Joint Inference of Named Entity Recognition and Normalization for Tweets

Xiaohua Liu [‡] [†], Ming Zhou [†], Furu Wei [†], Zhongyang Fu [§], Xiangyang Zhou [♯]
[‡]School of Computer Science and Technology
Harbin Institute of Technology, Harbin, 150001, China
[§]Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai, 200240, China
[♯]School of Computer Science and Technology
Shandong University, Jinan, 250100, China
[†]Microsoft Research Asia
Beijing, 100190, China
[†]{xiaoliu, fuwei, mingzhou}@microsoft.com
[§] zhongyang.fu@gmail.com [♯] v-xzho@microsoft.com

Abstract

Tweets represent a critical source of fresh information, in which named entities occur frequently with rich variations. We study the problem of named entity normalization (NEN) for tweets. Two main challenges are the errors propagated from named entity recognition (NER) and the dearth of information in a single tweet. We propose a novel graphical model to simultaneously conduct NER and NEN on multiple tweets to address these challenges. Particularly, our model introduces a binary random variable for each pair of words with the same lemma across similar tweets, whose value indicates whether the two related words are mentions of the same entity. We evaluate our method on a manually annotated data set, and show that our method outperforms the baseline that handles these two tasks separately, boosting the F1 from 80.2% to 83.6% for NER, and the Accuracy from 79.4% to 82.6% for NEN, respectively.

1 Introduction

Tweets, short messages of less than 140 characters shared through the Twitter service [1], have become an important source of fresh information. As a result, the task of named entity recognition (NER) for tweets, which aims to identify mentions of rigid designators from tweets belonging to named-entity types such as persons, organizations and locations (2007), has attracted increasing research interest. For example, Ritter et al. (2011) develop a system that exploits a CRF model to segment named entities and then uses a distantly supervised approach based on LabeledLDA to classify named entities. Liu et al. (2011) combine a classifier based on the k-nearest neighbors algorithm with a CRF-based model to leverage cross tweets information, and adopt the semi-supervised learning to leverage unlabeled tweets.

However, named entity normalization (NEN) for tweets, which transforms named entities mentioned in tweets to their unambiguous canonical forms, has not been well studied. Owing to the informal nature of tweets, there are rich variations of named entities in them. According to our investigation on the data set provided by Liu et al. (2011), every named entity in tweets has an average of 3.3 variations [2]. As an illustrative example, we show “Anneke Gronloh”, which may occur as “Mw.,Gronloh”, “Anneke Kronloh” or “Mevrouw G”. We thus propose NEN for tweets, which plays an important role in entity retrieval, trend detection, and event and entity tracking. For example, Khalid et al. (2008) show that even a simple normalization method leads to improvements of early precision, for both document and passage retrieval, and better normalization results in better retrieval performance.

Traditionally, NEN is regarded as a septated task, which takes the output of NER as its input (Li et al., 2002; Cohen, 2005; Jijkoun et al., 2008; Dai et al., 2011). One limitation of this cascaded approach is that errors propagate from NER to NEN and there is no feedback from NEN to NER. As demonstrated by Khalid et al. (2008), most NEN errors are caused

[1] http://www.twitter.com

[2] This data set consists of 12,245 randomly sampled tweets within five days.

Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, pages 526–535,
Jeju, Republic of Korea, 8-14 July 2012. ©2012 Association for Computational Linguistics

by recognition errors. Another challenge of NEN is the dearth of information in a single tweet, due to the short and noise-prone nature of tweets. Reportedly, the accuracy of a baseline NEN system based on Wikipedia drops considerably from 94% on edited news to 77% on news comments, a kind of user generated content (UGC) with similar style to tweets (Jijkoun et al., 2008).

We propose jointly conducting NER and NEN on multiple tweets using a graphical model, to address these challenges. Intuitively, improving the performance of NER boosts the performance of NEN. For example, consider the following two tweets: "$\cdots$ Alex's jokes. Justin's smartness. Max's randomnes$\cdots$ " and "$\cdots$ Alex Russo was like the best character on Disney Channel$\cdots$". Identifying "Alex" and "Alex Russo" as PERSON will encourage NEN systems to normalize "Alex" into "Alex Russo". On the other hand, NEN can guide NER. For instance, consider the following two tweets: "$\cdots$ she knew Burger King when he was a Prince!$\cdots$" and "$\cdots$ I'm craving all sorts of food: mcdonalds, burger king, pizza, chinese$\cdots$". Suppose the NEN system believes that "burger king" cannot be mapped to "Burger King" since these two tweets are not similar in content. This will help NER to assign them different types of labels. Our method optimizes these two tasks simultaneously by enabling them to interact with each other. This largely differentiates our method from existing work.

Furthermore, considering multiple tweets simultaneously allows us to exploit the redundancy in tweets, as suggested by Liu et al. (2011). For example, consider the following two tweets: "$\cdots$ Bobby Shaw you don't invite the wind$\cdots$" and "$\cdots$ I own yah ! Loool bobby shaw$\cdots$". Recognizing "Bobby Shaw" in the first tweet as a PERSON is easy owing to its capitalization and the following word "you", which in turn helps to identify "bobby shaw" in the second tweet as a PERSON.

We adopt a factor graph as our graphical model, which is constructed in the following manner. We first introduce a random variable for each word in every tweet, which represents the BILOU (Beginning, the Inside and the Last tokens of multi-token entities as well as Unit-length entities) label of the corresponding word. Then we add a factor to connect two neighboring variables, forming a conventional linear chain CRFs. Hereafter, we use t_m to denote the m^{th} tweet ,t_m^i and y_m^i to denote the i^{th} word of of t_m and its BILOU label, respectively, and f_m^i to denote the factor related to y_m^{i-1} and y_m^i. Next, for each word pair with the same lemma, denoted by t_m^i and t_n^j, we introduce a binary random variable, denoted by z_{mn}^{ij}, whose value indicates whether t_m^i and t_n^j belong to two mentions of the same entity. Finally, for any z_{mn}^{ij} we add a factor, denoted by f_{mn}^{ij}, to connect y_m^i, y_n^j and z_{mn}^{ij}. Factors in the same group ($\{f_{mn}^{ij}\}$ or $\{f_m^i\}$) share the same set of feature templates. Figure 1 illustrates an example of our factor graph for two tweets.

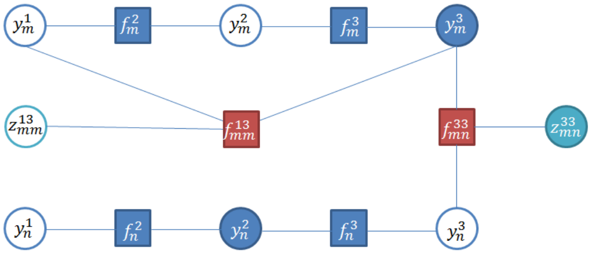

Figure 1: A factor graph that jointly conducts NER and NEN on multiple tweets. Blue and green circles represent NE type (y-$serials$) and normalization variables (z-$serials$), respectively; filled circles indicate observed random variables; blue rectangles represent the factors connecting neighboring y-$serial$ variables while red rectangles stand for the factors connecting distant y-$serial$ and z-$serial$ variables.

It is worth noting that our factor graph is different from the skip-chain CRFs (Galley, 2006) in the sense that any skip-chain factor of our model consists not only of two NE type variables (y_m^i and y_n^j), which is the case for skip-chain CRFs, but also a normalization variable (z_{mn}^{ij}). It is these normalization variables that enable us to conduct NER and NEN jointly.

We manually add normalization information to the data set shared by Liu et al. (2011), to evaluate our method. Experimental results show that our method achieves 83.6% F1 for NER and 82.6% Accuracy for NEN, outperforming the baseline with 80.2%F1 for NER and 79.4% Accuracy for NEN.

We summarize our contributions as follows.

1. We introduce the task of NEN for tweets, and propose jointly conducting NER and NEN for

multiple tweets using a factor graph, which leverages redundancy in tweets to make up for the dearth of information in a single tweet and allows these two tasks to inform each other.

2. We evaluate our method on a human annotated data set, and show that our method compares favorably with the baseline, achieving better performance in both tasks.

Our paper is organized as follows. In the next section, we introduce related work. In Section 3 and 4, we formally define the task and present our method. In Section 5, we evaluate our method. And finally we conclude our work in Section 6.

2 Related Work

Related work can be divided into two categories: NER and NEN.

2.1 NER

NER has been well studied and its solutions can be divided into three categories: 1) Rule-based (Krupka and Hausman, 1998); 2) machine learning based (Finkel and Manning, 2009; Singh et al., 2010); and 3) hybrid methods (Jansche and Abney, 2002). Owing to the availability of annotated corpora, such as ACE05, Enron (Minkov et al., 2005) and CoNLL03 (Tjong Kim Sang and De Meulder, 2003), data driven methods are now dominant.

Current studies of NER mainly focus on formal text such as news articles (Mccallum and Li, 2003; Etzioni et al., 2005). A representative work is that of Ratinov and Roth (2009), in which they systematically study the challenges of NER, compare several solutions, and show some interesting findings. For example, they show that the BILOU encoding scheme significantly outperforms the BIO schema (Beginning, the Inside and Outside of a chunk).

A handful of work on other genres of texts exists. For example, Yoshida and Tsujii build a biomedical NER system (2007) using lexical features, orthographic features, semantic features and syntactic features, such as part-of-speech (POS) and shallow parsing; Downey et al. (2007) employ capitalization cues and n-gram statistics to locate names of a variety of classes in web text; Wang (2009) introduces NER to clinical notes. A linear CRF model is trained on a manually annotated data set, which achieves an F1 of 81.48% on the test data set; Chiticariu et al. (2010) design and implement a high-level language NERL which simplifies the process of building, understanding, and customizing complex rule-based named-entity annotators for different domains.

Recently, NER for Tweets attracts growing interest. Finin et al. (2010) use Amazons Mechanical Turk service [3] and CrowdFlower [4] to annotate named entities in tweets and train a CRF model to evaluate the effectiveness of human labeling. Ritter et al. (2011) re-build the NLP pipeline for tweets beginning with POS tagging, through chunking, to NER, which first exploits a CRF model to segment named entities and then uses a distantly supervised approach based on LabeledLDA to classify named entities. Unlike this work, our work detects the boundary and type of a named entity simultaneously using sequential labeling techniques. Liu et al. (2011) combine a classifier based on the k-nearest neighbors algorithm with a CRF-based model to leverage cross tweets information, and adopt the semi-supervised learning to leverage unlabeled tweets. Our method leverages redundance in similar tweets, using a factor graph rather than a two-stage labeling strategy. One advantage of our method is that local and global information can interact with each other.

2.2 NEN

There is a large body of studies into normalizing various types of entities for formally written texts. For instance, Cohen (2005) normalizes gene/protein names using dictionaries automatically extracted from gene databases; Magdy et al. (2007) address cross-document Arabic name normalization using a machine learning approach, a dictionary of person names and frequency information for names in a collection; Cucerzan (2007) demostrates a large-scale system for the recognition and semantic disambiguation of named entities based on information extracted from a large encyclopedic collection and Web search results; Dai et al. (2011) employ a Markov logic network to model interweaved con-

[3] https://www.mturk.com/mturk/

[4] http://crowdflower.com/

straints in a setting of gene mention normalization.

Jijkoun et al. (2008) study NEN for UGC. They report that the accuracy of a baseline NEN system based on Wikipedia drops considerably from 94% on edited news to 77% on UGC. They identify three main error sources, i.e., entity recognition errors, multiple ways of referring to the same entity and ambiguous references, and exploit hand-crafted rules to improve the baseline NEN system.

We introduce the task of NEN for tweets, a new genre of texts with rich entity variations. In contrast to existing NEN systems, which take the output of NER systems as their input, our method conducts NER and NEN at the same time, allowing them to reinforce each other, as demonstrated by the experimental results.

3 Task Definition

A tweet is a short text message with no more than 140 characters. Here is an example of a tweet: “mycraftingworld: #Win Microsoft Office 2010 Home and Student #Contest from @office http://bit.ly/ $\cdots$ ”, where “mycraftingworld” is the name of the user who published this tweet. Words beginning with “#” like “”#Win” are hash tags; words starting with “@” like “@office” represent user names; and “http://bit.ly/” is a shortened link.

Given a set of tweets, e.g., tweets within some period or related to some query, our task is: 1) To recognize each mention of entities of predefined types for each tweet; and 2) to restore each entity mention into its unambiguous canonical form. Following Liu et al. (2011), we focus on four types of entities, i.e., PERSON, ORGANIZATION, PRODUCT, and LOCATION, and constrain our scope to English tweets. Note that the NEN sub-task can be transformed as follows. Given each pair of entity mentions, decide whether they denote the same entity. Once this is achieved, we can link all the mentions of the same entity, and choose a representative mention, e.g., the longest mention, as their canonical form.

As an illustrative example, consider the following three tweets: “$\cdots$Gaga’s Christmas dinner with her family. Awwwwn$\cdots$”, “$\cdots$Lady Gaaaaga with her family on Christmas$\cdots$” and “$\cdots$Buying a magazine just because Lady Gaga’s on the cover$\cdots$”. It is expected that “Gaga”, “Lady Gaaaaga” and “Lady Gaga” are all labeled as PERSON, and can be restored as “Lady Gaga”.

4 Our Method

In contrast to existing work, our method jointly conducts NER and NEN for multiple tweets. We first give an overview of our method, then detail its model and features.

4.1 Overview

Given a set of tweets as input, our method recognizes predefined types of named entities and for each entity outputs its unambiguous canonical form.

To resolve NER, we assign a label to each word in a tweet, indicating both the boundary and entity type. Following Ratinov and Roth (2009), we use the BILOU schema. For example, consider the tweet “$\cdots$without you is like an iphone without apps; Lady gaga without her telephone$\cdots$”, the labeled sequence using the BILOU schema is: “$\cdots$without$_O$ you$_O$ is$_O$ like$_O$ an$_O$ iphone$_{U-PRODUCT}$ without$_O$ apps$_O$; Lady$_{B-PERSON}$ gaga$_{L-PERSON}$ without$_O$ her$_O$ telephone$_O\cdots$” , where “iphone$_{U-PRODUCT}$” indicates that “iphone” is a product name of unit length; “Lady$_{B-PERSON}$” means “Lady” is the beginning of a person name while “gaga$_{L-PERSON}$” suggests that “gaga” is the last token of a person name.

To resolve NEN, we assign a binary value label z_{mn}^{ij} to each pair of words t_m^i and t_n^j which share the same lemma. $z_{mn}^{ij} = 1$ or -1, indicating whether t_m^i and t_n^j belong to two mentions of the same entity [5]. For example, consider the three tweets presented in Section 3. “Gaga$_1^1$” [6] and “Gaga$_3^1$” will be assigned a “1” label, since they are part of two mentions of the same entity “Lady Gaga”; similarly, “Lady$_2^1$” and “Lady$_3^1$” are connected with a “1” label. Note that there are no NEN labels for pairs like “her$_1^1$” and “her$_2^1$” or “with$_1^1$ and “with$_2^1$”, since words like “her” and “with” are stop words.

With NE type and normalization labels obtained, we judge two mentions, denoted by $t_m^{i_1 \cdots i_k}$ and

[5] Stop words have no normalization labels. The stop words are mainly from http://www.textfixer.com/resources/common-english-words.txt.

[6] We use w$_m^i$ to denote word w’s i^{th} appearance in the m^{th} tweet. For example, “Gaga$_1^1$” denotes the first occurance of “Gaga” in the first tweet.

$t_n^{j_1\cdots j_l}$, respectively, refer to the same entity if and only if: 1) The two mentions share the same entity type; 2) $t_m^{i_1\cdots i_k}$ is a sub-string of $t_n^{j_1\cdots j_l}$ or vise versa; and 3) $z_{mn}^{ij} = 1, i = i_1, \cdots, i_k$ and $j = j_1, \cdots, j_l$, if z_{mn}^{ij} exists. Still take the three tweets presented in Section 3 for example. Suppose "Gaga$_1^1$" and "Lady Gaga$_3^1$" are labeled as PERSON, and there is only one related NE normalization label, which is associated with "'Gaga$_1^1$" and "Gaga$_3^1$" and has 1 as its value. We then consider that these two mentions can be normalized into the same entity; in a similar way, we can align "Lady$_2^1$ Gaaaaga" with "Lady$_3^1$ Gaga". Combining these pieces information together, we can infer that "'Gaga$_1^1$", "Lady$_2^1$ Gaaaaga" and "Lady$_3^1$ Gaga" are three mentions of the same entity. Finally, we can select 'Lady$_3^1$ Gaga" as the representative, and output 'Lady Gaga" as their canonical form. We choose the mention with the maximum number of words as the representative. In case of a tie, we prefer the mention with an Wikipedia entry [7].

The central problem with our method is inferring all the NE type (y-$serial$) and normalization (z-$serial$) variables. To achieve this, we construct a factor graph according to the input tweets, which can evaluate the probability of every possible assignment of y-$serials$ and z-$serials$, by checking the characteristics of the assignment. Each characteristic is called a feature. In this way, we can select the assignment with the highest probability. Next we will introduce our model in detail, including its training and inference procedure and features.

4.2 Model

We adopt a factor graph as our model. One advantage of our model is that it allows y-$serials$ and z-$serials$ variables to interact with each other to jointly optimize NER and NEN.

Given a set of tweets $T = \{t_m\}_{m=1}^N$, we can build a factor graph $\mathcal{G} = (Y, Z, F, E)$, where: Y and Z denote y-$serials$ and z-$serials$ variables, respectively; F represents factor vertices, consisting of $\{f_m^i\}$ and $\{f_{mn}^{ij}\}$, $f_m^i = f_m^i(y_m^{i-1}, y_m^i)$ and $f_{mn}^{ij} = f_{mn}^{ij}(y_m^i, y_n^j, z_{mn}^{ij})$; E stands for edges, which depends on F, and consists of edges between y_m^{i-1} and y_m^i, and those between y_m^i,y_n^j and f_{mn}^{ij}.

$\mathcal{G} = (Y, Z, F, E)$ defines a probability distribution according to Formula 1.

$$\ln P(Y,Z|\mathcal{G},T) \propto \sum_{m,i} \ln f_m^i(y_m^{i-1}, y_m^i) + \sum_{m,n,i,j} \delta_{mn}^{ij} \cdot \ln f_{mn}^{ij}(y_m^i, y_n^j, z_{mn}^{ij}) \tag{1}$$

where $\delta_{mn}^{ij} = 1$ if and only if t_m^i and t_n^j have the same lemma and are not stop words, otherwise zero. A factor factorizes according to a set of features, so that:

$$\ln f_m^i(y_m^{i-1}, y_m^i) = \sum_k \lambda_k^{(1)} \phi_k^{(1)}(y_m^{i-1}, y_m^i)$$
$$\ln f_{mn}^{ij}(y_m^i, y_n^j, z_{mn}^{ij}) = \sum_k \lambda_k^{(2)} \phi_k^{(2)}(y_m^i, y_n^j, z_{mn}^{ij}) \tag{2}$$

$\{\phi_k^{(1)}\}_{k=1}^{K_1}$ and $\{\phi_k^{(2)}\}_{k=1}^{K_2}$ are two feature sets. $\Theta = \{\lambda_k^{(1)}\}_{k=1}^{K_1} \bigcup \{\lambda_k^{(2)}\}_{k=1}^{K_2}$ is called the feature weight set or parameter set of $\mathcal{G}$. Each feature has a real value as its weight.

Training Θ is learnt from annotated tweets T, by maximizing the data likelihood, i.e.,

$$\Theta^* = \arg\max_{\Theta} \ln P(Y, Z|\Theta, T) \tag{3}$$

To solve this optimization problem, we first calculate its gradient:

$$\frac{\partial \ln P(Y,Z|T;\Theta)}{\partial \lambda_k^1} = \sum_{m,i} \phi_k^{(1)}(y_m^{i-1}, y_m^i) - \sum_{m,i} \sum_{y_m^{i-1}, y_m^i} p(y_m^{i-1}, y_m^i|T;\Theta) \phi_k^{(1)}(y_m^{i-1}, y_m^i) \tag{4}$$

$$\frac{\partial \ln P(Y,Z|T;\Theta)}{\partial \lambda_k^2} = \sum_{m,n,i,j} \delta_{mn}^{ij} \cdot \phi_k^{(2)}(y_m^i, y_n^j, z_{mn}^{ij}) - \sum_{m,n,i,j} \delta_{mn}^{ij} \sum_{y_m^i, y_n^j, z_{mn}^{ij}} p(y_m^i, y_n^j, z_{mn}^{ij}|T;\Theta) \cdot \phi_k^{(2)}(y_m^i, y_n^j, z_{mn}^{ij}) \tag{5}$$

Here, the two marginal probabilities $p(y_m^{i-1}, y_m^i|T;\Theta)$ and $p(y_m^i, y_n^j, z_{mn}^{ij}|T;\Theta)$ are computed using loopy belief propagation (Murphy et al., 1999). Once we have computed the gradient, Θ^* can be worked out by standard techniques such as steepest descent, conjugate gradient and the

[7] If it still ends up as a draw, we will randomly choose one from the best.

limited-memory BFGS algorithm (L-BFGS). We choose L-BFGS because it is particularly well suited for optimization problems with a large number of variables.

Inference Supposing the parameters Θ have been set to Θ^*, the inference problem is: Given a set of testing tweets T, output the most probable assignment of Y and Z, i.e.,

$$(Y,Z)^* = \arg\max_{(Y,Z)} \ln P(Y,Z|\Theta^*,T) \tag{6}$$

We adopt the max-product algorithm to solve this inference problem. The max-product algorithm is nearly identical to the loopy belief propagation algorithm, with the sums replaced by maxima in the definitions. Note that in both the training and testing stage, the factor graph is constructed in the same way as described in Section 1.

Efficiency We take several actions to improve our model's efficiency. Firstly, we manually compile a comprehensive named entity dictionary from various sources including Wikipedia, Freebase [8], news articles and the gazetteers shared by Ratinov and Roth (2009). In total this dictionary contains 350 million entries [9]. By looking up this dictionary [10], we generate the possible BILOU labels, denoted by Y_m^i hereafter, for each word t_m^i. For instance, consider “$\cdots$ Good Morning new$_1^1$ york$_1^1$$\cdots$”. Suppose “New York City” and “New York Times” are in our dictionary, then “new$_1^1$ york$_1^1$” is the matched string with two corresponding entities. As a result, “B-LOCATION” and “B-ORGANIZATION” will be added to $Y_{new_1^1}$, and “I-LOCATION” and “I-ORGANIZATION” will be added to $Y_{york_1^1}$. If $Y_m^i \neq \varnothing$, we enforce the constraint for training and testing that $y_m^i \in Y_m^i$, to reduce the search space.

Secondly, in the testing phase, we introduce three rules related to z_{mn}^{ij}: 1) $z_{mm}^{ij} = 1$, which says two words sharing the same lemma in the same tweet denote the same entity; 2) set z_{mn}^{ij} to 1, if the similarity between t_m and t_n is above a threshold (0.8 in our work), or t_m and t_n share one hash tag; and 3)$z_{mn}ij = -1$, if the similarity between t_m and t_n is below a threshold (0.3 in work). To compute the similarity, each tweet is represented as a bag-of-words vector with the stop words removed, and the cosine similarity is adopted, as defined in Formula 7. These rules pre-label a significant part of $z\text{-}serial$ variables (accounting for 22.5%), with an accuracy of 93.5%.

$$sim(t_m,t_n) = \frac{\vec{t_m}\cdot\vec{t_n}}{|\vec{t_m}||\vec{t_n}|} \tag{7}$$

Note that in our experiments, these measures reduce the training and testing time by 36.2% and 62.8%, respectively, while no obvious performance drop is observed.

4.3 Features

A feature in $\{\phi_k^{(1)}\}_{k=1}^{K_1}$ involves a pair of neighboring NE-type labels, i.e., y_m^{i-1} and y_m^i, while a feature in $\{\phi_k^{(2)}\}_{k=1}^{K_2}$ concerns a pair of distant NE-type labels and its associated normalization label, i.e., y_m^i,y_n^j and z_{mn}^{ij}. Details are given below.

4.3.1 Feature Set One: $\{\phi_k^{(1)}\}_{k=1}^{K_1}$

We adopts features similar to Wang (2009), and Ratinov and Roth (2009), i.e., orthographic features, lexical features and gazetteer-related features. These features are defined on the observation. Combining them with y_m^{i-1} and y_m^i constitutes $\{\phi_k^{(1)}\}_{k=1}^{K_1}$.

Orthographic features: Whether t_m^i is capitalized or upper case; whether it is alphanumeric or contains any slashes; wether it is a stop word; word prefixes and suffixes.

Lexical features: Lemma of t_m^i, t_m^{i-1} and t_m^{i+1}, respectively; whether t_m^i is an out-of-vocabulary (OOV) word [11]; POS of t_m^i, t_m^{i-1} and t_m^{i+1}, respectively; whether t_m^i is a hash tag, a link, or a user account.

Gazetteer-related features: Whether Y_m^i is empty; the dominating label/entity type in Y_m^i. Which one is dominant is decided by majority voting of the entities in our dictionary. In case of a tie, we randomly choose one from the best.

4.3.2 Feature Set Two: $\{\phi_k^{(2)}\}_{k=1}^{K_2}$

Similarly, we define orthographic, lexical features and gazetteer-related features on the observation, y_m^i

[8] http://freebase.com/view/military

[9] One phrase refereing to L entities has L entries.

[10] We use case-insensitive leftmost longest match.

[11] We first conduct a simple dictionary-lookup based normalization with the incorrect/correct word pair list provided by Han et al. (2011) to correct common ill-formed words. Then we call an online dictionary service to judge whether a word is OOV.

and y_n^j; and then we combine these features with z_{mn}^{ij}, forming $\{\phi_k^{(2)}\}_{k=1}^{K_2}$.

Orthographic features: Whether t_m^i / t_n^j is capitalized or upper case; whether t_m^i / t_n^j is alphanumeric or contains any slashes; prefixes and suffixes of t_m^i.

Lexical features: Lemma of t_m^i; whether t_m^i is OOV; whether t_m^i / t_m^{i+1} / t_m^{i-1} and t_n^j / t_n^{j+1} / t_n^{j-1} have the same POS; whether y_m^i and y_n^j have the same label/entity type.

Gazetteer-related features: Whether $Y_m^i \bigcap Y_n^j$ / $Y_m^{i+1} \bigcap Y_n^{j+1}$ / $Y_m^{i-1} \bigcap Y_n^{j-1}$ is empty; whether the dominating label/entity type in Y_m^i is the same as that in Y_n^j.

5 Experiments

We manually annotate a data set to evaluate our method. We show that our method outperforms the baseline, a cascaded system that conducts NER and NEN individually.

5.1 Data Preparation

We use the data set provided by Liu et al. (2011), which consists of 12,245 tweets with four types of entities annotated: PERSON, LOCATION, ORGANIZATION and PRODUCT. We enrich this data set by adding entity normalization information. Two annotators [12] are involved. For any entity mention, two annotators independently annotate its canonical form. The inter-rater agreement measured by kappa is 0.72. Any inconsistent case is discussed by the two annotators till a consensus is reached. $2,245$ tweets are used for development, and the remainder are used for 5-fold cross validation.

5.2 Evaluation Metrics

We adopt the widely-used Precision, Recall and F1 to measure the performance of NER for a particular type of entity, and the average Precision, Recall and F1 to measure the overall performance of NER (Liu et al., 2011; Ritter et al., 2011). As for NEN, we adopt the widely-used Accuracy, i.e., to what percentage the outputted canonical forms are correct (Jijkoun et al., 2008; Cucerzan, 2007; Li et al., 2002).

5.3 Baseline

We develop a cascaded system as the baseline, which conducts NER and NEN sequentially. Its NER module, denoted by S_{BR}, is based on the state-of-the-art method introduced by Liu et al. (2011); and its NEN model , denoted by S_{BN}, follows the NEN system for user-generated news comments proposed by Jijkoun et al. (2008), which uses handcrafted rules to improve a typical NEN system that normalizes surface forms to Wikipedia page titles. We use the POS tagger developed by Ritter et al. (2011) to extract POS related features, and the OpenNLP toolkit to get lemma related features.

5.4 Results

Tables 1- 2 show the overall performance of the baseline and ours (denoted by S_{RN}). It can be seen that, our method yields a significantly higher F1 (with $p < 0.01$) than S_{BR}, and a moderate improvement of accuracy as compared with S_{BN} (with $p < 0.05$). As a case study, we show that our system successfully identified "jaxon$_1^1$" as a PERSON in the tweet "$\cdots$ come to see jaxon$_1^1$ someday$\cdots$", which is mistakenly labeled as a LOCATION by S_{BR}. This is largely owing to the fact that our system aligns "jaxon$_1^1$" with "Jaxson$_2^1$" in the tweet "$\cdots$ I love Jaxson$_2^1$,Hes like my little brother$\cdots$", in which "Jaxson$_2^1$" is identified as a PERSON. As a result, this encourages our system to consider "jaxon$_1^1$" as a PERSON. We also find cases where our system works but S_{BN} fails. For example, "Goldman$_1^1$" in the tweet "$\cdots$ Goldman sees massive upside risk in oil prices$\cdots$" is normalized into "Albert Goldman" by S_{BR}, because it is mistakenly identified as a PERSON by S_{BS}; in contrast, our system recognizes "Goldman$_2^1$ Sachs" as an ORGANIZATION, and successfully links 'Goldman$_2^1$" to "Goldman$_1^1$", resulting that "Goldman$_1^1$" is identified as an ORGANIZATION and normalized into "Goldman Sachs".

Table 3 reports the NER performance of our method for each entity type, from which we see that our system consistently yields better F1 on all entity types than S_{BR}. We also see that our system boosts the F1 for ORGANIZATION most significantly, reflecting the fact that a large number of organizations that are incorrectly labeled as PERSON by S_{BR}, are now correctly recognized by our method.

[12] Two native English speakers.

System	Pre	Rec	F1
S_{RN}	84.7	82.5	83.6
S_{BR}	81.6	78.8	80.2

Table 1: Overall performance (%) of NER.

System	Accuracy
S_{RN}	82.6
S_{BN}	79.4

Table 2: Overall Accuracy (%) of NEN .

System	PER	PRO	LOC	ORG
S_{RN}	84.2	80.5	82.1	85.2
S_{BR}	83.9	78.7	81.3	79.8

Table 3: F1 (%) of NER on different entity types.

Features	NER (F1)	NEN (Accuracy)
F_o	59.2	61.3
$F_o + F_l$	65.8	68.7
$F_o + F_g$	80.1	77.2
$F_o + F_l + F_g$	83.6	82.6

Table 4: Overall F1 (%) of NER and Accuracy (%) of NEN with different feature sets.

Table 4 shows the overall performance of our method with various feature set combinations, where F_o, F_l and F_g denote the orthographic features, the lexical features, and the gazetteer-related features, respectively. From Table 4 we see that gazetteer-related features significantly boost the F1 for NER and Accuracy for NEN, suggesting the importance of external knowledge for this task.

5.5 Discussion

One main error source for NER and NEN, which accounts for more than half of all the errors, is slang expressions and informal abbreviations. For instance, our method recognizes "California$_1^1$" in the tweet "$\cdots$And Now, He Lives All The Way In California$_1^1$$\cdots$" as a LOCATION, however, it mistakenly identifies "Cali$_2^1$" in the tweet "$\cdots$i love Cali so much$\cdots$" as a PERSON. One reason is our system does not generate any $z\text{-}serial$ variable for "California$_1^1$" and "Cali$_2^1$" since they have different lemmas. A more complicated case is "BS$_1^1$" in the tweet "$\cdots$I, bobby shaw, am gonna put BS$_1^1$ on everything$\cdots$", in which "BS$_1^1$" is the abbreviation of "bobby shaw". Our method fails to recognize "BS$_1^1$" as an entity. There are two possible ways to fix these errors: 1) Extending the scope of $z\text{-}serial$ variables to each word pairs with a common prefix; and 2) developing advanced normalization components to restore such slang expressions and informal abbreviations into their canonical forms.

Our method does not directly exploit Wikipedia for NEN. This explains the cases where our system correctly links multiple entity mentions but fails to generate canonical forms. Take the following two tweets for example: "$\cdots$nitip link win7$_1^1$ sp1$\cdots$" and "$\cdots$Hit the 3TB wall on SRT installing fresh Win7$_2^1$$\cdots$". Our system recognizes "win7$_1^1$" and "Win7$_2^1$" as two mentions of the same product, but cannot output their canonical forms "Windows 7". One possible solution is to exploit Wikipedia to compile a dictionary consisting of entities and their variations.

6 Conclusions and Future work

We study the task of NEN for tweets, a new genre of texts that are short and prone to noise. Two challenges of this task are the dearth of information in a single tweet and errors propagated from the NER component. We propose jointly conducting NER and NEN for multiple tweets using a factor graph, to address these challenges. One unique characteristic of our model is that a NE normalization variable is introduced to indicate whether a word pair belongs to the mentions of the same entity. We evaluate our method on a manually annotated data set. Experimental results show our method yields better F1 for NER and Accuracy for NEN than the state-of-the-art baseline that conducts two tasks sequentially.

In the future, we plan to explore two directions to improve our method. First, we are going to develop advanced tweet normalization technologies to resolve slang expressions and informal abbreviations. Second, we are interested in incorporating knowledge mined from Wikipedia into our factor graph.

Acknowledgments

We thank Yunbo Cao, Dongdong Zhang, and Mu Li for helpful discussions, and the anonymous reviewers for their valuable comments.

References

Laura Chiticariu, Rajasekar Krishnamurthy, Yunyao Li, Frederick Reiss, and Shivakumar Vaithyanathan. 2010. Domain adaptation of rule-based annotators for named-entity recognition tasks. In *EMNLP*, pages 1002–1012.

Aaron Cohen. 2005. Unsupervised gene/protein named entity normalization using automatically extracted dictionaries. In *Proceedings of the ACL-ISMB Workshop on Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics*, pages 17–24, Detroit, June. Association for Computational Linguistics.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on wikipedia data. In *In Proc. 2007 Joint Conference on EMNLP and CNLL*, pages 708–716.

Hong-Jie Dai, Richard Tzong-Han Tsai, and Wen-Lian Hsu. 2011. Entity disambiguation using a markov-logic network. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 846–855, Chiang Mai, Thailand, November. Asian Federation of Natural Language Processing.

Doug Downey, Matthew Broadhead, and Oren Etzioni. 2007. Locating Complex Named Entities in Web Text. In *IJCAI*.

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.*, 165(1):91–134.

Tim Finin, Will Murnane, Anand Karandikar, Nicholas Keller, Justin Martineau, and Mark Dredze. 2010. Annotating named entities in twitter data with crowdsourcing. In *CSLDAMT*, pages 80–88.

Jenny Rose Finkel and Christopher D. Manning. 2009. Nested named entity recognition. In *EMNLP*, pages 141–150.

Michel Galley. 2006. A skip-chain conditional random field for ranking meeting utterances by importance. In *Association for Computational Linguistics*, pages 364–372.

Bo Han and Timothy Baldwin. 2011. Lexical normalisation of short text messages: Makn sens a #twitter. In *ACL HLT*.

Martin Jansche and Steven P. Abney. 2002. Information extraction from voicemail transcripts. In *EMNLP*, pages 320–327.

Valentin Jijkoun, Mahboob Alam Khalid, Maarten Marx, and Maarten de Rijke. 2008. Named entity normalization in user generated content. In *Proceedings of the second workshop on Analytics for noisy unstructured text data*, AND '08, pages 23–30, New York, NY, USA. ACM.

Mahboob Khalid, Valentin Jijkoun, and Maarten de Rijke. 2008. The impact of named entity normalization on information retrieval for question answering. In Craig Macdonald, Iadh Ounis, Vassilis Plachouras, Ian Ruthven, and Ryen White, editors, *Advances in Information Retrieval*, volume 4956 of *Lecture Notes in Computer Science*, pages 705–710. Springer Berlin / Heidelberg.

George R. Krupka and Kevin Hausman. 1998. Isoquest: Description of the netowl^{TM} extractor system as used in muc-7. In *MUC-7*.

Huifeng Li, Rohini K. Srihari, Cheng Niu, and Wei Li. 2002. Location normalization for information extraction. In *COLING*.

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

Walid Magdy, Kareem Darwish, Ossama Emam, and Hany Hassan. 2007. Arabic cross-document person name normalization. In *In CASL Workshop 07*, pages 25–32.

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.

Einat Minkov, Richard C. Wang, and William W. Cohen. 2005. Extracting personal names from email: applying named entity recognition to informal text. In *HLT*, pages 443–450.

Kevin P. Murphy, Yair Weiss, and Michael I. Jordan. 1999. Loopy belief propagation for approximate inference: An empirical study. In *In Proceedings of Uncertainty in AI*, pages 467–475.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30:3–26.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *CoNLL*, pages 147–155.

Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. 2011. Named entity recognition in tweets: An experimental study. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

Sameer Singh, Dustin Hillard, and Chris Leggetter. 2010. Minimally-supervised extraction of entities from text advertisements. In *HLT-NAACL*, pages 73–81.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: language-independent named entity recognition. In *HLT-NAACL*, pages 142–147.

Yefeng Wang. 2009. Annotating and recognising named entities in clinical notes. In *ACL-IJCNLP*, pages 18–26.

Kazuhiro Yoshida and Jun'ichi Tsujii. 2007. Reranking for biomedical named-entity recognition. In *BioNLP*, pages 209–216.