DLocRL: A Deep Learning Pipeline for Fine-Grained Location Recognition and Linking in Tweets

Canwen Xu
Wuhan University
School of Computer Science
Hubei, China
xucanwen@whu.edu.cn

Jing Li
Inception Institute of Artificial Intelligence
United Arab Emirates
jing.li@inceptioniai.org

Xiangyang Luo*
State Key Lab of Mathematical Engineering and Advanced Computing
Henan, China
xiangyangluo@126.com

Jiaxin Pei
Wuhan University
School of Computer Science
Hubei, China
pedropei@whu.edu.cn

Chenliang Li
Wuhan University
School of Cyber Science and Engineering
Hubei, China
cllee@whu.edu.cn

Donghong Ji
Wuhan University
School of Cyber Science and Engineering
Hubei, China
dhji@whu.edu.cn

ABSTRACT

In recent years, with the prevalence of social media and smart devices, people causally reveal their locations such as shops, hotels, and restaurants in their tweets. Recognizing and linking such fine-grained location mentions to well-defined location profiles are beneficial for retrieval and recommendation systems. In this paper, we propose **DLocRL**, a new deep learning pipeline for fine-grained location recognition and linking in tweets, and verify its effectiveness on a real-world Twitter dataset.

CCS CONCEPTS

• **Information systems → Information extraction**.

KEYWORDS

POI recognition and linking; social media content analysis; named entity recognition; entity linking

ACM Reference Format:
Canwen Xu, Jing Li, Xiangyang Luo, Jiaxin Pei, Chenliang Li, and Donghong Ji. 2019. DLocRL: A Deep Learning Pipeline for Fine-Grained Location Recognition and Linking in Tweets. In *Proceedings of the 2019 World Wide Web Conference (WWW'19), May 13–17, 2019, San Francisco, CA, USA*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3308558.3313491

1 INTRODUCTION

Twitter is a place where users can share their daily life activities by posting tweets (*i.e.,* short messages, up to 140 characters each). In many tweets, locations are implicitly or causally revealed by users at fine-grained granularity [14, 16, 17], for example, a restaurant, a shopping mall, a park or a landmark building. Here, a fine-grained location is equivalent to a point-of-interest (POI), which is a focused geographical entity [16, 26]. In this paper, we target on *recognizing* mentions of POIs and *linking* these mentions to well-defined location profiles.

The two tasks are important for several reasons. First, recognizing mentions of POIs is beneficial for information retrieval [2]. An example is that query understanding can be enhanced by exploiting POI information in location-based information retrieval systems [7, 28]. In addition, recognized POIs can be integrated into knowledge bases and support many business intelligence applications such as POI recommendation and location-aware advertising [16, 20, 22]. Second, the linked location profile can serve as side information for the tweet, and vice versa. For example, Twitter sentiment analysis can be conducted in a more precise manner by incorporating the location profile content [10].

However, recognizing POI mentions and linking these mentions to well-defined location profiles are both challenging. Due to the informal writing of tweets, POIs are usually mentioned by incomplete name, nickname, acronym or misspellings. For example, the mention *vivo* may refer to *VivoCity* which is a comprehensive shopping mall in Singapore, or a smartphone brand. Even if we have successfully recognized a POI mention, it remains challenging to link the mention to a specified location profile (*i.e.,* a specific shopping mall with address and geo-coordinates in this case).

On the other hand, existing solutions [10, 14, 16] on these two tasks require a large set of features manually designed for each task and domain, which demands task and domain expertise. For example, Li *et al.* [16] carefully designed lexical, grammatical, geographical and BILOU schema features (totally 11 features) to extract fine-grained locations from tweets. Ji *et al.* [14] proposed a joint model to recognize and link fine-grained locations from tweets with 24 hand-crafted features. Recently, Han *et al.* [10] proposed a probabilistic model with seven types of hand-crafted features to link fine-grained location in user comments. It is now generally admitted that distributed representations could better capture lexical semantics [8]. We would envision for a system that is based on

*Corresponding author.

This paper is published under the Creative Commons Attribution 4.0 International (CC-BY 4.0) license. Authors reserve their rights to disseminate the work on their personal and corporate Web sites with the appropriate attribution.
WWW '19, May 13–17, 2019, San Francisco, CA, USA
© 2019 IW3C2 (International World Wide Web Conference Committee), published under Creative Commons CC-BY 4.0 License.
ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313491

distributed representations, and that can learn informative features for location recognition and linking by itself without human effort.

In this paper, we propose DLocRL, a new **D**eep pipeline for fine-grained **Loc**ation **R**ecognition and **L**inking in tweets. DLocRL is designed to adopt effective representation learning, semantic composition, and attention and gate mechanisms to exploit the multiple semantic context features for location recognition and linking. DLocRL consists of two core modules: *recognition* module and *linking* module.

The *recognition* module aims to extract a text segment referring to a fine-grained location (*i.e.,* POI) from a given tweet. In DLocRL, we formulate the recognition task as a sequence labeling problem (*i.e.,* assigning tags to tokens in the given sequence). We adopt bi-directional long short-term memory with conditional random fields output layer (BiLSTM-CRF) to train a POI recognizer.

The *linking* module aims to link each recognized POI to a corresponding location profile. Before linking POIs to location profiles, an extensive collection of location profiles (CLP) is constructed. Given an input pair ⟨POI, Profile⟩, the *linking* module is trained to judge whether the location profile corresponds to the POI. More specifically, DLocRL utilizes two pairs of parallel LSTMs to encode the left context and right context for a POI, respectively. The location profiles are represented by their constituents with TF-IDF weights and one-hot schema. The Manhattan distance is used to measure the geographical distances between users and profiles. Finally, the representation of ⟨POI, Profile⟩ comes from three sources: tweet-level contextual information, location profile representation and geographical distance. The representation is then fed into a fully-connected layer for final linking prediction. Moreover, to take advantage of the geographical coherence information among mentioned POIs in the same tweet, we develop Geographical Pair Linking (Geo-PL), a post-processing strategy to further enhance the linking accuracy.

In summary, we make the following contributions: (1) We propose **DLocRL**, a new deep learning pipeline for fine-grained Location Recognition and Linking in tweets. To the best of our knowledge, our work is the first attempt to address location recognition and linking in the paradigm of deep learning. (2) We develop the Geographical Pair Linking (Geo-PL) approach, a post-processing strategy to further improve linking performance. (3) We conduct extensive experiments on a real-world Twitter dataset. The experimental results show the effectiveness of DLocRL on fine-grained location recognition and linking. We also conduct ablation studies to validate the effectiveness of each design choice.

2 RELATED WORK

2.1 Location Mention Recognition and Disambiguation

Recognition. Facing noisy and short tweets, traditional NER methods suffer from their unreliable linguistic features [36]. Prior solutions exploit comprehensive linguistic features like Part-of-Speech tags, capitalization [27], Brown clustering [3] to improve recognition performance. For example, Li *et al.* [16] carefully designed lexical, grammatical, geographical and BILOU schema features (totally 11 features) to extract fine-grained locations from tweets. Ji *et al.* [14] proposed a joint model to recognize and link fine-grained locations from tweets with 24 hand-crafted features. Zhang *et al.* [34] designed four types of features and trained a classifier to select the best match among gazetteer candidates. Generally, location gazetteers are widely used in location mention recognition [16, 21, 34]. For noisy text, Malmasi *et al.* [21] proposed an approach based on Noun Phrase extraction and n-gram based matching instead of the traditional methods using NER or CRF. Different from these existing works, our approach DLocRL does not require hand-crafted features.

Disambiguation. For entity disambiguation, many studies attempt to exploit the coherence among mentioned entities. Pair-wise [15] and global collective methods [11, 12, 32] have been applied to explore the coherence. Zhang *et al.* [34] and Ji *et al.* [14] observed that the geographical coherence is effective in location disambiguation task. Also, instead of tweet-level coherence, Li *et al.* [18] utilized user-level coherence to facilitate entity disambiguation. Besides, Shen *et al.* [30] introduced user interest modeling to collective disambiguation.

To form an entirely end-to-end model for joint recognition and linking, Guo *et al.* [9] and Ji *et al.* [14] adopted structural SVM and perceptron, respectively. Furthermore, the beam search algorithm [35] is also used in [14] to search the best combination of the mention recognition and linking. In this work, we devise DLocRL, a new pipeline which can be upgraded and optimized much easier than previous joint models and still achieves better performance.

2.2 Neural Networks for NER and EL

Named Entity Recognition (NER). The use of neural models for NER was pioneered by Collobert *et al.* [5], where an architecture based on temporal convolutional neural networks (CNNs) over word sequence was proposed. Chiu *et al.* [4] utilized CNN to detect character-level features and LSTM to capture the word-level context in a sentence. Yang *et al.* [33] proposed a gated recurrent unit (GRU) network to learn useful morphological representation from the character sequence of a word. Recently, Peters *et al.* [24] proposed deep contextualized word representation, which can model syntax, semantics, and polysemy. Akbik *et al.* [1] proposed contextual string embeddings, to leverage the internal states of a trained character language model to produce a novel type of word embedding.

Entity Linking (EL). Neural networks were firstly used for entity linking by Sun *et al.* [31]. They proposed a model which takes consideration of the semantic representations of mention, context, and entity. Recently, Phan *et al.* [25] proposed a neural network for entity linking with LSTM and attention mechanism. They also proposed *Pair Linking* to enhance collective linking by measuring the cosine similarity of the text embeddings between two mentioned entities.

3 MODEL DESCRIPTION

Figure 1 shows the workflow of DLocRL, which consists of recognition module and location linking module.

3.1 Location Recognition

Figure 2 shows the architecture of the location recognition module. The representation of a word consists of its pre-trained word embedding, BILOU pre-label and character-level representation.

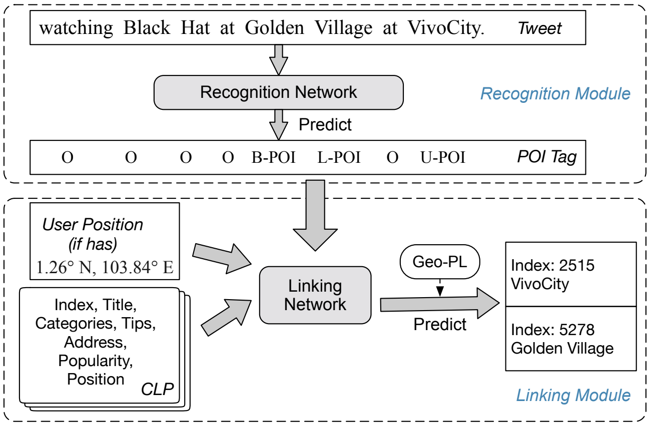

Figure 1: A running example to illustrate the workflow of DLocRL.

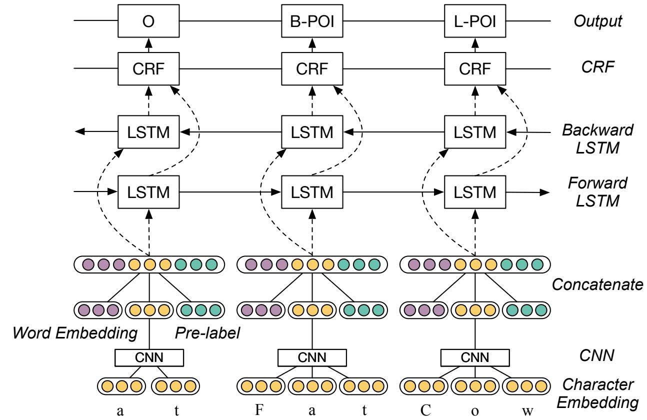

Figure 2: The architecture of location recognition module.

Finally, BiLSTM-CRF is utilized to infer tag sequence based on the representations.

Pre-trained Word Embedding. To deal with the problem of informal spellings and casual expressions, we should capture the information from tweets as accurate as possible. We use the GloVe[23] embeddings pre-trained on a large-scale Twitter corpus of two billion tweets. The vocabulary of pre-trained embeddings covers most common misspellings and aliases of most common words.

Pre-label. POI inventory, containing partial and familiar names from Foursquare, is proposed in [16] to pre-label candidate location mentions in tweets. It has been proved that pre-label is an essential resource to improve recognition performance. We use a CRF toolkit[1] to automatically assign the pre-labels with BILOU scheme[27].

Character-level Representation. Previous studies [4, 13] have shown that character-level information (*e.g.,* prefix and suffix of a word) is an effective resource for NER task. CNN and BiLSTM are commonly used in previous works to extract character-level representation. In our model, we use CNN because of its lower computational cost [4, 19].

BiLSTM-CRF. LSTM is a variant of recurrent neural network (RNN) and is designed to deal with vanishing gradients problem. BiLSTM uses two LSTMs to represent each token of the sequence based on both the past and the future context of the token. As shown in Figure 2, one LSTM processes the sequence from left to right, the other one from right to left. For a word, we concatenate its pre-trained word embedding, pre-label and character-level representation as its final representation, which is then fed into a BiLSTM layer. Then, the output sequence is fed into a CRF layer to infer the tag sequence. BiLSTM-CRF is a state-of-the-art approach to named entity recognition [6, 13].

3.2 Location Linking

Figure 3 shows the architecture of the location linking module. We use two pairs of parallel LSTMs (four in total) to encode the left-side and right-side contexts of a mention, respectively. Note that the input of the two parallel LSTMs for each side is re-weighted by two different *attentions* from the location profile. The output sequences of LSTMs are "fused" by a *fusion gate* and then fed into a max-pooling layer over all time steps as the final representation for each side context. Next, we concatenate the left-side context representation, right-side context representation, the representation of location profile together with the *Manhattan distance* between user coordinates (*i.e.,* user position attached to the tweet) and coordinates from location profile. Finally, a multilayer perceptron (MLP) with a sigmoid activated output layer is used to output a scalar ranged between zero and one as the matching score.

Location Profile Mapping Dictionary. Following [14], we use a location profile mapping dictionary to recall all possible candidates for a location mention. The *key* of the dictionary is a possible POI mention, and the *value* is the list of candidate profile indexes for the POI mention. If the mention is not in the mapping dictionary, we predict it as *un-linkable.* The dictionary is constructed with Foursquare check-in tweets.

Mention's Context. Following [25], we use LSTM networks to capture two-side contextual information of the POI mention. The difference is that we use all words in a tweet instead of specific window size. The left-side context starts from the leftmost word in a tweet and ends at the rightmost word inside the mention. Conversely, the right-side context starts from the tweet's rightmost word and ends at the leftmost word of the mention. Since the tweet does not have a long context, we use all words to understand the tweet as a whole. Note that the input of LSTM layers is re-weighted with a multi-attention mechanism, which is to be detailed shortly.

Behavioral and Semantic Information. Users often reveal their locations and describe what they are doing in tweets. For example, a user posts "Great lobster @ Red Robin!", which contains information about behavior (having lobster). Here, we name such information as ***behavioral information***. Obviously, since few restaurants serve lobster, it could be beneficial for POI disambiguation. According to our observation, tweets share the same behavioral information with the tips (*i.e.,* user comments) in the Foursquare location profile. On the other hand, some words in a tweet may be directly semantically relevant to the category or address of a location. For instance, in the tweet "Best wine at Elle's in the center", *wine* is closer to *bar* in word embedding space. Such information

[1] CRF++: https://github.com/taku910/crfpp

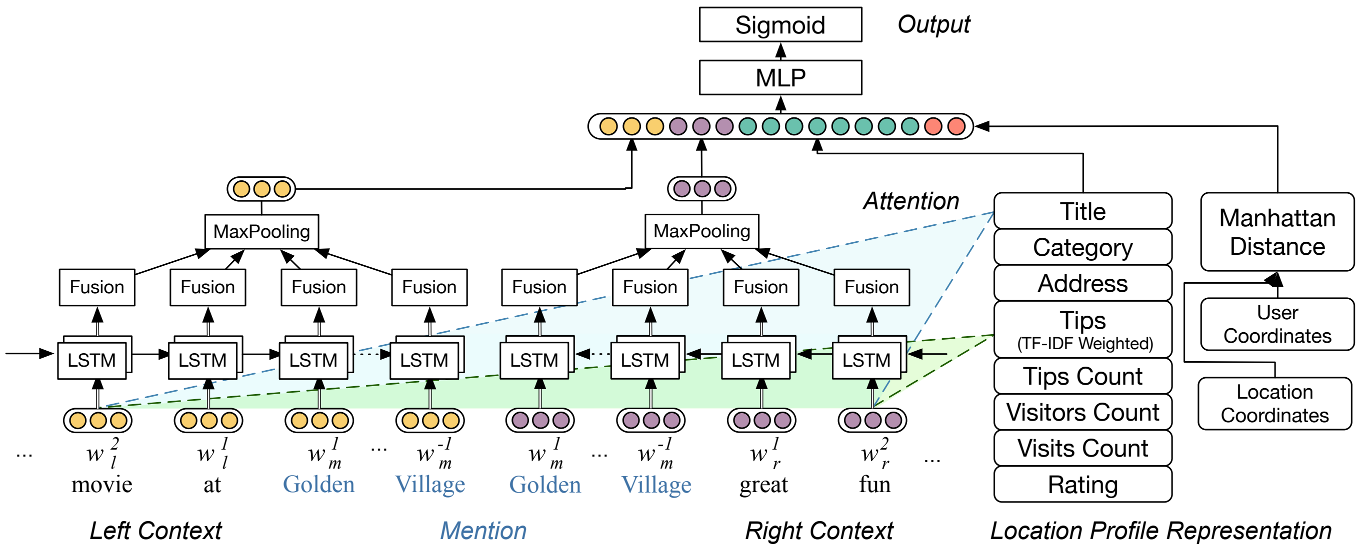

Figure 3: The architecture of location linking network. The input is a pair ⟨tweet, profile⟩, and the output is the matching score.

is beneficial for location disambiguation between *Elle's Bar* and *Elle's Salon*. Similarly, in the same example, *center* is a synonym for *Central Region*. This can be helpful for picking a location profile titled "Elle's Bar - Central Region" out of other branches of Elle's Bar. We name such information as ***semantic information***.

Multi-attention Mechanism. To comprehensively exploit *semantic information* and *behavioral information*, we develop a multi-attention mechanism to assign weights for the input sequence. As shown in Figure 3, we use the *title* representation and *tips* representation as two attention vectors. Specifically, given an input word embedding w_i in a tweet and an attention vector p (*i.e.,* representation of title or tips from location profile), the re-weighted input word embedding w_i' is defined by:

$$z_i = \tanh\left(V_p p + V_w w_i\right) \tag{1}$$

$$s_i = \mathrm{softmax}(v_k z_i + v_b) \tag{2}$$

$$w_i' = w_i s_i \tag{3}$$

where V_*, v_k, and v_b are attention parameters which can be learned during training. The TF-IDF weighted tips representation (to be detailed shortly) contains the most characteristic behavioral information about a location. Thus, the attention from tips can highlight the parts which are most relevant with the tips. Similarly, the title contains rich semantic information (including *location name*, *branch name* and *business status*) which can highlight the parts relevant to the location.

Fusion Gate. To collect important information from the output of two parallel LSTMs, we borrow a simple but practical *fusion gate* from [29]. Given the output of two parallel LSTMs with different attentions (*i.e.,* x_i^1 and x_i^2), the output of fusion gate u_i is formally defined by:

$$f_i = \mathrm{sigmoid}(W_1 x_i^1 + W_2 x_i^2 + b) \tag{4}$$

$$u_i = f_i \odot x_i^1 + (1 - f_i) \odot x_i^2 \tag{5}$$

where W_1, W_2, and b are learnable parameters. The fusion gate enables the model to learn how to weight the two input sequences, which solves the problem of the imbalanced importance of behavioral information and semantic information.

Location Profile. The properties (exclude "category") of a location profile fall into two classes: *word-based* and *numeric-based*. For *word-based* properties (*i.e.,* title, address, and tips), we represent each property by averaging all word embeddings inside it. In particular, the word embeddings of "tips" are weighted by TF-IDF schema. For *numeric-based* properties (*i.e.,* tips count, visitors count and visits count), we normalize them through the whole CLP. For *category* property, we encode it as a one-hot vector. Finally, we concatenate representations of all properties together as the location profile representation.

Manhattan Distance. To measure the distance between user coordinates and location coordinates in a city, we use the Manhattan distance (*i.e.,* taxicab distance) defined by:

$$d(p, q) = |p_x - q_x| + |p_y - q_y| \tag{6}$$

where p and q are two pairs of coordinates. Manhattan distance is better than Euclidean distance since the road network in a city or suburb is usually orthogonal. Note that not every tweet has an attached user position. For these situations, we fix the distance value to a pre-defined constant. Manhattan distance is also used in our post-processing, geographical pair linking, which will be discussed in Section 4.

Final Prediction. For all candidate location profiles in our mapping dictionary for a mention, we predict the matching score between the mention and each candidate. Formally, the final predicted location profile p_i for mention m_i is computed by:

$$p_i = \underset{p_j \in c(m_i)}{\arg\max}\, \phi(m_i, p_j) \tag{7}$$

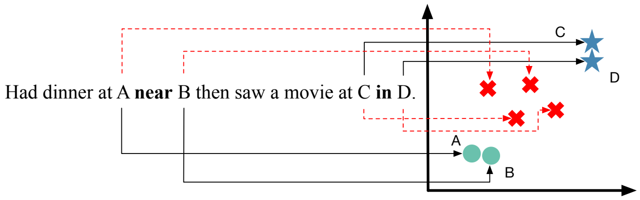

Figure 4: A wrong case of taking into account the coherence among all mentions. A, B, C and D are true POIs mentioned by a user. The incorrect predictions are marked with red dashed arrows and crosses.

where $c(m_i)$ is the candidate set for m_i and $\phi(m_i, p_j)$ is the matching score between m_i and p_j.

4 GEOGRAPHICAL PAIR LINKING

In this section, we introduce Geographical Pair Linking (Geo-PL), a post-processing strategy to enhance our linking performance.

For location linking, one of the most valuable information is the geographical coherence among mentioned locations in the same tweet. First, instead of directly calling the branch name of a chain restaurant, users are more likely to mention them with another location (usually a landmark). Second, people may describe a route by listing the locations along the way one by one, which also reveals the coherence among the mentioned locations.

For geographical coherence, prior studies often measure the distances among all mentions in a tweet. According to our observation, geographical coherence between two locations is strong enough for location disambiguation. Moreover, as illustrated in Figure 4, when a user mentions four locations with a conventional way like "Had dinner at A near B then saw a movie at C in D," the geographical coherence measurement among all mentions brings obvious error. It obtains the collective minimum but disregards the strong connection between two mentions.

To exploit pair-wise geographical coherence, we developed **Geographical Pair Linking (Geo-PL)** algorithm based on the *Pair Linking* algorithm [25]. The original algorithm measures the cosine similarity between text embeddings, while we use Manhattan distance discussed in Section 3.2 to measure the geographical coherence. Geo-PL iteratively resolves all pairs of mentions, starting from the most *confident* pair. The confidence score of a pair of links $m_i \mapsto p_i$ and $m_j \mapsto p_j$ is defined by:

$$conf(i,j) = (1-\beta)\frac{[\phi(m_i,p_i)+\phi(m_j,p_j)]}{2} + \beta\frac{1}{d(p_i,p_j)} \quad (8)$$

where β is a given coefficient representing the preference between the matching scores and the geographical coherence; d is the Manhattan distance defined by Equation 6. In the case of $p_i = p_j$, we temporarily set β to 0 when calculating. The procedure of Geo-PL (the same as Pair Linking) is detailed in Algorithm 1. Note that if there is only one POI mention in a tweet, we simply predict its location profile by Equation 7.

Algorithm 1: Geo-PL Algorithm

Input: N mentions($m_1, \ldots, m_N$). Mention m_i has candidate profiles $\{e|e_i \in C(m_i)\}$
Output: $\Gamma = (p_1, \ldots, p_N)$
$p_i \leftarrow null, \forall p_i \in \Gamma$;
foreach $pair(m_i, m_j) \wedge m_i \neq m_j$ **do**
  $Q_{m_i,m_j} \leftarrow top\_pair(m_i, C(m_i), m_j, C(m_j))$;
  $Q.add(Q_{m_i,m_j})$;
end
while $\exists p_i \in \Gamma, p_i = null$ **do**
  $(m_i, e_i, m_j, e_j) \leftarrow most\_confident\_pair(Q)$;
  $p_i \leftarrow e_i$;
  $p_j \leftarrow e_j$;
  for $k := 1 \to N \wedge p_k = null$ **do**
    $Q_{m_k,m_i} \leftarrow top\_pair(m_k, C(m_k), m_i, \{p_i\})$;
    $Q_{m_k,m_j} \leftarrow top\_pair(m_k, C(m_k), m_j, \{p_j\})$;
  end
end

5 EXPERIMENT

5.1 Data Preparation

We use a Singaporean national Twitter dataset released by Ji *et al.* [14]. The dataset includes a CLP and a set of labeled tweets.

326,853 Foursquare *check-in tweets*, containing both formal auto-generated POI name and informal user mentions, are collected to build the CLP (including 22,414 valid location profiles) and the POI inventory (including 27,386 entries). Location profile mapping dictionary (see Section 3.2) is also constructed with check-in tweets. Informal mentions in check-in tweets and the indexes of crawled profiles are used as keys and values, respectively. The final location profile mapping dictionary has 24,750 keys and 63,091 <*key, value*> mappings.

The dataset consists of 3,611 labeled tweets and 1,542 POI locations, 543 of which can be linked to a profile in the CLP. 10% of tweets have an attached user position. These tweets are labeled by human annotators. All possible POI mentions are labeled with one of *NPOI* (*i.e.,* not a POI mention), *index* of linked location profile, or *NIL* (*i.e.,* cannot be linked to a profile).

We split the 3,611 labeled tweets randomly into three subsets: 2,500 tweets for training, 211 tweets for validation, and 900 tweets for testing. Same POI inventory and location profile mapping dictionary used in [14] are employed through all models in our experiments.

5.2 Parameter Setting and Evaluation Metrics

We set β, the coefficient of Geo-PL preference to 0.8 based on fine-tuning on the validation set. When concatenating the scalars to be the input of MLP, we duplicate them to the same number of dimensions as the word-based properties (*i.e.,* 200 dimensions in our experiments). This setting prevents the scalars from being "ignored" during training.

We conduct experiments on location recognition subtask, location linking subtask and the whole task, respectively. We adopt three metrics, Precision (*Pr*), Recall (*Re*) and F_1 which are widely

Table 1: Performance on Location Recognition and Location Linking.

Methods	Location Recognition			Location Linking			Recognition + Linking		
	Pr	*Re*	F_1	*Pr*	*Re*	F_1	*Pr*	*Re*	F_1
Li *et al.* [16]	**0.8962**	0.7661	0.8261	-	-	-	-	-	-
Shen *et al.* [30]	-	-	-	0.6723	0.6349	0.6531	-	-	-
Pipelined [16] [30]	-	-	-	-	-	-	0.6339	0.5635	0.5966
JoRL_L [14]	0.8926	0.7823	**0.8338**	-	-	-	0.8152	0.5952	0.6881
DLocRL	0.8575	**0.8091**	0.8326	**0.8235**	**0.7778**	**0.8000**	**0.8350**	**0.6825**	**0.7511**

used to evaluate NER and EL tasks. In particular, we use the ground-truth mentions instead of the predicted output of the recognition module, to get the metric scores for location linking.

5.3 Overall Comparison

We compare our model with state-of-the-art solutions. More specifically, we compare our model with [16] and [14] for recognition, and [30] for linking, respectively. Note that the performance of [14] on linking subtask is not provided since it is a joint model. For the whole task, we choose JoRL_L as the baseline since it is the state-of-the-art joint model in supervised learning manner. All models are fine-tuned on the validation set.

The result is shown in Table 1. We observe that DLocRL achieves the highest *recall* on location recognition. DLocRL beats Li *et al.* [16] and JoRL_L [14] with relative recall improvements of 5.61% and 3.43%, respectively. We contribute this to the fact that word/character embeddings enable our model to be more "tolerant" than prior works which depend on hand-crafted features. Although the precision score of DLocRL is not good, it does not have an impact on the whole task, because the location profile mapping dictionary can filter out most false positive (FP) prediction in the linking module.

On location linking subtask, our method outperforms the work [30] by 22.49% on precision, 22.51% on recall and 22.49% on F_1.

On the whole task, our model prominently outperforms the state-of-the-art joint solution (JoRL_L) on all three metrics (2.43% on precision, 14.67% on recall, 9.16% on F_1). Also, our model dramatically outperforms the prior state-of-the-art *pipeline* by 31.72% on precision, 21.12% on recall and 25.90% on F_1.

5.4 Effect of Multi-attention for Linking

We conduct experiments to verify the effectiveness of the multi-attention mechanism. Table 2 shows the linking performance with single attention from tips/title attentions and multi-attention. Note that the single attention approaches can only slightly improve the performance because they have "prejudice" which considers either behavioral information or semantic information. By "fusing" this two information, DLocRL eliminates the prejudice so it can effectively filter out noisy information. With the multi-attention mechanism, DLocRL improves the performance of linking module by 4.25% on precision, 4.26% on recall, and 4.26% on F_1, compared with the baseline.

5.5 Effect of Geo-PL for Linking

As discussed in Section 4, we introduce a novel post-processing strategy, named Geo-PL. Here, we conduct experiments to show

Table 2: Effect of Multi-attention Mechanism for Linking.

Attention	Location Linking		
	Pr	*Re*	F_1
Baseline (*without attention*)	0.7899	0.7460	0.7673
Single Tips Attention	0.7983	0.7540	0.7755
Single Title Attention	0.7899	0.7460	0.7673
Tips + Title (*multi-attention*)	0.8235	0.7778	0.8000

Table 3: Effect of Geo-PL for Linking.

Strategy	Location Linking		
	Pr	*Re*	F_1
Without Geo-PL	0.7258	0.7143	0.7200
With Geo-PL	0.8235	0.7778	0.8000

the impact of Geo-PL for location linking. Table 3 shows the experimental results. Compared with no Geo-PL component, DLocRL (with Geo-PL) significantly improves precision by 13.46%, recall by 8.89% and F_1 by 11.11%.

6 CONCLUSIONS

In this paper, we introduce DLocRL, the first deep neural network based pipeline to recognize fine-grained location mentions in tweets and link the recognized locations to location profiles. Moreover, we develop a novel post-processing strategy which can further improve location linking performance. Through extensive experiments, we demonstrate the effectiveness of DLocRL against state-of-the-art solutions on a real-world Twitter dataset. The ablation experiments show the effectiveness of multi-attention mechanism and Geo-PL strategy on location linking.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their careful reading and their many insightful comments and suggestions. This research was supported by National Natural Science Foundation of China (No.U1636219, No.U1804263, No.61872278, No.61502344), Natural Scientific Research Program of Hubei Province (No.2017CFB-502, No.2017CFA007), the National Key R&D Program of China (No.2016QY01W0105, No.2016YFB0801303), and Plan for Scientific Innovation Talent of Henan Province (No.2018JR0018). Xiangyang Luo is the corresponding author.

REFERENCES

[1] Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual String Embeddings for Sequence Labeling. In *Proc. COLING*. 1638–1649.

[2] Krisztian Balog. 2018. Entity-Oriented Search. (2018).

[3] Peter F. Brown, Vincent J. Della Pietra, Peter V. de Souza, Jennifer C. Lai, and Robert L. Mercer. 1992. Class-Based n-gram Models of Natural Language. *Computational Linguistics* 18, 4 (1992), 467–479.

[4] Jason PC Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional LSTM-CNNs. In *TACL*, Vol. 4. 357–370.

[5] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12, Aug (2011), 2493–2537.

[6] Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. Transition-Based Dependency Parsing with Stack Long Short-Term Memory. In *Proc ACL*. 334–343.

[7] Fredrik Espinoza, Per Persson, Anna Sandin, Hanna Nyström, Elenor Cacciatore, and Markus Bylund. 2001. Geonotes: Social and navigational aspects of location-based information systems. In *Proc. UBICOMP*. 2–17.

[8] Yoav Goldberg. 2017. Neural network methods for natural language processing. *Synthesis Lectures on Human Language Technologies* 10, 1 (2017), 1–309.

[9] Stephen Guo, Ming-Wei Chang, and Emre Kiciman. 2013. To Link or Not to Link? A Study on End-to-End Tweet Entity Linking. In *Proc. NAACL-HLT*. 1020–1030.

[10] Jialong Han, Aixin Sun, Gao Cong, Wayne Xin Zhao, Zongcheng Ji, and Minh C Phan. 2018. Linking Fine-Grained Locations in User Comments. *IEEE Transactions on Knowledge and Data Engineering* 30, 1 (2018), 59–72.

[11] Xianpei Han, Le Sun, and Jun Zhao. 2011. Collective entity linking in web text: a graph-based method. In *Proc. SIGIR*. 765–774.

[12] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust Disambiguation of Named Entities in Text. In *Proc. EMNLP*. 782–792.

[13] Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991* (2015).

[14] Zongcheng Ji, Aixin Sun, Gao Cong, and Jialong Han. 2016. Joint Recognition and Linking of Fine-Grained Locations from Tweets. In *Proc. WWW*. 1271–1281.

[15] Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti. 2009. Collective annotation of Wikipedia entities in web text. In *Proc. SIGKDD*. 457–466.

[16] Chenliang Li and Aixin Sun. 2014. Fine-grained location extraction from tweets with temporal awareness. In *Proc. SIGIR*. 43–52.

[17] Daifeng Li, Zhipeng Luo, Ying Ding, Jie Tang, Gordon Guo-Zheng Sun, Xiaowen Dai, John Du, Jingwei Zhang, and Shoubin Kong. 2017. User-level microblogging recommendation incorporating social influence. *Journal of the Association for Information Science and Technology* 68, 3 (2017), 553–568.

[18] Guoliang Li, Jun Hu, Jianhua Feng, and Kian-Lee Tan. 2014. Effective location identification from microblogs. In *Proc. ICDE*. 880–891.

[19] Jing Li, Aixin Sun, and Shafiq Joty. 2018. SegBot: A Generic Neural Text Segmentation Model with Pointer Network. In *Proc. IJCAI*. 4166–4172.

[20] John Lingad, Sarvnaz Karimi, and Jie Yin. 2013. Location extraction from disaster-related microblogs. In *Proc. WWW*. 1017–1020.

[21] Shervin Malmasi and Mark Dras. 2015. Location Mention Detection in Tweets and Microblogs. In *Proc. PACLING*. 123–134.

[22] Lih-Bin Oh and Heng Xu. 2003. Effects of multimedia on mobile consumer behavior: An empirical study of location-aware advertising. *Proc. ICiS* (2003), 56.

[23] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In *Proc. EMNLP*. 1532–1543.

[24] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. NAACL-HLT*. 2227–2237.

[25] Minh C. Phan, Aixin Sun, Yi Tay, Jialong Han, and Chenliang Li. 2017. NeuPL: Attention-based Semantic Matching and Pair-Linking for Entity Disambiguation. In *Proc. CIKM*. 1667–1676.

[26] Adam Rae, Vanessa Murdock, Adrian Popescu, and Hugues Bouchard. 2012. Mining the web for points of interest. In *Proc. SIGIR*. 711–720.

[27] Lev-Arie Ratinov and Dan Roth. 2009. Design Challenges and Misconceptions in Named Entity Recognition. In *Proc. CoNLL*. 147–155.

[28] Jochen Schiller and Agnès Voisard. 2004. *Location-based services.* Elsevier.

[29] Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. 2018. DiSAN: Directional Self-Attention Network for RNN/CNN-Free Language Understanding. In *Proc. AAAI*.

[30] Wei Shen, Jianyong Wang, Ping Luo, and Min Wang. 2013. Linking named entities in Tweets with knowledge base via user interest modeling. In *Proc. SIGKDD*. 68–76.

[31] Yaming Sun, Lei Lin, Duyu Tang, Nan Yang, Zhenzhou Ji, and Xiaolong Wang. 2015. Modeling Mention, Context and Entity with Neural Networks for Entity Disambiguation. In *Proc. IJCAI*. 1333–1339.

[32] Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. 2005. Large Margin Methods for Structured and Interdependent Output Variables. *Journal of Machine Learning Research* 6 (2005), 1453–1484.

[33] Zhilin Yang, Ruslan Salakhutdinov, and William Cohen. 2016. Multi-task cross-lingual sequence tagging from scratch. *arXiv preprint arXiv:1603.06270* (2016).

[34] Wei Zhang and Judith Gelernter. 2014. Geocoding location expressions in Twitter messages: A preference learning method. *J. Spatial Information Science* 9, 1 (2014), 37–70.

[35] Yue Zhang and Stephen Clark. 2008. Joint Word Segmentation and POS Tagging Using a Single Perceptron. In *Proc. ACL*. 888–896.

[36] Xin Zheng, Jialong Han, and Aixin Sun. 2018. A Survey of Location Prediction on Twitter. *IEEE Trans. Knowl. Data Eng.* 30, 9 (2018), 1652–1671.