

3 Model

Conditional Random Fields (CRFs) (Lafferty et al., 2001) and its variants have been successfully applied to various sequence labeling tasks (Maaten et al., 2011; Collins, 2002; McCallum and Li, 2003; Kim and Snyder, 2012; Kim et al., 2015b; Kim et al., 2015a; Kim and Snyder, 2013a; Kim and Snyder, 2013b). The NER task produces a sequence of named entity tags, $y = (y_1 \dots y_n)$, given a sequence of words, $x = (x_1 \dots x_n)$. We model the conditional probability $p(y|x;\theta)$ using linear-chain CRFs:

$$p(y|x;\theta) = \frac{\exp(\theta \cdot \Phi(x,y))}{\sum_{y' \in \mathcal{Y}(x)} \exp(\theta \cdot \Phi(x,y'))}$$

where θ denotes a set of model parameters. $\mathcal{Y}$ returns all possible label sequences of x, and Φ maps (x, y) into a feature vector that is a linear sum of the local feature vectors: $\Phi(x,y) = \sum_{j=1}^{n} \phi(x, j, y_{j-1}, y_j)$. Given the fully labeled sequences $\{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$, the objective of the training is to find θ that maximizes the log likelihood of the training data under the model with l_2-regularization:

$$\theta^* = \underset{\theta}{\text{argmax}} \sum_{i=1}^{N} \log p(y^{(i)}|x^{(i)};\theta) - \frac{\lambda}{2} ||\theta||^2 .$$

4 Features

In this section, we describe a variety of features that we have used in this study. We also used CRFsuite[2] because it makes the application of new features easy. Apart from the base features and gazetteer features provided by the organizers, we have used the following new features: POS tags, chunks, brown clustering, and word representation. Our model is composed of the following features:

4.1 Base features

Base features include the gazetteer features and orthographic features. In the NER task, a huge amount of unlabeled data is often used for identifying unseen entities. There are already 53 gazetteers in the baseline system. The maximum window size for gazetteer features is 6, and the model will learn the named entity type associated

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

with a specific phrase, if it is in one or more of the gazetteer lexicons. Orthographic features can be divided into five types. The orthographic feature templates are as follows:

- n-gram: w_i for i in {-1,0,1}, conjunction of previous word and current word $w_{i-1}|w_i$ for i in {-1,0}.
- Affixes: Prefixes and suffixes of x_i. The first and last n characters ranging from 1 to 3.
- Capitalization: There are two patterns of capitalization: One is an indicator of capitalization for the first character, and the other is an indicator of capitalization for all characters.
- Digit: There are three patterns for numbers: i) Whether the current word has a digit, ii) whether the current word is a single digit, and iii) whether the current word has two digits.
- Non-alphabet: Whether the current word contains a hyphen and other punctuation marks. Among the other punctuation marks is the colon(:). In general, what follows right after a colon mark represents a feature weight. To make the model learn correctly, we normalize only the colon mark.

4.2 POS tags and chunks

In the NER task, POS tags and chunks contain very useful information for finding and classifying named entities. We predict POS tags and chunks by using a model trained with Twitter data. For POS tags, we use a model trained with the Penn Treebank-style tagset (Ritter et al., 2011). In a model, some Twitter-specific tags are added by Ritter et al. (2011): retweets, @usernames, #hashtags, and urls. For chunks, we use a named entity tagger[3] by Ritter et al. (2012). Predicted tags are used as features as follows:

- POS tag: a conjunction feature with the current word and the current POS tag $w_0|p_0$.
- Chunk tag: a unigram feature for chunk tag c_0 and a conjunction feature with the current word and the current chunk tag $w_0|c_0$.

4.3 Brown clustering

Brown clustering is a hierarchical clustering method that groups words into a binary tree of classes (Brown et al., 1992). We downloaded a brown clustering[4] based on Wikipedia provided by Turian et al. (2010). We used whole bit string of the current word.

4.4 Word representation

As a new source, tweet data are not applicable to the traditional model because of the different text structure. For a new model, it is natural to use annotated data. However, it is difficult to create new labeled data for a rapid generation of tweets. Instead of constantly annotate new data, the general solution is creating induced word representations from a large body of unlabeled data (Mikolov et al., 2013; Pennington et al., 2014; Kim et al., 2014; Anastasakos et al., 2014). A lot of previous work have used CCA because of its simplicity and generality (Kim et al., 2015c; Kim et al., 2015d; Stratos et al., 2014; Kim et al., 2015b). We create a word representation by using the canonical correlation analysis (Hotelling, 1936). Furthermore, word embeddings are induced from 13 million tweets containing 270 million tokens. The dimension of word embeddings we used is 50 with words occurring more than twice in the data . The window size for the contextual information is 3: the current word and a word to the left and the right of the current word.

5 Results

5.1 Error analysis

Twitter contains noisy and informal style text, and most of the state-of-art applications show a weak performance on Twitter data (Ritter et al., 2011). In this section, we check the errors for noisy text from the baseline system and categorize them. The last two errors are related to user-generated texts such as Twitter data.

Unseen word sequences: The main cause of this error is in a previously unseen sequence. A huge number of tweets are posted on Twitter every day and they contain up-to-date information on events. The most recent information such as new product information can lead to the formation of unprecedented word sequences. These sequences do not appear in

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

[4] http://metaoptimize.com/projects/wordreprs/

	$M_{noEmbedding}$			$M_{Embedding}$			+/-
Type	**P**	**R**	**F1**	**P**	**R**	**F1**	
Overall	35.95	31.92	33.81	39.59	35.10	37.21	+
company	27.59	20.51	23.53	32.14	23.08	26.87	+
facility	24.14	18.42	20.90	32.00	21.05	25.40	+
geo-loc	42.66	52.59	47.10	46.00	59.48	51.88	+
movie	14.29	6.67	9.09	8.33	6.67	7.41	-
musicartist	0.00	0.00	0.00	7.69	2.44	3.70	+
other	18.33	16.67	17.46	20.49	18.94	19.69	+
person	53.27	61.99	57.30	56.99	64.33	60.44	+
product	3.57	2.70	3.08	14.29	8.11	10.34	+
sportsteam	62.50	7.14	12.82	54.55	8.57	14.81	+
tvshow	0.00	0.00	0.00	0.00	0.00	0.00	.

Table 2: Results for model with and without word embedding. $M_{noEmbedding}$ and $M_{Embedding}$ represent the model with and without word embedding, respectively. The rightmost column shows the decrease or increase in the F1 score with respect to the model without word embedding. $M_{Embedding}$ denotes our final model.

the training data and gazetteers, and thus, the model cannot learn them.

Foreign languages: This error is caused by tweets written in languages other than English. Words written in foreign languages are annotated by the O tag and not include a named entity. However, some words have the same spelling as an English word and thus, activate the gazetteer features. This problem leads to words with the O tag being predicted as a named entity type.

Type disambiguation: There are some words that have the same spelling but belong to different types according to the contextual information. This error is often observed for named entities such as *sportsteam* and *musicartist*. The word sequences with this error have a correctly distinguished entity boundary but predict the wrong entity type. For example, *Tampa Bay* in "Losing to the Penguins quasi-AHL lineup in December is a non-issue for *Tampa Bay*" is an entity for *sportsteam*, but the model classifies it as *geo-loc* instead of *sportsteam*. In another example, the names of two music artists in "Will *Shawn Mendez* be opening up for *Taylor Swift*" are predicted as *person* and not as *musicartist*.

Informal name or abbreviations: Twitter users compress what they want to say to meet the limit of 140 characters. This leads to informal texts unlike in news articles. Note that abbreviations do not indicate official full forms such as airports or countries. For example, *Southie* in "Proud that the 1st modern Olympic Champion is James Brendan Connolly of *#Southie* ." is an informal name of *South Boston*, and this word does not appear in the training set and gazetteers. With respect to abbreviations, people use abbreviattions for indicating a day or a month, such as *Mon* for Monday and *Jan* for January. These words are contained in gazetteers and activate the gazetteer features. A model makes errors by predicting them as named entities.

Hashtag: A hashtag is a combination of the "#" sign and some characters for organizing word sequences as searchable links in Twitter. The rule is to not use any space between the characters in the hashtag. For instance, the word *New Delhi* is transformed into *#NewDelhi* as a hashtag, so it is difficult to check the gazetteer lexicons for such text.

5.2 The effectiveness of word embedding

In this subsection, we describe the effectiveness of word embedding by analyzing the results obtained by using the model with and without word embedding. The only difference between both the models is the use of brown clustering and the word representation based on CCA.

In the NER task, the F1 score is a more appropriate metric than accuracy. Most of the labels in the NER data contain the O tag, indicating that they are not an entity. Since this leads to high accuracy, by using the F1 score, we obtain a more reasonable harmonic function of the precision and the recall.

Table 2 shows the results obtained by using models with and without word embedding. As shown in table 2, brown clustering and word embedding have a good effect on performance. All types of entities except *movie* show error reduction. For determining the efficacy of word embedding, we compare the errors between the models without word embedding and with word embedding. We find that word embedding plays an important role in resolving the problem of unseen word sequences and the problem of type disambiguation. First, the model without word embedding does not learn about an entity *ipad Mini Retina 2nd Generation 16GB wifi* because some of the words do not appear in the training data. In contrast, the model with embedding can learn unseen words from the induced word representation. This helps the model to predict that the abovementioned entity indicates a product name. The model without word embedding also has the problem of disambiguation of a word *Edison* because the model only learns that this word is a person's name from the gazetteers. However, in the word sequence "Edison #weather on January 16 , 2015", *Edison* indicates a town in New Jersey. The model with word embedding is provided additional information by the word embedding process and predicts the abovementioned word as *geo-loc* correctly.

6 Conclusion

In this paper, we described the data and features used for generating our model. Besides POS tags and chunk tags, we used a word representation based on CCA for improving the model's performance. Our final model shows an error reduction of 14.08% from the baseline system. We also presented some primary and Twitter-specific problems by categorizing errors.

References

Tasos Anastasakos, Young-Bum Kim, and Anoop Deoras. 2014. Task specific continuous word representations for mono and multi-lingual spoken language understanding. In *ICASSP*, pages 3246–3250. IEEE.

Johan Bollen, Huina Mao, and Xiaojun Zeng. 2011. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8.

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. *Computational linguistics*, 18(4):467–479.

Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 1–8. Association for Computational Linguistics.

Harold Hotelling. 1936. Relations between two sets of variates. *Biometrika*, pages 321–377.

Young-Bum Kim and Benjamin Snyder. 2012. Universal grapheme-to-phoneme prediction over latin alphabets. In *EMNLP*, pages 332–343. Association for Computational Linguistics.

Young-Bum Kim and Benjamin Snyder. 2013a. Optimal data set selection: An application to grapheme-to-phoneme conversion. In *HLT-NAACL*, pages 1196–1205. Association for Computational Linguistics.

Young-Bum Kim and Benjamin Snyder. 2013b. Unsupervised consonant-vowel prediction over hundreds of languages. In *ACL (1)*, pages 1527–1536.

Young-Bum Kim, Heemoon Chae, Benjamin Snyder, and Yu-Seop Kim. 2014. Training a korean srl system with rich morphological features. In *ACL*, pages 637–642. Association for Computational Linguistics.

Young-Bum Kim, Minwoo Jeong, Karl Stratos, and Ruhi Sarikaya. 2015a. Weakly supervised slot tagging with partially labeled sequences from web search click logs. In *HLT-NAACL*, pages 84–92. Association for Computational Linguistics.

Young-Bum Kim, Karl Stratos, Xiaohu Liu, and Ruhi Sarikaya. 2015b. Compact lexicon selection with spectral methods. In *ACL*. Association for Computational Linguistics.

Young-Bum Kim, Karl Stratos, and Ruhi Sarikaya. 2015c. Pre-training of hidden-unit crfs. In *ACL*. Association for Computational Linguistics.

Young-Bum Kim, Karl Stratos, Ruhi Sarikaya, and Minwoo Jeong. 2015d. New transfer learning techniques for disparate label sets. In *ACL*. Association for Computational Linguistics.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Laurens Maaten, Max Welling, and Lawrence K Saul. 2011. Hidden-unit conditional random fields. In *International Conference on Artificial Intelligence and Statistics*.

Michael Mathioudakis and Nick Koudas. 2010. Twittermonitor: trend detection over the twitter stream. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 1155–1158. ACM.

Andrew McCallum and Wei Li. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *HLT-NAACL*, pages 188–191. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, 12:1532–1543.

Alan Ritter, Sam Clark, Oren 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. Association for Computational Linguistics.

Alan Ritter, Mausam, Oren Etzioni, and Sam Clark. 2012. Open domain event extraction from twitter. In *KDD*.

Karl Stratos, Do-kyum Kim, Michael Collins, and Daniel Hsu. 2014. A spectral algorithm for learning class-based n-gram models of natural language. In *UAI*.

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*, pages 384–394. Association for Computational Linguistics.