Drop-out Conditional Random Fields for Twitter with Huge Mined Gazetteer

Eunsuk Yang‡§ Young-Bum Kim‡§ Ruhi Sarikaya† Yu-Seop Kim‡

†Microsoft Corporation, Redmond, WA
‡Hallym University, South Korea
esyang219@gmail.com {ybkim, ruhi.sarikaya}@microsoft.com
yskim01@hallym.ac.kr

Abstract

In named entity recognition task especially for massive data like Twitter, having a large amount of high quality gazetteers can alleviate the problem of training data scarcity. One could collect large gazetteers from knowledge graph and phrase embeddings to obtain high coverage of gazetteers. However, large gazetteers cause a side-effect called “feature under-training”, where the gazetteer features overwhelm the context features. To resolve this problem, we propose the dropout conditional random fields, which decrease the influence of gazetteer features with a high weight. Our experiments on named entity recognition with Twitter data lead to higher F1 score of 69.38%, about 4% better than the strong baseline presented in Smith and Osborne (2006).

1 Introduction

Nowadays, people are generating tremendous amount of information on social websites. For example, more than 200 million tweets are generated everyday on Twitter (Ritter et al., 2011). Twitter has become a key news source, in addition to standard news channels. As such, social scientists are starting to pay attention to it in recent years (Bollen et al., 2011; Chung and Mustafaraj, 2011; Xu et al., 2014; Calvin et al., 2015; Baldwin et al., 2015; Bellmore et al., 2015). The traditional machine learned modeling approaches trained with small and clean general text, such as news articles, perform poorly when applied to tweets, because tweets are structurally very different from general text. Thus, it is necessary to build new models for Twitter. One could label a reasonable size of tweets to train a model for a natural language processing (NLP) application. The problem is that it is very expensive to refresh the annotated data to keep the model up-to-date, because users generate tweets in an unprecedented rate (Hachman, 2011).

An obvious solution to the problem is to develop methods of utilizing a large amount of unlabeled data. One way is to induce word embeddings in a real-valued vector space from a large number of tweets (Kim et al., 2015a; Mikolov et al., 2013; Pennington et al., 2014). It is shown that the task-specific embeddings induced on tweets provide more powerful than those created from out-of-domain texts (Owoputi et al., 2012; Anastasakos et al., 2014).

Another method is to build the task-specific gazetteers. Task-specific gazetteers make the models more general and increase their coverage for unseen events. They have been proven to be useful on a number of tasks (Smith and Osborne, 2006; Li et al., 2009; Liu and Sarikaya, 2014; Kim et al., 2015b; Kim et al., 2015c). Since gazetteers can improve modeling performance, here we more focus on how to use gazetteer more effectively. To build gazetteers with sufficient coverage for our task, we first expand gazetteers from knowledge graph and phrase embeddings.

However, since the expanded gazetteers cover significant proportions of the entities in the training data, the weight of gazetteers features are easily inflated and thus the model tends to rely too much on lexical features extracted from the gazetteers fea-
tures to assign a tag rather than the contextual features such as n-gram, a phenomenon called “feature under-training”. As a result, we often observe noticeable performance degradation at test time when the entity value does not exist in the training set or the entity dictionary.

To solve this problem, we introduce a model called dropout CRFs and compare to the combination model proposed by Smith and Osborne (2006). In our experiments, we show that the proposed method significantly improves the F1 score from 65.54% to 69.38%, compared to the baseline.

2 Model

For the named entity recognition (NER) task, the input is a sentence consisting of a sequence of words, \( x = (x_1 \ldots x_n) \) and the output is a sequence of corresponding named entity tags \( y = (y_1 \ldots y_n) \). We model the conditional probability \( p(y|x; \theta) \) using linear-chain CRFs (Lafferty et al., 2001):

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

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

\[
\theta^* = \arg\max_{\theta} \sum_{i=1}^{N} \log p(y^{(i)}|x^{(i)}; \theta) - \frac{\lambda_1}{2} ||\theta||^2 - \frac{\lambda_2}{2} \sum_{g \in G} \theta_g \text{freq}(g),
\]

where \( G \) is a set of gazetteers and \( \text{freq}(g) \) counts how many times words appear in gazetteer \( g \) from training data. In our experiments, we tuned both penalty weights for local features and for gazetteer features based on a small held-out validation set. The \( \theta_g \) is a member of model parameter \( \theta \) and each gazetteer has its own parameter \( \theta_g \). The introduced penalty decreases common gazetteers’ influence on model’s decisions. By this term, we call our model dropout CRFs. The original dropout technique removes features randomly - for each training instance, only a random subset of the features will be activated (Hinton et al., 2012; Xu and Sarikaya, 2014). While it can be perceived as a general treatment to the under-training problem, it is not specifically directed at the problem we are facing in named entity recognition (NER) task. In NER, the under-training problem is more specific - the contextual features may not get large enough weights due to the strong influence of the gazetteer features. The negative impact of such under-training is also more measurable - if a named entity is unseen, the chance of a detection error becomes much higher. Therefore, we focus on decreasing influence of specific features. For specific features, we reduce the coverage of dropout from all features to gazetteer feature through \textit{feature dependent regularization}. Also, the objective function of dropout CRFs, given in equation (2), is still convex because the equation (1) is convex and the new penalty term is linear with respect to \( \theta \). Therefore, a standard optimization algorithm finds optimal \( \theta \) without sacrificing any abilities, which original CRFs have.

\footnote{The original dropout technique is to inactivate features randomly. Here, we consider to decrease the weight of a specific feature.}
3 Features

In this section, we detail the feature templates used for our experiments. Besides basic features, we also employ part-of-speech (POS) tags, chunks, word representations and gazetteers. We run task-specific POS-tagger and chunker, which are trained on tweets annotated with Twitter-specific tags (Ritter et al., 2011) as well as standard Penn Treebank tags, of Owoputi et al. (2012) to produce POS tags and chunks. We explain the word representations and gazetteer features in the following subsections.

3.1 Basic Features

The model of Ritter et al. (2011) employs the features described in this subsection. They are composed of the following features: (1) n-grams: unigrams and bigrams, (2) capitalization, (3) three character suffix and prefix presence, (4) binary features that indicate presence of hyphen, punctuation mark, single-digit and double-digit, (5) gazetteers (6) topics inferred by LabeledLDA (Ramage et al., 2009), and (7) brown cluster (Brown et al., 1992) produced by Ritter et al. (2011).

To alleviate the problem of word sparsity, we also use task-specific latent continuous word representations, induced on 65 million unlabeled tweets with 1.3 billion tokens. We create three sets of word representations: CCA (Dhillon et al., 2012; Kim et al., 2015a) based on matrix factorization, word2vec (Mikolov et al., 2013) and glove (Pennington et al., 2014), which are gradient based. All word representation algorithms produce 50-dimensional word vectors for all words occurring at least 40 times in the data. We use left and right word of the target word as context for learning the word representations.

We also use compounding embeddings as an additional feature. Combining multiple sets of features has been proven to be effective (Koo et al., 2008; Kim and Snyder, 2013; Yu et al., 2013). We explore four different ways of combining the word representations: element-wise averaging, element-wise multiplication, concatenation and hierarchical clustering. We empirically determined that the element-wise averaging achieves better performance than single embeddings and other combination methods. We do not describe the results for embedding combinations in detail here.

4 Gazetteers

NER models degrades when they encounter unseen words during training. To make the problem worse, tweets contain many rare words and it is prohibitively expensive to create a training set with sufficient lexical coverage. To alleviate the problem, we extend the original gazetteers with two methods: gathering data from knowledge graph and constructing task-specific gazetteer with phrase embeddings.

4.1 Expansion from Knowledge Graph

To expand gazetteers from knowledge graph, we apply the following processing steps. We first extract the seed words from training data. With seed words, we then collect the relevant lexicons from knowledge graph such as Freebase, Wikipedia and Yelp. For example, “Dior” is related to company and product from knowledge graph. We collect all lexicons associated with seed words. In addition, we post-process gazetteers for variance: i) organization: it is composed with full name with abbreviation, such as “Indigenous Land Corporation (ILC)”. We also generate variants of full names (“Indigenous Land Corporation”) and abbreviation (“ILC”), respectively, ii) facility: because the term elementary indicates a school, we add a lexicon removing the word school of “tedder elementary school”. At the end of the processing, we end up with 2.7 millions lexicon items.

4.2 Constructing Gazetteers with Phrase Embeddings

We now describe how to construct task-specific gazetteer with phrase embeddings. We use canonical correlation analysis (CCA) (Hotelling, 1936) to induce vector representations for phrase embeddings. To extract candidate phrases from unlabeled Twitter data, we first count the frequency of the context words set for each token. The size of context words set ranges from 1 to 3. The context words set occurring more than 100 are used as a rule to extract candidate phrases.

Let $n$ be the number of candidate phrases extracted by rules. Let $x_1 \ldots x_n$ be the original representations of the candidate phrases itself and $y_1 \ldots y_n$ be the original representations of two words to the left and right of the candidate phrases.
We use the following definition for the original representations. Let $d$ be the number of distinct candidate phrases and $d'$ be the number of distinct context words set.

- $x_l \in \mathbb{R}^d$ is a zero vector, in which the entry corresponding to the candidate phrases of the $l$-th instance is set to 1.
- $y_l \in \mathbb{R}^{d'}$ is a zero vector, in which the entries corresponding to context words set surrounding candidate phrases are set to 1.

Using CCA, we obtain phrase embeddings $U$ with $k$-dimensional space. To train a classifier, we manually construct a training data with 5 positive and 5 negative samples, for each gazetteer. With this data, we learn a binary classifier with the phrase embeddings as a feature. Using this classifier, we predict whether the phrases fit to the gazetteers; we refer the readers to Neelakantan and Collins (2014) for details.

5 Experiments

To demonstrate the effectiveness of the dropout CRFs, we run experiments on named entity recognition task on the Twitter dataset of Baldwin et al. (2015). We refer the readers to Baldwin et al. (2015) for the details of the dataset. We split the data into 70% for training, 10% for tuning, and 20% for testing. For all the experiments presented in this section, both CRFs and dropout CRFs are trained using the L-BFGS (Liu and Nocedal, 1989).

5.1 Effectiveness of the Gazetteers

One of our contributions is to augment the size of gazetteers with knowledge graph and phrase embeddings. Table 1 represents the performance of a model with original gazetteers, which are collected by Ritter et al. (2011) from freebase (Base Gazet) and with gazetteers we extended (Our Gazet). The size of Base Gazet is 2.9 million and the size of Our Gazet is 6.6 million, which has an additional 3.7 million entries compared to the Base Gazet. The model trained Our Gazet improves the F1 score from 62.76% to 64.67%, compared to the baseline. As shown in Table 1, we believe that larger gazetteers can mitigate the “unseen words” problem by increasing the coverage of the gazetteers.

<table>
<thead>
<tr>
<th>Gazetteer</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base Gazet</td>
<td>62.76</td>
</tr>
<tr>
<td>Our Gazet</td>
<td>64.67</td>
</tr>
</tbody>
</table>

Table 1: Comparison of models with or without new gazetteers. Base Gazet is a model with gazetteers collected by Ritter et al. (2011) and Our Gazet is a model with gazetteers we constructed by augmenting the Base Gazet with additional items, using knowledge graph and phrase embeddings.

5.2 Effectiveness of the Dropout CRFs

We conducted additional experiments with the CRF model that uses Our Gazet. Table 2 shows the overall results for models with and without dropout. We compare three models: the vanilla CRFs ($\text{CRFs}_{\text{vanilla}}$), the combination model as described in Smith and Osborne (2006) ($\text{CRFs}_{\text{LOP}}$) and our dropout model ($\text{CRFs}_{\text{dropout}}$). To avoid model parameters for gazetteer features getting over-regularized, Smith and Osborne (2006) propose to train separate models with and without gazetteers. They combine predictions from the two models by using logarithmic opinion pool (LOP).

<table>
<thead>
<tr>
<th>CRFs</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{CRFs}_{\text{vanilla}}$</td>
<td>64.67</td>
</tr>
<tr>
<td>$\text{CRFs}_{\text{LOP}}$</td>
<td>65.54</td>
</tr>
<tr>
<td>$\text{CRFs}_{\text{dropout}}$</td>
<td>69.38</td>
</tr>
</tbody>
</table>

Table 2: Comparison of models with or without dropout. $\text{CRFs}_{\text{vanilla}}$ is the vanilla CRFs with all features. $\text{CRFs}_{\text{LOP}}$ is a combination of CRFs with all features except for gazetteers and CRFs with gazetteers only, using logarithmic opinion pool (LOP). $\text{CRFs}_{\text{dropout}}$ is the dropout CRFs with all features.

5.3 Analysis

While previous NER tasks mostly focus on reporting numbers on the original data set (Baldwin et al., 2015; Yang and Kim, 2015; Kim et al., 2015c), we further investigate how the tagging performance
may change, if entities are unseen at test time. To enable such analysis, we create additional test set based on the original test set by replacing each word in person and company entities with a special token, XXXXX, indicating unseen words. This new test set represents an extreme case, where none of the words contained in the gazetteers are observed in the training data.

Table 3 represents the comparison of vanilla CRF model and dropout model for unseen test. Gazetteer is helpful to resolve “unseen words” problem. Unfortunately, frequent appearance of gazetteer makes a model learn weak context feature and strong gazetteer feature. By forcing a weight of gazetteer feature low, the dropout model allows the weak context features to become strong and the large weight of gazetteer feature to become smaller. Consequently, CRF\text{dropout} shows the significant improvement compared to CRF\text{vanilla}.

<table>
<thead>
<tr>
<th>Tags</th>
<th>CRF\text{dropout}</th>
<th>CRF\text{vanilla}</th>
</tr>
</thead>
<tbody>
<tr>
<td>person</td>
<td>74.43</td>
<td>65.81</td>
</tr>
<tr>
<td>company</td>
<td>65.74</td>
<td>57.19</td>
</tr>
</tbody>
</table>

Table 3: Comparison of vanilla CRF model and dropout model for unseen test

To see a change of feature weight when we apply dropout technique, we show the feature weights for the word “cahill” of vanilla CRFs and dropout CRFs in Table 4. In vanilla CRFs, gazetteers have a strong weight compared to the context features. However, our dropout CRFs decrease the weight of gazetteer features, while making the context features larger, to steer the models’ decision in the right direction.

<table>
<thead>
<tr>
<th>CRFs\text{vanilla}</th>
</tr>
</thead>
<tbody>
<tr>
<td>people.person → I-person : 7.46</td>
</tr>
<tr>
<td>last-name.5000 → I-person : 9.63</td>
</tr>
<tr>
<td>last-name.5000 → I-geo-loc : 4.01</td>
</tr>
<tr>
<td>people.person.last-names → I-person : 6.6</td>
</tr>
<tr>
<td>w[-1]</td>
</tr>
<tr>
<td>w[-1]</td>
</tr>
<tr>
<td>w[-1]=’s → I-person : 0.97</td>
</tr>
<tr>
<td>w[-1]=’s → I-geo-loc : -0.13</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CRFs\text{dropout}</th>
</tr>
</thead>
<tbody>
<tr>
<td>people.person → I-person : 5.2</td>
</tr>
<tr>
<td>last-name.5000 → I-person : 4.67</td>
</tr>
<tr>
<td>last-name.5000 → I-geo-loc : 4.19</td>
</tr>
<tr>
<td>people.person.last-names → I-person : 4.36</td>
</tr>
<tr>
<td>w[-1]</td>
</tr>
<tr>
<td>w[-1]</td>
</tr>
<tr>
<td>w[-1]=’s → I-person : 1.41</td>
</tr>
<tr>
<td>w[-1]=’s → I-geo-loc : 1.82</td>
</tr>
</tbody>
</table>

Table 4: Snapshot of feature weights for the word “cahill”, given sentence tonight’s cahill event. The vanilla CRFs predict it to person and dropout CRFs predict it to geo-loc correctly.

6 Conclusion

In this paper, we showed how to improve the CRF based NER model for Twitter by exploiting a large number of gazetteers. Using gazetteers in modeling helps the coverage and generalization but simply incorporating gazetteers of all of large sizes into the model may lead to “under-training” of parameters corresponding to the context features. We addressed this problem by adding the dropout penalty term in the CRF training, which improves better parameter. The proposed technique results in significant improvements over the baseline.

Acknowledgments

We thank Do Kook Choe, Puyang Xu, Alan Ritter and Karl Startos for helpful discussion and feedback.

References


Timothy Baldwin, Young-Bum Kim, Marie Catherine de Marneffe, Alan Ritter, Bo Han, and Wei Xu. 2015. Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition. ACL-IJCNLP 2015, page 126.


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


Daniel Ramage, David Hall, Ramesh Nallapati, and Gökhan Tür. 2011. Learning weighted entity lists based on named entity dictionary weighting approach for spoken language understanding.


