Visual Attention Model for Name Tagging in Multimodal Social Media

Di Lu\textsuperscript{1}, Leonardo Neves\textsuperscript{2}, Vitor Carvalho\textsuperscript{3}, Ning Zhang\textsuperscript{2}, Heng Ji\textsuperscript{1}

\textsuperscript{1}Computer Science, Rensselaer Polytechnic Institute
\{lud2,jih\}@rpi.edu

\textsuperscript{2}Snap Research
\{lneves, ning.zhang\}@snap.com

\textsuperscript{3}Intuit
vitor.carvalho@intuit.com

Abstract

Everyday billions of multimodal posts containing both images and text are shared in social media sites such as Snapchat, Twitter or Instagram. This combination of image and text in a single message allows for more creative and expressive forms of communication, and has become increasingly common in such sites. This new paradigm brings new challenges for natural language understanding, as the textual component tends to be shorter, more informal, and often is only understood if combined with the visual context. In this paper, we explore the task of name tagging in multimodal social media posts. We start by creating two new multimodal datasets: one based on Twitter posts\textsuperscript{1} and the other based on Snapchat captions (exclusively submitted to public and crowdsourced stories). We then propose a novel model based on Visual Attention that not only provides deeper visual understanding on the decisions of the model, but also significantly outperforms other state-of-the-art baseline methods for this task. \textsuperscript{2}

1 Introduction

Social platforms, like Snapchat, Twitter, Instagram and Pinterest, have become part of our lives and play an important role in making communication easier and accessible. Once text-centric, social media platforms are becoming increasingly multimodal, with users combining images, videos, audios, and texts for better expressiveness. As social media posts become more multimodal, the natural language understanding of the textual components of these messages becomes increasingly challenging. In fact, it is often the case that the textual component can only be understood in combination with the visual context of the message.

In this context, here we study the task of Name Tagging for social media containing both image and textual contents. Name tagging is a key task for language understanding, and provides input to several other tasks such as Question Answering, Summarization, Searching and Recommendation. Despite its importance, most of the research in name tagging has focused on news articles and longer text documents, and not as much in multimodal social media data (Baldwin et al., 2015).

However, multimodality is not the only challenge to perform name tagging on such data. The textual components of these messages are often very short, which limits context around names. Moreover, there linguistic variations, slangs, typos and colloquial language are extremely common, such as using ‘loooove’ for ‘love’, ‘LosAngeles’ for ‘Los Angeles’, and ‘#Chicago #Bull’ for ‘Chicago Bulls’. These characteristics of social media data clearly illustrate the higher difficulty of this task, if compared to traditional newswire name tagging.

In this work, we modify and extend the current state-of-the-art model (Lample et al., 2016; Ma and Hovy, 2016) in name tagging to incorporate the visual information of social media posts using an Attention mechanism. Although the usually short textual components of social media posts provide limited contextual information, the accompanying images often provide rich information that can be useful for name tagging. For ex-

\textsuperscript{1}This work was mostly done during the first author’s internship at Snap Research.

\textsuperscript{2}The Twitter data and associated images presented in this paper were downloaded from https://archive.org/details/twitterstream

\textsuperscript{2}We will make the annotations on Twitter data available for research purpose upon request.
example, as shown in Figure 1, both captions include the phrase ‘Modern Baseball’. It is not easy to tell if each Modern Baseball refers to a name or not from the textual evidence only. However using the associated images as reference, we can easily infer that Modern Baseball in the first sentence should be the name of a band because of the implicit features from the objects like instruments and stage, and the Modern Baseball in the second sentence refers to the sport of baseball because of the pitcher in the image.

In this paper, given an image-sentence pair as input, we explore a new approach to leverage visual context for name tagging in text. First, we propose an attention-based model to extract visual features from the regions in the image that are most related to the text. It can ignore irrelevant visual information. Secondly, we propose to use a gate to combine textual features extracted by a Bidirectional Long Short Term Memory (BLSTM) and extracted visual features, before feed them into a Conditional Random Fields (CRF) layer for tag predication. The proposed gate architecture plays the role to modulate word-level multimodal features.

We evaluate our model on two labeled datasets collected from Snapchat and Twitter respectively. Our experimental results show that the proposed model outperforms state-for-the-art name tagger in multimodal social media.

The main contributions of this work are as follows:

- We create two new datasets for name tagging in multimedia data, one using Twitter and the other using crowd-sourced Snapchat posts. These new datasets effectively constitute new benchmarks for the task.
- We propose a visual attention model specifically for name tagging in multimodal social media data. The proposed end-to-end model only uses image-sentence pairs as input without any human designed features, and a Visual Attention component that helps understand the decision making of the model.

2 Model

Figure 2 shows the overall architecture of our model. We describe three main components of our model in this section: BLSTM-CRF sequence labeling model (Section 2.1), Visual Attention Model (Section 2.3) and Modulation Gate (Section 2.4).

Given a pair of sentence and image as input, the Visual Attention Model extracts regional visual features from the image and computes the weighted sum of the regional visual features as the visual context vector, based on their relatedness with the sentence. The BLSTM-CRF sequence labeling model predicts the label for each word in the sentence based on both the visual context vector and the textual information of the words. The modulation gate controls the combination of the visual context vector and the word representations for each word before the CRF layer.

2.1 BLSTM-CRF Sequence Labeling

We model name tagging as a sequence labeling problem. Given a sequence of words: \( S = \{s_1, s_2, ..., s_n\} \), we aim to predict a sequence of labels: \( L = \{l_1, l_2, ..., l_n\} \), where \( l_i \in \mathcal{L} \) and \( \mathcal{L} \) is a pre-defined label set.

**Bidirectional LSTM.** Long Short-term Memory Networks (LSTMs) (Hochreiter and Schmidhuber, 1997) are variants of Recurrent Neural Networks (RNNs) designed to capture long-range dependencies of input. The equations of a LSTM cell are as follows:

\[
\begin{align*}
i_t & = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \\
f_t & = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \\
\tilde{c}_t & = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\
c_t & = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \\
o_t & = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \\
h_t & = o_t \odot \tanh(c_t)
\end{align*}
\]

where \( x_t, c_t \) and \( h_t \) are the input, memory and hidden state at time \( t \) respectively. \( W_{xi}, W_{hi}, W_{xf}, W_{hf}, W_{xc}, W_{hc}, W_{xo}, \) and \( W_{ho} \) are weight matrices. \( \odot \) is the element-wise product function and \( \sigma \) is the element-wise sigmoid function.
Name Tagging benefits from both of the past (left) and the future (right) contexts, thus we implement the Bidirectional LSTM (Graves et al., 2013; Dyer et al., 2015) by concatenating the left and right context representations, $h_t = [h_t^L, h_t^R]$, for each word.

**Character-level Representation.** Following (Lample et al., 2016), we generate the character-level representation for each word using another BLSTM. It receives character embeddings as input and generates representations combining implicit prefix, suffix and spelling information. The final word representation $x_i$ is the concatenation of word embedding $e_i$ and character-level representation $c_i$.

$$
c_i = BLSTM_{char}(s_i) \quad s_i \in S
$$

$$
x_i = [e_i, c_i]
$$

**Conditional random fields (CRFs).** For name tagging, it is important to consider the constraints of the labels in neighborhood (e.g., I-LOC must follow B-LOC). CRFs (Lafferty et al., 2001) are effective to learn those constraints and jointly predict the best chain of labels. We follow the implementation of CRFs in (Ma and Hovy, 2016).

### 2.2 Visual Feature Representation

We use Convolutional Neural Networks (CNNs) (LeCun et al., 1989) to obtain the representations of images. Particularly, we use Residual Net (ResNet) (He et al., 2016), which achieves state-of-the-art on ImageNet (Russakovsky et al., 2015) detection, ImageNet localization, COCO (Lin et al., 2014) detection, and COCO segmentation tasks. Given an input pair $(S, I)$, where $S$ represents the word sequence and $I$ represents the image rescaled to 224x224 pixels, we use ResNet to extract visual features for regional areas as well as for the whole image (Fig 3):

$$
V_g = ResNet_g(I)
$$

$$
V_r = ResNet_r(I)
$$

where the global visual vector $V_g$, which represents the whole image, is the output before the last fully connected layer\(^3\). The dimension of $V_g$ is 1,024. $V_r$ are the visual representations for regional areas and they are extracted from the last convolutional layer of ResNet, and the dimension is 1,024x7x7 as shown in Figure 3. 7x7 is the number of regions in the image and 1,024 is the

\[^3\]the last fully connect layer outputs the probabilities over 1,000 classes of objects.
dimension of the feature vector. Thus each feature vector of $V_r$ corresponds to a 32x32 pixel region of the rescaled input image.

### 2.3 Visual Attention Model

![Figure 4: Example of partially related image and sentence. (‘I have just bought Jeremy Pied.’)](image)

The global visual representation is a reasonable representation of the whole input image, but not the best. Sometimes only parts of the image are related to the associated sentence. For example, the visual features from the right part of the image in Figure 4 cannot contribute to inferring the information in the associated sentence ‘I have just bought Jeremy Pied.’ In this work we utilize visual attention mechanism to combat the problem, which has been proven effective for vision-language related tasks such as Image Captioning (Xu et al., 2015) and Visual Question Answering (Yang et al., 2016b; Lu et al., 2016), by enforcing the model to focus on the regions in images that are mostly related to context textual information while ignoring irrelevant regions. Also the visualization of attention can also help us to understand the decision making of the model. Attention mechanism is mapping a query and a set of key-value pairs to an output. The output is a weighted sum of the values and the assigned weight for each value is computed by a function of the query and corresponding key. We encode the sentence into a query vector using an LSTM, and use regional visual representations $V_r$ as both keys and values.

**Text Query Vector.** We use an LSTM to encode the sentence into a query vector, in which the inputs of the LSTM are the concatenations of word embeddings and character-level word representations. Different from the LSTM model used for sequence labeling in Section 2.1, the LSTM here aims to get the semantic information of the sentence and it is unidirectional:

$$Q = LSTM_{query}(S) \quad (1)$$

**Attention Implementation.** There are many implementations of visual attention mechanism such as Multi-layer Perceptron (Bahdanau et al., 2014), Bilinear (Luong et al., 2015), dot product (Luong et al., 2015), Scaled Dot Product (Vaswani et al., 2017), and linear projection after summation (Yang et al., 2016b). Based on our experimental results, dot product implementations usually result in more concentrated attentions and linear projection after summation results in more dispersed attentions. In the context of name tagging, we choose the implementation of linear projection after summation because it is beneficial for the model to utilize as many related visual features as possible, and concentrated attentions may make the model bias. For implementation, we first project the text query vector $Q$ and regional visual features $V_r$ into the same dimensions:

$$P_t = \tanh(W_tQ)$$
$$P_v = \tanh(W_vV_r)$$

then we sum up the projected query vector with each projected regional visual vector respectively:

$$A = P_t \oplus P_v$$

the weights of the regional visual vectors:

$$E = \text{softmax}(W_aA + b_a)$$

where $W_a$ is weights matrix. The weighted sum of the regional visual features is:

$$v_c = \sum \alpha_i v_i \quad \alpha_i \in E, v_i \in V_r$$

We use $v_c$ as the visual context vector to initialize the BLSTM sequence labeling model in Section 2.1. We compare the performances of the models using global visual vector $V_g$ and attention based visual context vector $V_c$ for initialization in Section 4.

### 2.4 Visual Modulation Gate

The BLSTM-CRF sequence labeling model benefits from using the visual context vector to initialize the LSTM cell. However, the better way to utilize visual features for sequence labeling is to incorporate the features at word level individually. However visual features contribute quite
differently when they are used to infer the tags of different words. For example, we can easily find matched visual patterns from associated images for verbs such as ‘*sing*’, ‘*run*’, and ‘*play*’. Words/Phrases such as names of basketball players, artists, and buildings are often well-aligned with objects in images. However it is difficult to align function words such as ‘*the*’, ‘*of*’ and ‘*well*’ with visual features. Fortunately, most of the challenging cases in name tagging involve nouns and verbs, the disambiguation of which can benefit more from visual features.

We propose to use a visual modulation gate, similar to (Miyamoto and Cho, 2016; Yang et al., 2016a), to dynamically control the combination of visual features and word representation generated by BLSTM at word-level, before feed them into the CRF layer for tag prediction. The equations for the implementation of modulation gate are as follows:

\[
\begin{align*}
\beta_v &= \sigma(W_v h_i + U_v v_c + b_v) \\
\beta_w &= \sigma(W_w h_i + U_w v_c + b_w) \\
m &= \tanh(W_m h_i + U_m v_c + b_m) \\
w_m &= \beta_w \cdot h_i + \beta_v \cdot m
\end{align*}
\]

where \(h_i\) is the word representation generated by BLSTM, \(v_c\) is the computed visual context vector, \(W_v, W_w, W_m, U_v, U_w\) and \(U_m\) are weight matrices, \(\sigma\) is the element-wise sigmoid function, and \(w_m\) is the modulated word representations fed into the CRF layer in Section 2.1. We conduct experiments to evaluate the impact of modulation gate in Section 4.

3 Datasets

We evaluate our model on two multimodal datasets, which are collected from Twitter and Snapchat respectively. Table 1 summarizes the data statistics. Both datasets contain four types of named entities: Location, Person, Organization and Miscellaneous. Each data instance contains a pair of sentence and image, and the names in sentences are manually tagged by three expert labelers.

Twitter name tagging. The Twitter name tagging dataset contains pairs of tweets and their associated images extracted from May 2016, January 2017 and June 2017. We use sports and social event related key words, such as *concert*, *festival*, *soccer*, *basketball*, as queries. We don’t take into consideration messages without images for this experiment. If a tweet has more than one image associated to it, we randomly select one of the images.

Snap name tagging. The Snap name tagging dataset consists of caption and image pairs exclusively extracted from snaps submitted to public and live stories. They were collected between May and July of 2017. The data contains captions submitted to multiple community curated stories like the Electric Daisy Carnival (EDC) music festival and the Golden State Warrior’s NBA parade.

Both Twitter and Snapchat are social media with plenty of multimodal posts, but they have obvious differences with sentence length and image styles. In Twitter, text plays a more important role, and the sentences in the Twitter dataset are much longer than those in the Snap dataset (16.0 tokens vs 8.1 tokens). The image is often more related to the content of the text and added with the purpose of illustrating or giving more context. On the other hand, as users of Snapchat use cameras to communicate, the roles of text and image are switched. Captions are often added to complement what is being portrayed by the snap. On our experiment section we will show that our proposed model outperforms baseline on both datasets.

We believe the Twitter dataset can be an important step towards more research in multimodal name tagging and we plan to provide it as a benchmark upon request.

4 Experiment

4.1 Training

Tokenization. To tokenize the sentences, we use the same rules as (Owoputi et al., 2013), except we separate the hashtag ‘#’ with the words after.

Labeling Schema. We use the standard BIO schema (Sang and Veenstra, 1999), because we see little difference when we switch to BIOES schema (Ratinov and Roth, 2009).

Word embeddings. We use the 100-dimensional GloVe\(^4\) (Pennington et al., 2014) embeddings trained on 2 billions tweets to initialize the lookup table and do fine-tuning during training.

Character embeddings. As in (Lample et al., 2016), we randomly initialize the character embeddings with uniform samples. Based on experimental results, the size of the character embeddings affects little, and we set it as 50.

\(^4\)https://nlp.stanford.edu/projects/glove/
Table 1: Sizes of the datasets in numbers of sentence and token.

<table>
<thead>
<tr>
<th></th>
<th>Training</th>
<th>Development</th>
<th>Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Snapchat</td>
<td>Sentences</td>
<td>4,817</td>
<td>1,032</td>
</tr>
<tr>
<td></td>
<td>Tokens</td>
<td>39,035</td>
<td>8,334</td>
</tr>
<tr>
<td>Twitter</td>
<td>Sentences</td>
<td>4,290</td>
<td>1,432</td>
</tr>
<tr>
<td></td>
<td>Tokens</td>
<td>68,655</td>
<td>22,872</td>
</tr>
</tbody>
</table>

Pretrained CNNs. We use the pretrained ResNet-152 (He et al., 2016) from Pytorch.

Early Stopping. We use early stopping (Caruana et al., 2001; Graves et al., 2013) with a patience of 15 to prevent the model from over-fitting.

Fine Tuning. The models are optimized with fine-tuning on both the word-embeddings and the pre-trained ResNet.

Optimization. The models achieve the best performance by using mini-batch stochastic gradient descent (SGD) with batch size 20 and momentum 0.9 on both datasets. We set an initial learning rate of \( \eta_0 = 0.03 \) with decay rate of \( \rho = 0.01 \). We use a gradient clipping of 5.0 to reduce the effects of gradient exploding.

Hyper-parameters. We summarize the hyper-parameters in Table 2.

<table>
<thead>
<tr>
<th>Hyper-parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM hidden state size</td>
<td>300</td>
</tr>
<tr>
<td>Char LSTM hidden state size</td>
<td>50</td>
</tr>
<tr>
<td>visual vector size</td>
<td>100</td>
</tr>
<tr>
<td>dropout rate</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Table 2: Hyper-parameters of the networks.

4.2 Results

Table 3 shows the performance of the baseline, which is BLSTM-CRF with sentences as input only, and our proposed models on both datasets.

**BLSTM-CRF + Global Image Vector:** use global image vector to initialize the BLSTM-CRF.

**BLSTM-CRF + Visual attention:** use attention based visual context vector to initialize the BLSTM-CRF.

**BLSTM-CRF + Visual attention + Gate:** modulate word representations with visual vector.

Our final model **BLSTM-CRF + Visual Attention + Gate**, which has visual attention component and modulation gate, obtains the best F1 scores on both datasets. Visual features successfully play a role of validating entity types. For example, when there is a person in the image, it is more likely to include a person name in the associated sentence, but when there is a soccer field in the image, it is more likely to include a sports team name.

All the models get better scores on Twitter dataset than on Snap dataset, because the average length of the sentences in Snap dataset (8.1 tokens) is much smaller than that of Twitter dataset (16.0 tokens), which means there is much less contextual information in Snap dataset.

Also comparing the gains from visual features on different datasets, we find that the model benefits more from visual features on Twitter dataset, considering the much higher baseline scores on Twitter dataset. Based on our observation, users of Snapchat often post selfies with captions, which means some of the images are not strongly related to their associated captions. In contrast, users of Twitter prefer to post images to illustrate texts.

4.3 Attention Visualization

Figure 5 shows some good examples of the attention visualization and their corresponding name tagging results. The model can successfully focus on appropriate regions when the images are well aligned with the associated sentences. Based on our observation, the multimodal contexts in posts related to sports, concerts or festival are usually better aligned with each other, therefore the visual features easily contribute to these cases. For example, the ball and shoot action in example (a) in Figure 5 indicates that the context should be related to basketball, thus the ‘Warriors’ should be the name of a sports team. A singing person with a microphone in example (b) indicates that the name of an artist or a band (‘Radiohead’) may appear in the sentence.

The second and the third rows in Figure 5 show some more challenging cases whose tagging results benefit from visual features. In example (d), the model pays attention to the big Apple logo, thus tags the ‘Apple’ in the sentence as an Organization name. In example (e) and (i), a small
<table>
<thead>
<tr>
<th>Model</th>
<th>Snap Captions</th>
<th>Twitter</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
</tr>
<tr>
<td>BLSTM-CRF</td>
<td>57.71</td>
<td>58.65</td>
</tr>
<tr>
<td>BLSTM-CRF + Global Image Vector</td>
<td>61.49</td>
<td>57.84</td>
</tr>
<tr>
<td>BLSTM-CRF + Visual attention</td>
<td>65.53</td>
<td>57.03</td>
</tr>
<tr>
<td>BLSTM-CRF + Visual attention + Gate</td>
<td><strong>66.67</strong></td>
<td>57.84</td>
</tr>
</tbody>
</table>

Table 3: Results of our models on noisy social media data.

A group of people indicates that it is likely to include names of bands (‘Florence and the Machine’ and ‘BTS’). And a crowd can indicate an organization (‘Warriorette’ in example (i)). A jersey shirt on the table indicates a sports team. (‘Leicester’ in example (h) can refer to both a city and a soccer club based in it.)

### 4.4 Error Analysis

Figure 6 shows some failed examples that are categorized into three types: (1) bad alignments between visual and textual information; (2) blur images; (3) wrong attention made by the model.

Name tagging greatly benefits from visual features when the sentences are well aligned with the associated image as we show in Section 4.3. But it is not always the case in social media. The example (a) in Figure 6 shows a failed example resulted from poor alignment between sentences and images. In this image, there are two bins standing in front of a wall, but the sentence talks about basketball players. The unrelated visual information makes the model tag ‘Cleveland’ as a Location, however it refers to the basketball team ‘Cleveland Cavaliers’.

The image in example (b) is blur, so the extracted visual information extracted actually introduces noise instead of additional information. The
image in example (c) is about a baseball pitcher, but our model pays attention to the top right corner of the image. The visual context feature computed by our model is not related to the sentence, and results in missed tagging of ‘SBU’, which is an organization name.

5 Related Work

In this section, we summarize relevant background on previous work on name tagging and visual attention.

Name Tagging. In recent years, (Chiu and Nichols, 2015; Lample et al., 2016; Ma and Hovy, 2016) proposed several neural network architectures for named tagging that outperform traditional explicit features based methods (Chieu and Ng, 2002; Florian et al., 2003; Ando and Zhang, 2005; Ratinov and Roth, 2009; Lin and Wu, 2009; Passos et al., 2014; Luo et al., 2015). They all use Bidirectional LSTM (BLSTM) to extract features from a sequence of words. For character-level representations, (Lample et al., 2016) proposed to use another BLSTM to capture prefix and suffix information of words, and (Chiu and Nichols, 2015; Ma and Hovy, 2016) used CNN to extract position-independent character features. On top of BLSTM, (Chiu and Nichols, 2015) used a softmax layer to predict the label for each word, and (Lample et al., 2016; Ma and Hovy, 2016) used a CRF layer for joint prediction. Compared with traditional approaches, neural networks based approaches do not require hand-crafted features and achieved state-of-the-art performance on name tagging (Ma and Hovy, 2016). However, these methods were mainly developed for newswire and paid little attention to social media.

Name Tagging for more fine-grained types (e.g.

Visual Attention. Since the attention mechanism was proposed by (Bahdanau et al., 2014), it has been widely adopted to language and vision related tasks, such as Image Captioning and Visual Question Answering (VQA), by retrieving the visual features most related to text context (Zhu et al., 2016; Anderson et al., 2017; Xu and Saenko, 2016; Chen et al., 2015). (Xu et al., 2015) proposed to predict a word based on the visual patch that is most related to the last predicted word for image captioning. (Yang et al., 2016b; Lu et al., 2016) applied attention mechanism for VQA, to find the regions in images that are most related to the questions. (Yu et al., 2016) applied the visual attention mechanism on video captioning. Our attention implementation approach in this work is similar to those used for VQA. The model finds the regions in images that are most related to the accompanying sentences, and then feed the visual features into an BLSTM-CRF sequence labeling model. The differences are: (1) we add visual context feature at each step of sequence labeling; and (2) we propose to use a gate to control the combination of the visual information and textual information based on their relatedness.

6 Conclusions and Future Work

We propose a gated Visual Attention for name tagging in multimodal social media. We construct two multimodal datasets from Twitter and Snapchat. Experiments show an absolute 3%-4% F-score gain. We hope this work will encourage more research on multimodal social media in the future and we plan on making our benchmark available upon request.
soccer team, basketball team, politician, artist) can benefit more from visual features. For example, an image including a pitcher indicates that the ‘Giants’ in context should refer to the baseball team ‘San Francisco Giants’. We plan to expand our model to tasks such as fine-grained Name Tagging or Entity Liking in the future.

Acknowledgments

This work was partially supported by the U.S. DARPA AIDA Program No. FA8750-18-2-0014 and U.S. ARL NS-CTA No. W911NF-09-2-0053. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References


Timothy Baldwin, Marie-Catherine de Marneffe, Bo Han, Young-Bum Kim, Alan Ritter, and Wei Xu. 2015. Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition. In Proceedings of the Workshop on Noisy User-generated Text.


