information

MDPI

Article

Learning Improved Semantic Representations with Tree-Structured LSTM for Hashtag Recommendation: An Experimental Study

Rui Zhu †, Delu Yang † and Yang Li *

College of Information and Computer Engineering, Northeast Forestry University, Harbin 150040, China; zhubing@nefu.edu.cn (R.Z.); ydl855@nefu.edu.cn (D.Y.)

* Correspondence: yli@nefu.edu.cn

† These authors contributed equally to this work.

Received: 20 January 2019; Accepted: 3 April 2019; Published: 6 April 2019

Abstract: A hashtag is a type of metadata tag used on social networks, such as Twitter and other microblogging services. Hashtags indicate the core idea of a microblog post and can help people to search for specific themes or content. However, not everyone tags their posts themselves. Therefore, the task of hashtag recommendation has received significant attention in recent years. To solve the task, a key problem is how to effectively represent the text of a microblog post in a way that its representation can be utilized for hashtag recommendation. We study two major kinds of text representation methods for hashtag recommendation, including shallow textual features and deep textual features learned by deep neural models. Most existing work tries to use deep neural networks to learn microblog post representation based on the semantic combination of words. In this paper, we propose to adopt Tree-LSTM to improve the representation by combining the syntactic structure and the semantic information of words. We conduct extensive experiments on two real world datasets. The experimental results show that deep neural models generally perform better than traditional methods. Specially, Tree-LSTM achieves significantly better results on hashtag recommendation than standard LSTM, with a 30% increase in F1-score, which indicates that it is promising to utilize syntactic structure in the task of hashtag recommendation.

Keywords: hashtag recommendation; syntactic information; Tree-LSTM

1. Introduction

Social networking services (SNSs), such as Twitter, Facebook and Google+, have attracted millions of users to publish and share the most up-to-date information, emergent social events, and personal opinions [1]. As a typical representative of SNSs, microblog services have become increasingly popular. For example, Twitter [2] has more than 240 million active users, and users generate approximately 500 million tweets every day. The massive amounts of information generated every day makes it difficult to discover the hidden information in that data. To organize this information more accurately and effectively, many microblogging services allow users to create and use hashtags by placing the pound sign #, usually in front of a word or unspaced phrase in a post. The hashtag may contain letters, digits, and underscores. Searching for that hashtag yields each message that has been tagged with it. A hashtag archive is consequently collected into a single stream under the same hashtag. For example, the hashtag #HarryPotter allows users to find all the posts that have been tagged using that hashtag. It has also been proven that hashtags are important for many applications in microblogs, including microblog retrieval [3], query expansion [4], and sentiment analysis [5–7]. However, only about 11% of tweets were annotated with one or more hashtags by their authors [8]. Therefore, hashtag recommendation is a key challenge and has attracted much attention of researchers.

Existing approaches to hashtag recommendation range from classification and collaborative filtering to probabilistic models, such as naive Bayes and topic models. We formally formulate the hashtag recommendation task as a multi-class classification problem only using the text of posts. A typical approach is to learn the representation of a microblog post and then performing text classification based on this representation. Hence, it is of practical values to explore how to effectively represent the text of a microblog post for hashtag recommendation. We perform an experimental study of the short text representation methods in hashtag recommendation task. To be more specific, we study two major kinds of text representation methods for hashtag recommendation. For the first kind, we mainly consider shallow textual features, including bag-of-word (BoW) models and exquisitely designed patterns, which have been used in previous work [9,10]. However, feature engineering is labor-intensive and the *sparse* and *discrete* features cannot effectively encode semantic and syntactic information of words. Given neural network models can effectively learn feature representation, their advantages in recommendation research have gradually emerged. We further propose to learn deep textual features using multiple deep neural network models in hashtag recommendation task. In addition to the widely used Convolutional neural network (CNN) [11] and Long Short-Term Memory network (LSTM) [12] in text classification, we propose using Tree-LSTM [13] to introduce syntactic structure while learning the representation of microblog post. Experiments were designed on two real-world datasets to compare our method with many competitive baselines. It was found that Tree-LSTM outperformed all the baselines by incorporating the syntactic information. Specifically, Tree-LSTM achieved significantly better results in hashtag recommendation than standard LSTM, with a 30% increase in F1-score. In order to foster reproducibility of our experiments we also made our code [14] publicly available.

Our contributions are summarized as follows:

- We take the initiative to study how the syntactic structure can be utilized for hashtag recommendation, and our experimental results have indicated that the syntactic structure is indeed useful in the recommendation task. To the best of our knowledge, few studies address this task in a systematic and comprehensive way.
- We propose and compare various approaches to represent the text for hashtag recommendation, and extensive experiments demonstrate the powerful predictive ability of our deep neural network models on two real-world datasets.

2. Methodology for Hashtag Recommendation

We formulate the task of hashtag recommendation as a multi-class classification problem. We consider two major approaches: one extracts textual features using traditional text representation models, and the other learns effective text representations using deep neural network models. Specially, we propose to incorporate syntactic structure to enhance the representation learning of the text. In what follows, we introduce the two approaches: a traditional classification approach with shallow textual features and a neural network approach with deep and syntactic textual features respectively.

2.1. Traditional Approach with Shallow Textual Features

2.1.1. Naive Bayes (NB)

To solve the above recommendation problem, traditional machine learning algorithms such as naive Bayes (NB), was applied to model the posterior probability of each hashtag given a microblog in [15]. Similarly, we use a naive Bayes model to determine the relevance of hashtags to an individual tweet. Given an input microblog s with N words $w_1, ..., w_N$, the features of a tweet can be represented as a one-hot vector $\mathbf{x}$ where $x_i = 0$ or 1 to indicate the presence or absence of the ith dictionary word.

Using Bayes' rule, we can determine the posterior probability of C_i given the features of a tweet $x_1, ..., x_N$:

$$p(C_i|x_1, ..., x_N) = \frac{p(C_i)p(x_1|C_i)...p(x_N|C_i)}{p(x_1...x_N)} \quad (1)$$

In the simplest case, the prior probability $p(C_i)$ may be assumed the same for all C_i to give a maximum likelihood approach. The term $p(x_1...x_N)$ essentially serves as a normalizing constant and can be ignored when selecting the most probable hashtags because it does not depend on the hashtag C_i. Finally, using the naive Bayes assumption that the features x_i are conditionally independent given C_i, we can write the likelihood $p(x_1, ..., x_N|C_i)$ as the product $p(x_1|C_i)...p(x_N|C_i)$. Each of these conditional probabilities is based on the frequency with which each word appears in tweets with hashtag C_i.

2.1.2. Latent Dirichlet Allocation (LDA)

Following the same idea from [16], we use the standard LDA model [17] for the task. In LDA, each post s has a probability on a particular topic z_i, while each hashtag C_i is also assigned a probability belonging to each topic z_i. Therefore, we can obtain the probability of each hashtag C_i assigned to a post s by the combination of these two probabilities as follows:

$$P(C_i|s) = \sum_{j=1}^{Z} P(C_i|z_i = j)P(z_i = j|s) \quad (2)$$

2.1.3. Term Frequency-Inverse Document Frequency (TF-IDF)

The vector space model is an algebraic model to represent text as vectors of identifiers. It is widely used in information retrieval and relevancy ranking. We use the text of all posts in the dataset to generate the vocabulary V which contains all words. A microblog post feature vector $\mathbf{x} \in \mathbb{R}^{|V|}$ can be considered as a representation of all words in the post over the dictionary V. The weighting of a word in w_i is determined by term frequency-inverse document frequency (i.e., tf-idf) model [18]; then, based on the TF-IDF feature vectors, we build a multi-class SVM classification model [19] for hashtag recommendation.

2.2. Neural Network Approach with Deep Textual Features

It may be not effective enough if the above shallow textual features are directly used for classification task [20,21], since these features cannot capture the script beyond the surface form of the language. Inspired by the recent success of deep learning techniques in many NLP tasks, we further propose to learn deep textual features using multiple deep neural network models including FastText, CNN, LSTM and its variants in hashtag recommendation task. Specially, we propose to incorporate syntactic structure to enhance the representation learning of the text using Tree-LSTM model.

We typically use the neural models to learn the representation of a microblog post. Then, text classification is performed based on the representation of the microblog post. These deep neural network models are based on the fact that each word is represented as a low dimensional, continuous, and real-valued vector, which is also known as a word embedding [22,23]. All the word vectors are stacked in a word embedding matrix $L_w \in \mathbb{R}^{d_{emb} \times |V|}$, where d_{emb} is the dimension of word vector and $|V|$ is vocabulary size. To make better use of the semantic and grammatical associations of words, the values of the word vectors are pre-trained from the text corpus with the word2vec embedding learning algorithm [23]. Given an input microblog s, we take the embeddings $\mathbf{x}_t \in \mathbb{R}^{d_{emb} \times 1}$ for each word in the microblog to obtain the first layer. Hence, a microblog post of length N is represented with a sequence of word vectors $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N)$.

The output of deep neural network models is the representation vector of a microblog post *vec*. The final output vector *vec* is then fed to a linear layer whose output length is the number of hashtags. A softmax layer is finally added to output the probability distributions of all candidate hashtags. The softmax function is calculated as shown below, where C is the number of hashtag categories:

$$softmax(c_i) = \frac{exp(c_i)}{\sum_{i'=1}^{C} exp(c_{i'})} \tag{3}$$

2.2.1. FastText

FastText is a simple and efficient text classification method [24], which treats the average of word/n-gram embeddings as document embeddings, and then feeds document embeddings into a classifier. In Figure 1, the text representation is an hidden variable which can be potentially be reused. This architecture is similar to the CBOW (Continuous Bag-of-Words) model of [23], where the middle word is replaced by a label. We use the softmax function to compute the probability distribution over the predefined hashtags.

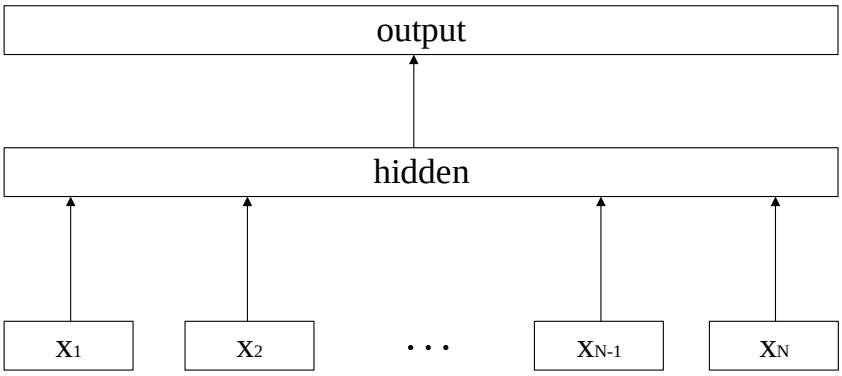

Figure 1. Model architecture of FastText for a post with N n-gram features $x_1, ..., x_N$. The features are embedded and averaged to form the hidden variable.

2.2.2. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) model [11] has subsequently achieved excellent results in many NLP tasks. The architecture of the CNN model is shown in Figure 2. In the training process, a convolution operation uses a filter $\mathbf{w}$ which has a window of h words to produce a new feature. For example, a feature f_i generated from a window h of words sequence $\mathbf{x}_{i:i+h-1}$ is defined as follows:

$$f_i = g(\mathbf{w} \cdot \mathbf{x}_{i:i+h-1} + b) \tag{4}$$

where b is a bias term and g is a non-linear function such as hyperbolic tangent. This filter is applied to each possible window of words in the sentence $x_{1:h}, x_{2:h+1}, ..., x_{n-h+1:n}$ to produce a feature map $\mathbf{f} = \{f_1, f_2, ..., f_{n-h+1}\}$. Then a max-pooling operation is applied on the feature map to capture the most important feature by taking the maximum value $\hat{f} = max\{\mathbf{f}\}$. Finally, we use a softmax function to generate the probability distribution over labels.

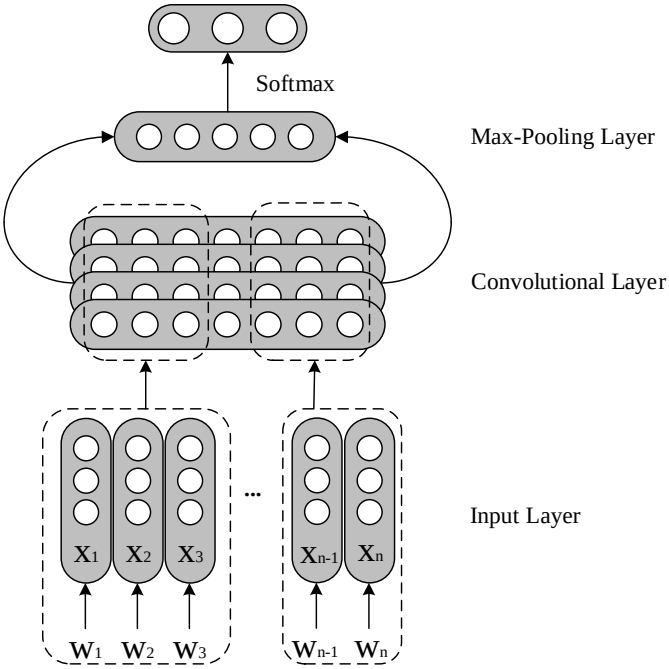

Figure 2. Hashtag recommendation with CNN model.

2.2.3. Standard LSTM

Long Short-Term Memory [12] is a special RNN that addresses the shortcomings of RNN in long-term dependencies. The transition equations for LSTM are as follows:

$$\begin{aligned}
\mathbf{i}_t &= \sigma(\mathbf{W}^i\mathbf{x}_t + \mathbf{U}^i\mathbf{h}_{t-1} + \mathbf{b}^i) \\
\mathbf{f}_t &= \sigma(\mathbf{W}^f\mathbf{x}_t + \mathbf{U}^f\mathbf{h}_{t-1} + \mathbf{b}^f) \\
\mathbf{o}_t &= \sigma(\mathbf{W}^o\mathbf{x}_t + \mathbf{U}^o\mathbf{h}_{t-1} + \mathbf{b}^o) \\
\widetilde{\mathbf{c}}_t &= \tanh(\mathbf{W}^c\mathbf{x}_t + \mathbf{U}^c\mathbf{h}_{t-1} + \mathbf{b}^c) \\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \widetilde{\mathbf{c}}_t \\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t)
\end{aligned} \tag{5}$$

where $\mathbf{h}_t$ represents the hidden layer, $\mathbf{x}_t$ represents the current word, and the last hidden layer $\mathbf{h}_{t-1}$ forms a new cell state $\widetilde{\mathbf{c}}_t$ with current new word $\mathbf{x}_t$. To determine the importance of the currently entered word, the input gate $\mathbf{i}_t$ uses the last hidden layer $\mathbf{h}_{t-1}$ and the current word $\mathbf{x}_t$. The forget gate $\mathbf{f}_t$ is opposite to the input gate $\mathbf{i}_t$, and it uses the last hidden layer $\mathbf{h}_{t-1}$ and the current word $\mathbf{x}_t$ to determine the importance of the old memory. The final cell state $\mathbf{c}_t$ is obtained by adding the new cell state $\widetilde{\mathbf{c}}_t$ under the condition of the input gate $\mathbf{i}_t$, and the previous cell state $\mathbf{c}_{t-1}$ under the condition of forget gate $\mathbf{f}_t$. The output gate $\mathbf{o}_t$ is used to distinguish the current final cell state $\mathbf{c}_t$ from the current hidden layer $\mathbf{h}_t$, $\odot$ denotes element wise multiplication, and $\mathbf{b}$ is the bias. As shown from the above-stated analysis, the hidden layer $\mathbf{h}_t$ summarizes the information of the whole sequence centered around $\mathbf{x}_t$.

The last hidden vector from the standard LSTM is used as the microblog representation to recommend hashtags. Then, it is fed to a linear layer whose output length is the number of hashtags. Finally, a softmax layer is added to output the probability distributions of all candidate hashtags. The structure of this model is shown in Figure 3.

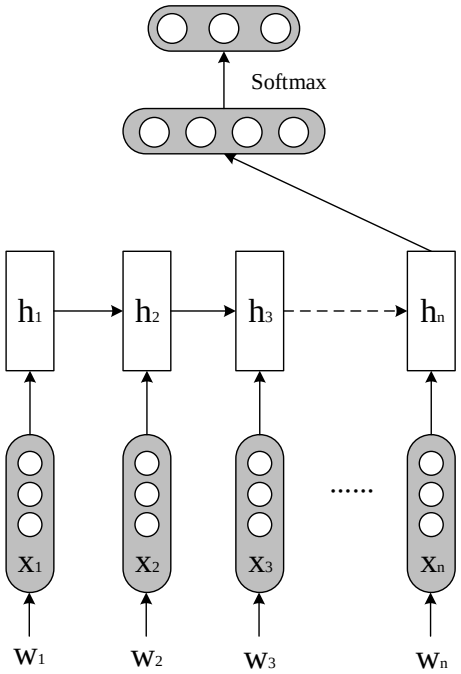

Figure 3. Hashtag recommendation with standard LSTM model.

2.2.4. LSTM with Average Pooling (AVG-LSTM)

However, a potential issue with standard LSTM approach is that all the necessary information of the input post has to be compressed into a fixed-length vector. This may make it difficult to cope with long sentences. Each annotation h_t contains information about the whole input microblog with a strong focus on the parts surrounding the tth word of the input microblog. One possible solution is to perform an average pooling operation over the hidden vectors of LSTM [12]. We call it AVG-LSTM for short.

2.2.5. Bidirectional LSTM (Bi-LSTM)

Bidirectional LSTM is an extension of traditional LSTM that enables the hidden states to capture both historical and future context information. In problems where all time steps of the input sequence are available, Bidirectional LSTM models text semantics both from forward and backward. For a microblog $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N)$, the forward LSTM reads sequence from $\mathbf{x}_1$ to $\mathbf{x}_N$ and the backward LSTM reads sequence from $\mathbf{x}_N$ to $\mathbf{x}_1$, and similarly processes the sequence according to Equation (5). Then we concatenate the forward hidden state $\overrightarrow{\mathbf{h}_t}$ and backward hidden state $\overleftarrow{\mathbf{h}_t}$, i.e., $\mathbf{h}_t = [\overrightarrow{\mathbf{h}_t}; \overleftarrow{\mathbf{h}_t}]$, where the $[\cdot;\cdot]$ denotes concatenation operation. Finally, the $\mathbf{h}_t$ summarizes the information of the whole sequence centred around $\mathbf{x}_t$.

2.2.6. Tree-LSTM

All the above deep neural models can learn deep textual features, however, they ignore the syntactic structures. We take the initiative to study how the syntactic structure can be utilized for hashtag recommendation with Tree structured LSTM model. It has been demonstrated that Tree-LSTM is able to improve the performance of many tasks, such as sentiment analysis [13], natural language inference [25], etc. In our task, the text is spoken language, which is quite different from the texts in previous research tasks. It is worth exploring that whether the syntactic information is effective short informal text like tweet.

The difference between the standard LSTM units and Tree-LSTM units is that gating vectors and memory cell updates are dependent on the states of possibly many child units. Unlike the standard

LSTM, in which each node of LSTM only has one son, Tree-LSTM has a forget gate for each child node, which selectively combines the semantic information of each child node. For example, if there are three words in a sentence, $\mathbf{c}_2$ and $\mathbf{c}_3$ are child nodes of $\mathbf{c}_1$ in the syntactic tree, the structure of LSTM units should be constructed as Figure 4, while the Tree-LSTM units should be constructed as Figure 5.

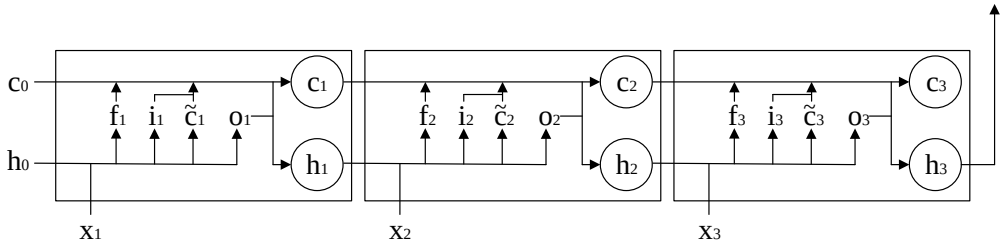

Figure 4. The structure of LSTM units with three nodes.

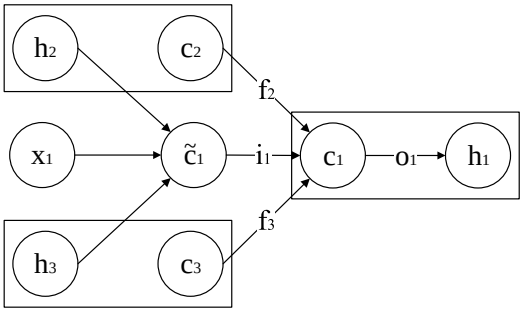

Figure 5. The structure of Tree-LSTM units with two child nodes.

Furthermore, if we use C_t to represent the set of all the children of current node t, then the transition equation of Child-Sum Tree-LSTM is as follows:

$$\begin{aligned}
\widetilde{\mathbf{h}}_t &= \sum_{k \in C(t)} \mathbf{h}_k \\
\mathbf{i}_t &= \sigma(\mathbf{W}^{(i)}\mathbf{x}_t + \mathbf{U}^{(i)}\widetilde{\mathbf{h}}_t + \mathbf{b}^{(i)}) \\
\mathbf{f}_{tk} &= \sigma(\mathbf{W}^{(f)}\mathbf{x}_t + \mathbf{U}^{(f)}\mathbf{h}_k + \mathbf{b}^{(f)}) \\
\mathbf{o}_t &= \sigma(\mathbf{W}^{(o)}\mathbf{x}_t + \mathbf{U}^{(o)}\widetilde{\mathbf{h}}_t + \mathbf{b}^{(o)}) \\
\widetilde{\mathbf{c}}_t &= \tanh(\mathbf{W}^{(c)}\mathbf{x}_t + \mathbf{U}^{(c)}\widetilde{\mathbf{h}}_t + \mathbf{b}^{(c)}) \\
\mathbf{c}_t &= \mathbf{i}_t \odot \widetilde{\mathbf{c}}_t + \sum_{k \in C(t)} \mathbf{f}_{tk} \odot \mathbf{c}_k \\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t)
\end{aligned} \tag{6}$$

As above, each Tree-LSTM node includes an input gate $\mathbf{i}_t$, multiple forget gates $\mathbf{f}_{tk}$ corresponding to multiple child nodes, an output gate $\mathbf{o}_t$, a memory cell $\mathbf{c}_t$, and a hidden layer $\mathbf{h}_t$. Additionally, $\mathbf{x}_t$ is the input at the current step, σ denotes the logistic sigmoid function, $\odot$ denotes element wise multiplication, $\mathbf{U}$ represents the weight matrix of the hidden layer of the children, $\mathbf{W}$ represents the weight matrix of the current step, and $\mathbf{b}$ is the threshold vector. The values of $\mathbf{W}$, $\mathbf{U}$, $\mathbf{b}$, and $\mathbf{h}$ are initially set to random numbers between $(-1.0, 1.0)$.

Syntactic analysis is one of the key underlying technologies in natural language processing (NLP). Its basic task is to determine the syntactic structure of a sentence or the dependency between words in a sentence. Dependency parsing is a typical approach of syntactic parsing in which syntactic units are arranged according to the dependency relation, as opposed to the constituency relation of phrase structure grammars. Therefore, in this task, we use the dependency tree structure as the syntactic information.

Through an example, this section introduces how to incorporate syntactic information into the Tree-LSTM architecture for the task of hashtag recommendation.

Example tweet : Am I the only one in the world that does not watch glee?

Given the tweet above, the Stanford Parser [26] is first used to obtain the dependency tree structure of the tweet, which is shown in Figure 6. However, there are some alternatives, such as the Berkeley parser [27], LTP [28], and FudanNLP [29].

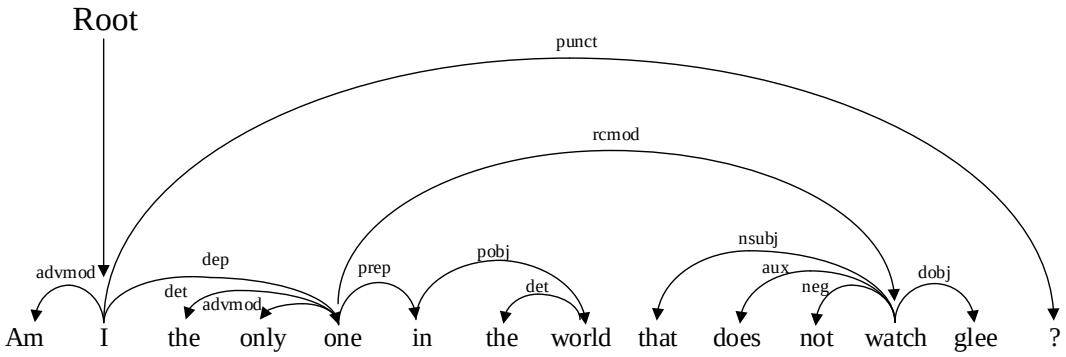

Figure 6. The dependency relations of the example tweet.

A tree structure for the example tweet in Figure 7 can then be obtained, corresponding to Figure 6.

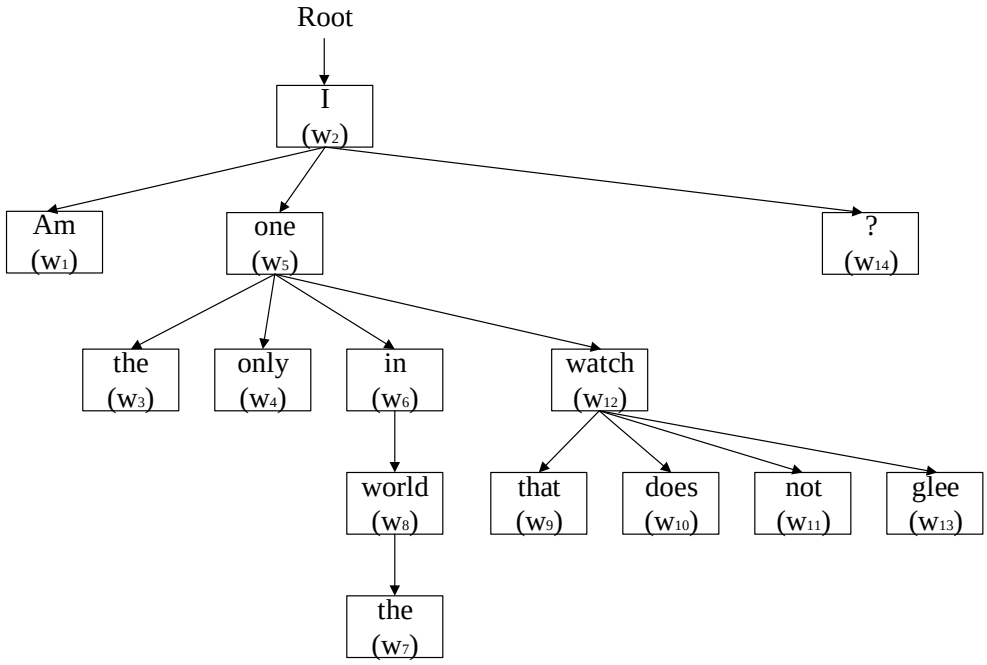

Figure 7. The syntactic tree structure of the example tweet.

The Tree-LSTM model in Figure 8 can be constructed based on the results of the tree structure. In this task, each x_j is a vector representation of a word in a sentence and h_j is the hidden representation of each word. As shown in Figure 7, "one" is the head word of "the", "only", "in" and "watch". Respectively, in Figure 8, h_5 is the parent node of h_3 , h_4 , h_6 and h_{12}, and so on. Each node from the upper layer in the tree takes the vector corresponding to the head word from the lower layer as input recursively. Finally, the hidden representation of the root in the tree is used as the representation of the tweet.

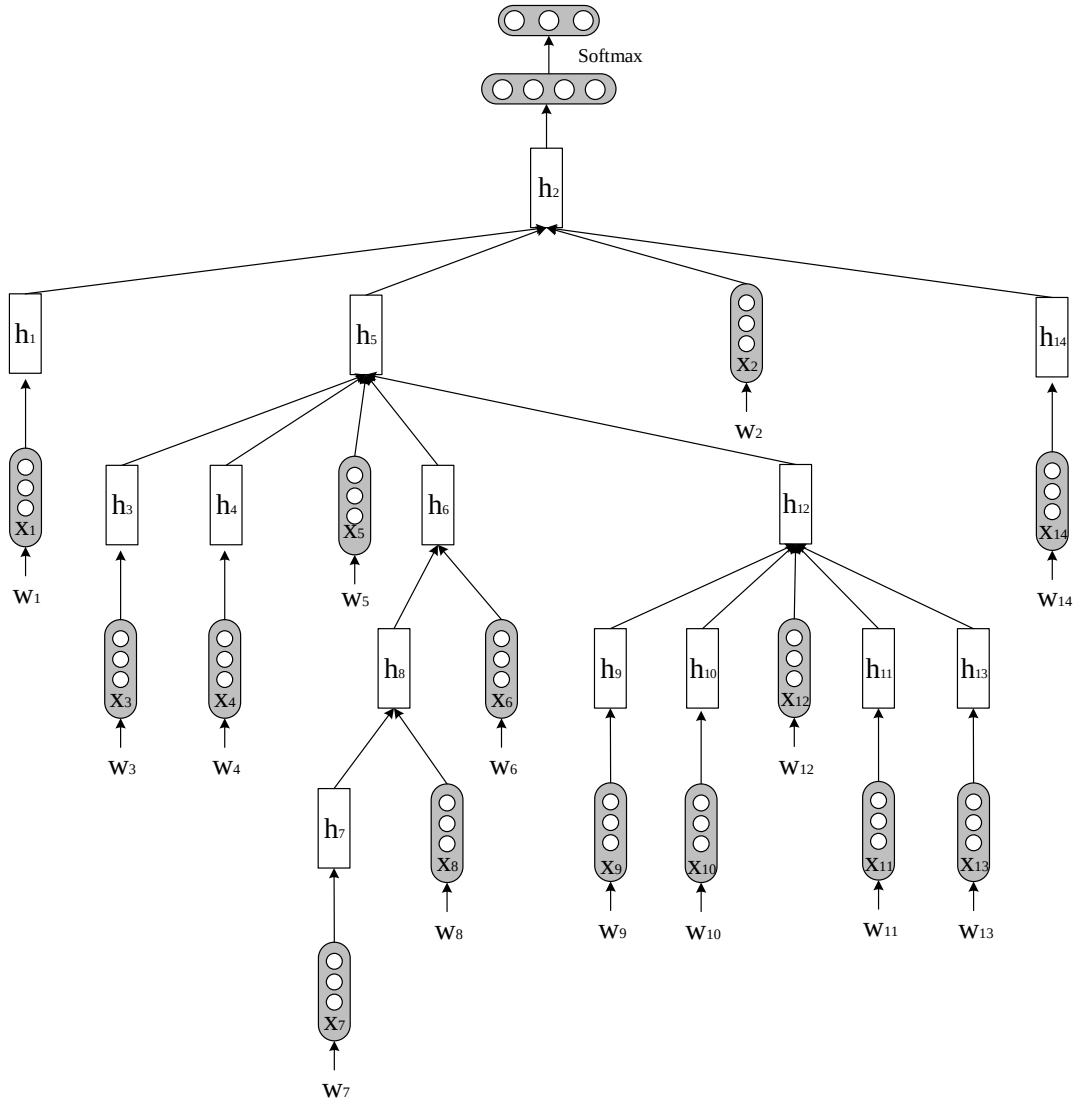

Figure 8. Hashtag recommendation with Tree-LSTM model.

2.2.7. Model Training

We train all our deep neural models in a supervised manner by minimizing the cross-entropy error of the hashtag classification. For tweets with more than one hashtag, each hashtag and the post are used as a training instance. The loss function is given below:

$$\mathcal{J} = -\sum_{s \in S} \sum_{t \in tags(s)} \log p(t|s) \tag{7}$$

where S stands for all training instances and $tags(s)$ is the hashtag collection for microblog s.

3. Experiment

We apply the proposed methods to the task of hashtag recommendation and evaluate the performance on two real-world datasets. This section describes the experiments that were designed to answer the research question that whether syntactic structure is useful for this task, and how much does syntactic structure help to improve the recommendation performance.

3.1. The Datasets

3.1.1. Twitter Dataset

The data used in this paper was constructed from a larger Twitter dataset ranging from June to December of 2009, with a total of 476,553,560 tweets [30]. The original data set was larger than 1 TB, and a sub dataset with 185,391,742 tweets from October to December were collected. Among them, there were 16,744,189 tweets including hashtags annotated by users. Due to the computational performance, we randomly select 50,000 tweets as training set, along with 5000 tweets as the validation and test set, which is similar to previous work [20,21,31]. We repeat this process for five times, and then take the average results on the test data as the final experimental results.

The statistics of our dataset is shown in Table 1.

Table 1. Statistics of the dataset, Nt(avg) is the average number of hashtags in the Twitter dataset.

# Tweets	# Hashtags	Vocabulary Size	Nt(avg)
60,000	14,320	106,348	1.352

Figures 9 and 10 show the distribution of the number of tweets for each hashtag and the distribution of the number of hashtags for each tweet, respectively, which follow a power-law distribution. From the two figures, the following can be learned: (1) more than 95% of the hashtags appeared in fewer than 20 tweets; and (2) about 10% of the tweets had more than one hashtag, while 98% of the tweets had less than five hashtags.

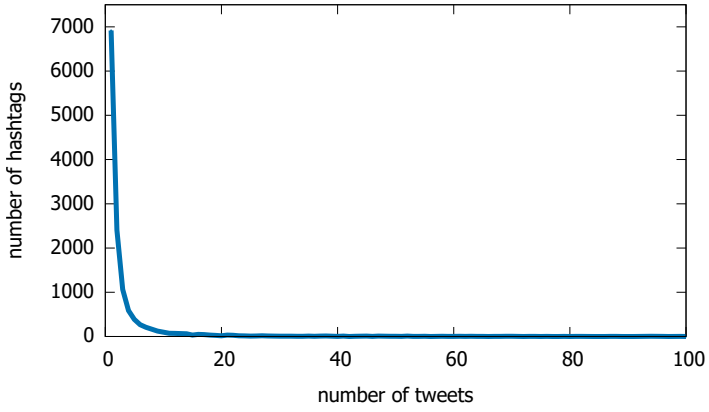

Figure 9. The distribution of the number of tweets for each hashtag.

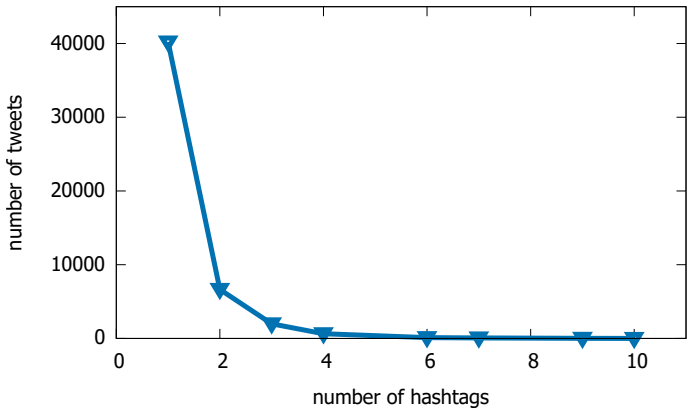

Figure 10. The distribution of the number of hashtags for each tweet.

3.1.2. Zhihu Dataset

The second dataset was crawled from Zhihu, a Chinese question-and-answer website. We collected our dataset from the Zhihu topic square [32]. We randomly crawled 150,000 hot questions and their hashtags from 100 subtopics under the 33 main topics. Then, the hashtags that appeared less than three times were removed as well as questions that had more than five hashtags. Finally, 278 42,060 questions and their 3487 hahstags remained. We randomly se80% questions as the training data, while 10% were used as the validation and 10% for test data. We repeat this process for five times, and then take the average results on the test data as the final experimental results. The statistics of the dataset are shown in Table 2 and Figures 11 and 12.

Table 2. Statistics of the dataset, Nt(avg) is the average number of hashtags in the Zhihu dataset.

# Questions	# Hashtags	Vocabulary Size	Nt(avg)
42,060	3487	37,566	3.394

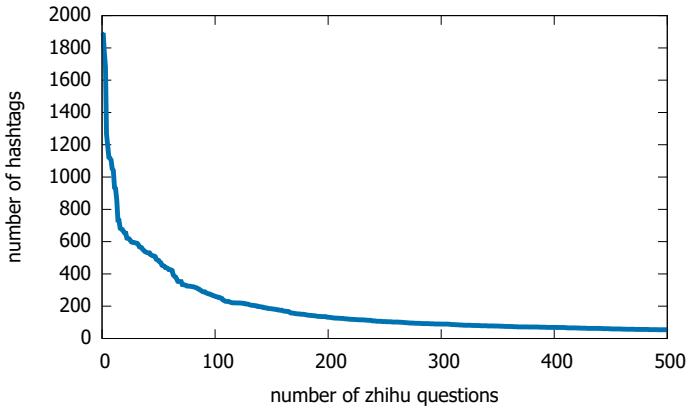

Figure 11. The distribution of the number of questions for each hashtag.

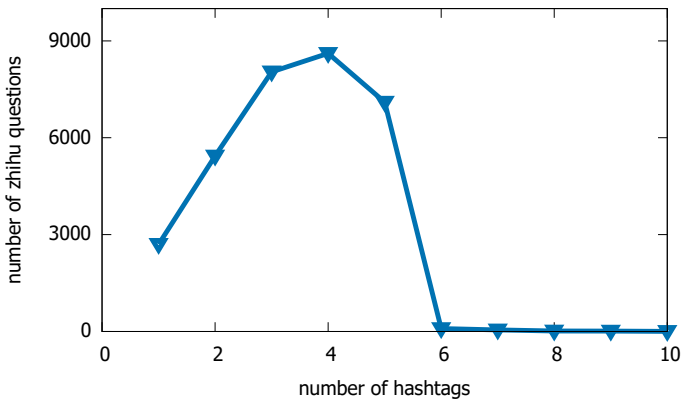

Figure 12. The distribution of the number of hashtags for each question.

3.2. Experimental Settings

We recommend hashtags in the following ways: we first use the training dataset to train our model and save the model, which has the best performance on the validation dataset. For the unlabelled data, we use the trained model to perform the hashtag classification. We repeat this process five times, and then take the average results on the test data as the final experimental results.

All the neural models were trained with sentences of lengths up to 50 words. For each of the above models, the dimension of all the hidden states in the LSTMs was set to 200 and the dimension of word embeddings was 300, unless otherwise noted. We use a minibatch stochastic gradient descent

(SGD) algorithm together with the Adam method to train each model [33]. The hyperparameter $\beta 1$ was set to 0.9 and $\beta 2$ was set to 0.999 for optimization. The learning rate was set to 0.001. The batch size was set to 50. The network was trained for 20 epochs with early stopping. For both our models and the baseline methods, we use the validation data to tune the hyperparameters, and report the results of the test data in the same setting of hyperparameters. Furthermore, the word embeddings used in all methods are pre-trained from the original twitter data released by [30] with the word2vec toolkit [23].

We use hashtags annotated by users as the golden set. To evaluate the performance, we use precision (P), recall (R), and F1-score (F) as the evaluation metrics. Precision means the percentage of "tags truly assigned" among "tags assigned by system". Recall denotes that "tags truly assigned" among "tags manually assigned". F1-score is the average of Precision and Recall. The same settings are adopted by previous work [20,21,31].

3.3. Experimental Results

Tables 3 and 4 show a comparison between the results of our method and the state-of-the-art discriminative and generative methods on the two datasets respectively. It is worth noting that we only take the hashtag with highest probability as the result predicted by the model. In other words, we only evaluate the top-1 recommendation result. To summarize, we find the following conclusions.

First, considering the comparison between the NB and TF-IDF methods, it can be observed that TF-IDF performed much better than NB. This indicates that the TF-IDF features were more effective than bag-of-words (BoW).

Secondly, for the general topic model LDA, due to the word-gap problem, sometimes the topic of the sentence was not captured well, and it could only obtain sparse features of sentences. LDA was able to achieve a little improvement in results over the NB and TF-IDF methods in Zhihu dataset. However, in Twitter dataset, the results of LDA is worse than TF-IDF, we suppose this is because the topics of twitter are more diverse.

Thirdly, the neural models had better experimental results than the traditional baseline methods, such as NB, TF-IDF and LDA. This indicates that neural networks are effective in learning the semantic information of microblog posts and can improve the performance considerably. Additionally, LSTM, AVG-LSTM and Bi-LSTM have produced comparable results.

Finally, Tree-LSTM, which incorporates the syntactic information, obtained the best results. Tree-LSTM achieved a more than 30% of relative improvement in the F1-score compared with LSTM and CNN. It is clear that the syntactic structure was effective in this task.

Tables 5 and 6 show the influence of the number of training data elements in two datasets. We randomly select a part of the training set according to a certain proportion. As shown, the performance of Tree-LSTM increased when larger training data sets were used. The results also demonstrate that our proposed method could achieve significantly better performance than the traditional baseline methods even with only 20% of the training data.

Table 3. Evaluation results of different methods for top-1 hashtag recommendation in Twitter dataset.

Methods	Precision	Recall	F1-Score
NB	0.137	0.117	0.126
LDA	0.190	0.163	0.176
TF-IDF	0.249	0.221	0.234
fastText	0.276	0.234	0.253
CNN	0.321	0.271	0.294
LSTM	0.509	0.443	0.473
AVG-LSTM	0.514	0.449	0.479
Bi-LSTM	0.513	0.447	0.478
Tree-LSTM	0.676	0.589	0.629

Table 4. Evaluation results of different methods for top-1 hashtag recommendation in Zhihu dataset.

Methods	Precision	Recall	F1-Score
NB	0.084	0.027	0.040
LDA	0.122	0.059	0.079
TF-IDF	0.100	0.032	0.048
fastText	0.347	0.109	0.166
CNN	0.394	0.129	0.194
LSTM	0.423	0.138	0.209
AVG-LSTM	0.437	0.144	0.217
Bi-LSTM	0.474	0.158	0.237
Tree-LSTM	0.522	0.176	0.263

Table 5. The influence of number of training data in Twitter dataset.

Trainingdata	Precision	Recall	F1-Score
100K (20%)	0.459	0.399	0.417
200K (40%)	0.532	0.459	0.480
300K (60%)	0.589	0.512	0.534
400K (80%)	0.633	0.550	0.574
500K (100%)	0.676	0.589	0.629

Table 6. The influence of number of training data in Zhihu dataset.

Trainingdata	Precision	Recall	F1-Score
100K (20%)	0.379	0.123	0.178
200K (40%)	0.442	0.148	0.211
300K (60%)	0.474	0.159	0.227
400K (80%)	0.502	0.171	0.242
500K (100%)	0.522	0.176	0.263

Many microblog posts have more than one hashtag; therefore, we also evaluated the top-k recommendation results of different methods. Figures 13–15 show the precision, recall, and F1 curves of NB, LDA, CNN, LSTM, AVG-LSTM, Bi-LSTM and Tree-LSTM on the test data respectively. Each point of a curve represents the extraction of a different number of hashtags, ranging from 1 to 5. It can be observed that although the precision and F1-score of Tree-LSTM decreased when the number of hashtags was larger, Tree-LSTM still outperformed the other methods. Moreover, the relative improvement on extracting only one hashtag was higher than that on more than one hashtag, which shows that it is more difficult to recommend hashtags for a microblog post with more than one hashtag.

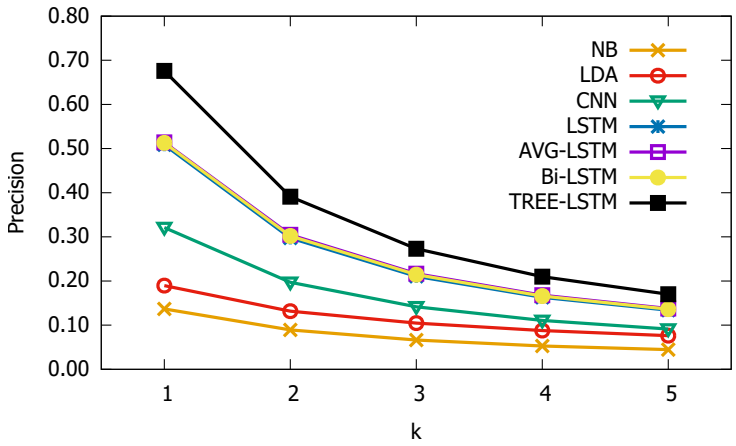

Figure 13. Precision with recommended hashtags range from 1 to 5 in Twitter dataset.

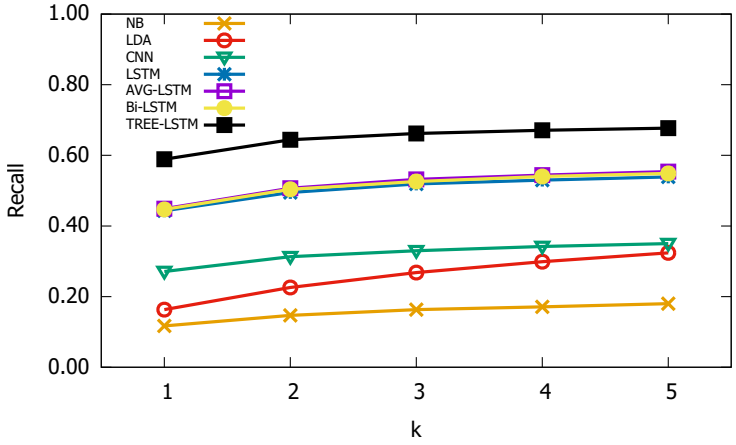

Figure 14. Recall values with recommended hashtags range from 1 to 5 in Twitter dataset.

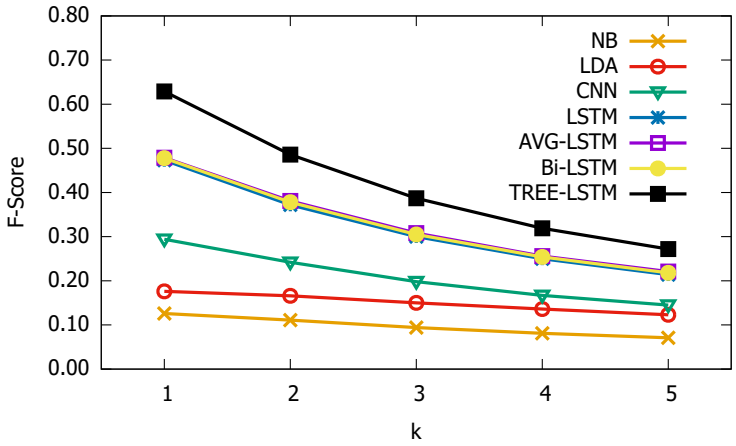

Figure 15. F1 values with recommended hashtags range from 1 to 5 in Twitter dataset.

Similarly, we also evaluated the top-*k* recommendation results of different methods of Zhihu dataset in Figures 16–18. It can be observed that Tree-LSTM still outperformed the other methods substantially in all cases. However, for all methods, the performance of the Zhihu dataset is worse than that of the twitter dataset. We suppose this is because the average number of hashtags in the Zhihu dataset is large.

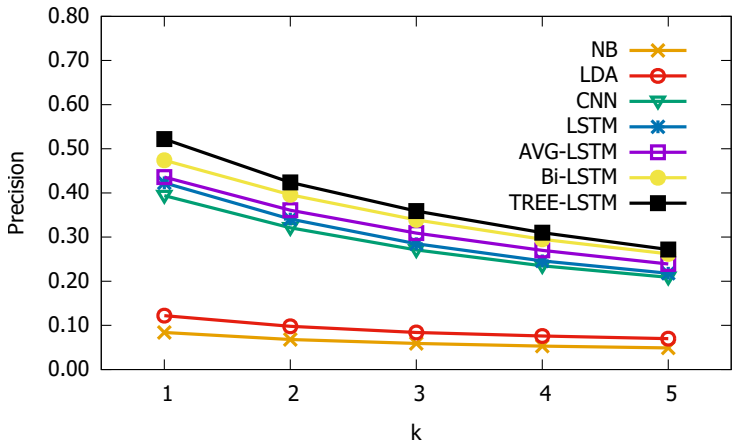

Figure 16. Precision with recommended hashtags range from 1 to 5 in Zhihu dataset.

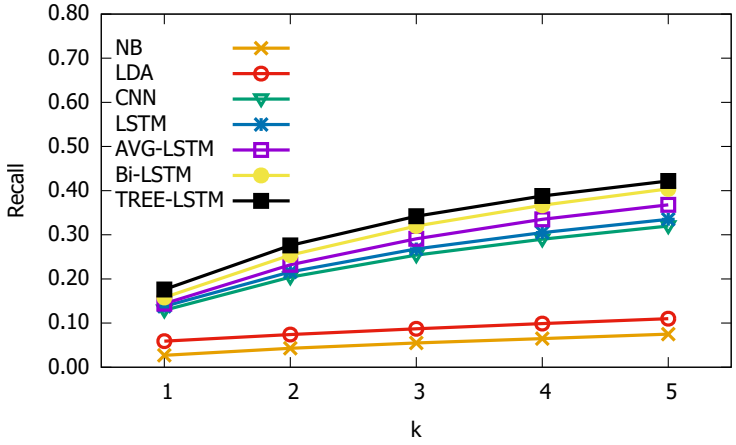

Figure 17. Recall values with recommended hashtags range from 1 to 5 in Zhihu dataset.

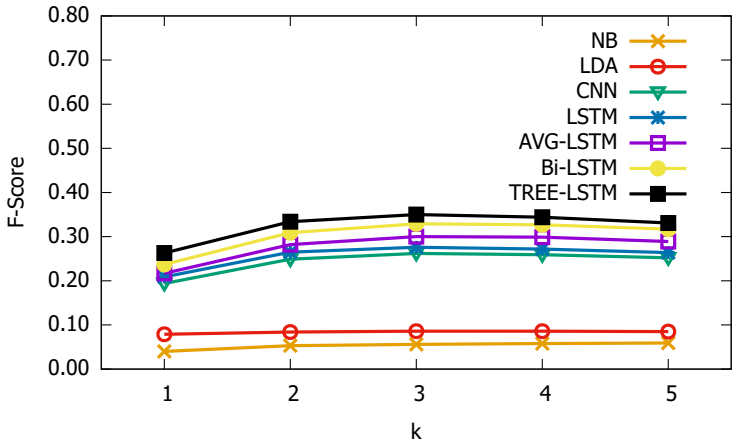

Figure 18. F1 values with recommended hashtags range from 1 to 5 in Zhihu dataset.

3.4. Parameter Sensitive Analysis

We further investigate the effect of hyperparameters to the performance in Twitter dataset. We vary the dimension of hidden layers in Tree-LSTM from 50 to 300, with a gap of 50. As shown in Figure 19, we find the performance improved when the dimension of hidden layer increased.

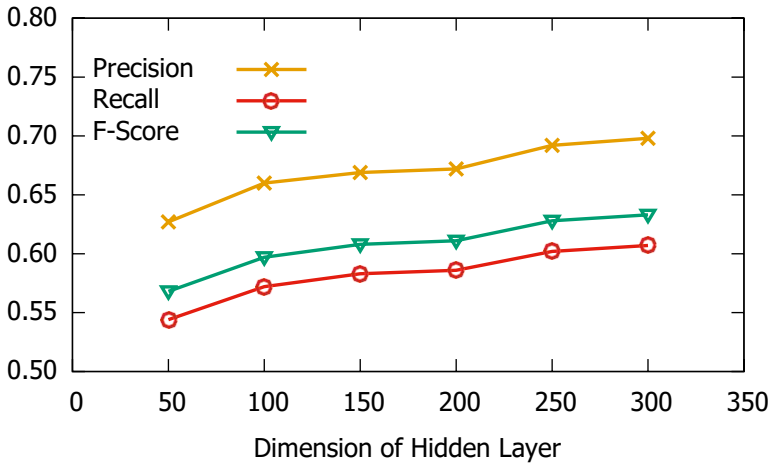

Figure 19. Results of Tree-LSTM with dimension of hidden layers range from 50 to 300.

3.5. Qualitative Analysis

A qualitative analysis of our results was also conducted through case studies in Twitter data.

Example 1: Worldcup football fans have their say on the draw—socceroos. #worldcup

Example 2: Gon na watch some flashforward now, while waiting for parcelforce to arrive hopefully! #flashforward

In both examples, the hashtag #worldcup and #flashforward were correctly recommended by Tree-LSTM; however, they were not recommended by LSTM. As shown in Figure 20, the hashtag "worldcup" acts as a noun phrase in the sentence. In Figure 21, the hashtag "flashforward" is an object in a verb phrase. Through the analysis of the recommendation results, it was found that Tree-LSTM has a strong advantage over the standard LSTM model in the above two cases. These indicate that the syntactic structure is useful in this task.

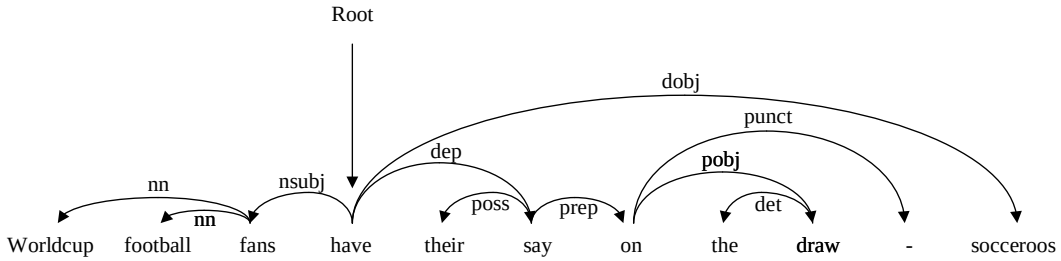

Figure 20. Syntactic parsing tree of Example 1.

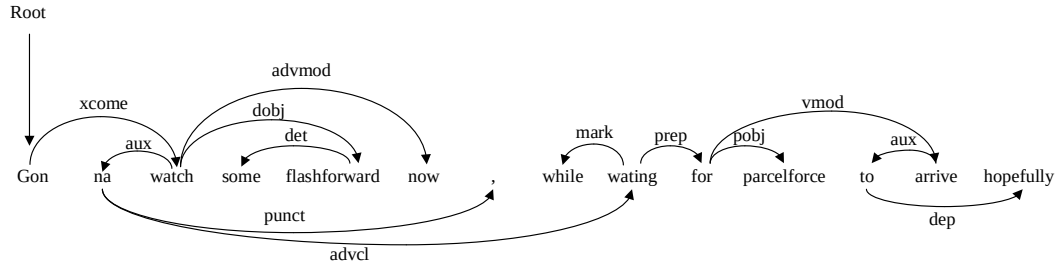

Figure 21. Syntactic parsing tree of Example 2.

4. Related Work

There has been much research in the field of hashtag recommendation. The practical approaches can be roughly divided into two categories: content-based methods and collaborative filtering methods. Content-based methods use different techniques to measure the content of the tweets to find the relevant hashtags from historical tweets [9,10]. Zangerle et al. [9] presented an approach based on the similar microblogs and the hashtags in these microblogs. Mazzia and Juett [15] used the naive Bayesian model to calculate the maximum posterior probability of a hashtag under the condition that all words in the microblog are known. Krestel et al. [16] used a topic model based on the Latent Dirichlet Allocation (LDA) [17] to train the tags and the words in the text together to obtain the topic distribution of each document and the topic distribution of the words. Ding et al. [34] treated the text and its labels as different descriptions of the same resource, and used the topic model and single-word alignment techniques to learn the probability of each word being aligned with the label under a particular topic. When performing the prediction task, they calculated the ranking score of a certain hashtag relative to each microblog according to the probability distribution of the topic and the word alignment probability.

Collaborative filtering is a popular technique for recommendation systems. It considers user preferences by building a model from a user's past behaviours as well as similar decisions made by other users. Kywe et al. [35] proposed a collaborative filtering model based on a user group similar to the target user or based on a microblog similar to the target microblog to obtain the hashtags.

Wang et al. [36] proposed a joint model based on topic modelling and collaborative filtering to take advantages of both local (the current microblog content and the user) and global (hashtag-related content and like-minded users' usage preferences) information. Zhao et al. [37] used a hashtag-LDA recommendation approach combines user profile-based collaborative and LDA-based collaborative filtering. They jointly model the relations between users, hashtags, and words through latent topics.

In addition to these works, some researchers have attempted to incorporate external information to improve the recommendation. Li et al. [38] incorporated the topic enhanced word embeddings, tweet entity data, tag frequency, tag temporal data and tweet URL domain information to recommend the hashtags for microblogs. Ma et al. [39] explored the latent relationship between tweet content, user interest, time, and tag. Motivated by the observation that the average hashtag experiences a life cycle of increasing and decreasing popularity, Lu et al. [40] proposed a model captures the temporal clustering effect of latent topics in tweets. Kowald et al. [41] proposed a cognitive-inspired hashtag recommendation algorithm that is based on temporal usage patterns of hashtags derived from empirical evidence (individual hashtag reuse patterns and social hashtag reuse patterns).

Recently, there have been some attempts to use deep neural models for hashtag recommendation [21,31,42]. Gong et al. [31] proposed a method incorporating textual and visual information. After observing that hashtags indicate the primary topics of microblog posts, Li et al. [20] proposed an attention-based LSTM model that incorporates topic modelling into the LSTM architecture through an attention mechanism. In this paper, we take the initiative to study how to effectively represent the text of a microblog post for hashtag recommendation. We study two major kinds of text representation methods for hashtag recommendation, including shallow textual features and deep textual features learned by deep neural models. Most existing work tries to use deep neural networks to learn microblog post representation based on the semantic combination of words. Specially, we propose to adopt Tree-LSTM to improve the representation by combining the syntactic structure and the semantic information of words. To the best of our knowledge, this is the first attempt to introduce syntactic information into a neural network for this task.

5. Conclusions

This paper performs an experimental study of how to effectively represent the text of a microblog post for hashtag recommendation. We study two major kinds of text representation methods including traditional approach with shallow textual features and neural network approach with deep textual features. In addition to the widely used CNN, LSTM models in text classification, we propose using Tree-LSTM to introduce syntactic structure while learning the representation of microblog post. The experimental results on two real-world datasets demonstrated that Tree-LSTM had the best performance, which indicates that it is promising to utilize syntactic structure in the task of hashtag recommendation.

For future work, we will try to incorporate more external information into the task, such as time, author occupation and user relationships.

Author Contributions: Conceptualization, Yang Li; Methodology, Y.L.; Software, D.Y.; Validation, R.Z. and D.Y.; Formal Analysis, R.Z.; Investigation, D.Y.; Resources, D.Y.; Data Curation, R.Z.; Writing—Original Draft Preparation, R.Z.; Writing—Review & Editing, Y.L.; Visualization, D.Y. and R.Z.; Supervision, Y.L.; Project Administration, Y.L.; Funding Acquisition, R.Z., D.Y. and Y.L.

Funding: This research was funded by [the National Undergraduate Training Programs for Innovation and Entrepreneurship] via grant number [201810225182], [the National Natural Science Foundation of China (NSFC)] via grant number [61806049] and [the Natural Science Foundation of Heilongjiang Province of China] via grant number [F2018001].

Acknowledgments: We thank the anonymous reviewers for their constructive comments.

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

References

1. Tran, V.C.; Hwang, D.; Nguyen, N.T. Hashtag Recommendation Approach Based on Content and User Characteristics. *Cybern. Syst.* **2018**, *49*, 368–383. [CrossRef]
2. Twitter. Available online: https://www.twitter.com (accessed on 4 April 2019).
3. Efron, M. Hashtag retrieval in a microblogging environment. In Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Geneva, Switzerland, 19–23 July 2010; pp. 787–788.
4. Bandyopadhyay, A.; Ghosh, K.; Majumder, P.; Mitra, M. Query expansion for microblog retrieval. *Int. J. Web. Sci.* **2012**, *1*, 368–380. [CrossRef]
5. Davidov, D.; Tsur, O.; Rappoport, A. Enhanced sentiment learning using twitter hashtags and smileys. In Proceedings of the 23rd International Conference on Computational Linguistics: Posters, Beijing, China, 23–27 August 2010; pp. 241–249.
6. Wang, X.; Wei, F.; Liu, X.; Zhou, M.; Zhang, M. Topic sentiment analysis in twitter: A graph-based hashtag sentiment classification approach. In Proceedings of the 20th ACM International Conference on Information and Knowledge Management, Glasgow, UK, 24–28 October 2011; pp. 1031–1040.
7. Li, Y.; Jiang, J.; Liu, T.; Sun, X. Personalized Microtopic Recommendation with Rich Information. In Proceedings of the 4th National Conference on Social Media Processing (SMP 2015), Guangzhou, China, 16–17 November 2015; pp. 1–14.
8. Hong, L.; Ahmed, A.; Gurumurthy, S.; Smola, A.J.; Tsioutsiouliklis, K. Discovering Geographical Topics in the Twitter Stream. In Proceedings of the 21st International Conference on World Wide Web, Lyon, France, 16–20 April 2012; pp. 769–778. [CrossRef]
9. Zangerle, E.; Gassler, W.; Specht, G. Recommending#-tags in twitter. In Proceedings of the 2nd International Workshop on Semantic Adaptive Social Web (SASWeb 2011), Girona, Spain, 15 July 2011; Volume 730, pp. 67–78.
10. Godin, F.; Slavkovikj, V.; De Neve, W.; Schrauwen, B.; Van de Walle, R. Using Topic Models for Twitter Hashtag Recommendation. In Proceedings of the 22nd International Conference on World Wide Web, Rio de Janeiro, Brazil, 13–17 May 2013; pp. 593–596. [CrossRef]
11. Kim, Y. Convolutional Neural Networks for Sentence Classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1746–1751.
12. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
13. Tai, K.S.; Socher, R.; Manning, C.D. Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks. *arXiv* **2015**, arXiv:1503.00075.
14. yangdelu855/Tree-LSTM. Available online: https://github.com/yangdelu855/Tree-LSTM (accessed on 4 April 2019).
15. Mazzia, A.; Juett, J. *Suggesting Hashtags on Twitter*; EECS 545m: Machine Learning; Computer Science and Engineering, University of Michigan: Ann Arbor, MI, USA, 2009.
16. Krestel, R.; Fankhauser, P.; Nejdl, W. Latent Dirichlet Allocation for Tag Recommendation. In Proceedings of the Third ACM Conference on Recommender Systems (RecSys '09), New York, NY, USA, 23–25 October 2009; pp. 61–68. [CrossRef]
17. Blei, D.M.; Ng, A.Y.; Jordan, M.I. Latent Dirichlet Allocation. *J. Mach. Learn. Res.* **2003**, *3*, 993–1022.
18. Robertson, S. Understanding inverse document frequency: On theoretical arguments for IDF. *J. Doc.* **2004**, *60*, 503–520. [CrossRef]
19. Hearst, M.A.; Dumais, S.T.; Osman, E.; Platt, J.; Scholkopf, B. Support vector machines. *IEEE Intell. Syst. Their Appl.* **1998**, *13*, 18–28. [CrossRef]
20. Li, Y.; Liu, T.; Jiang, J.; Zhang, L. Hashtag Recommendation with Topical Attention-Based LSTM. In Proceedings of the 26th International Conference on Computational Linguistics, Osaka, Japan, 11–16 December 2016; pp. 3019–3029.
21. Gong, Y.; Zhang, Q. Hashtag Recommendation Using Attention-Based Convolutional Neural Network. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI 2016), New York, NY, USA, 9–15 July 2016; pp. 2782–2788.

22. Bengio, Y.; Ducharme, R.; Vincent, P.; Jauvin, C. A neural probabilistic language model. *J. Mach. Learn. Res.* **2003**, *3*, 1137–1155.
23. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. In Proceedings of the Advances in Neural Information Processing Systems (NIPS 2013), Lake Tahoe, NV, USA, 5–10 December 2013; pp. 3111–3119.
24. Joulin, A.; Grave, E.; Bojanowski, P.; Mikolov, T. Bag of Tricks for Efficient Text Classification. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Valencia, Spain, 3–7 April 2017; Volume 2, pp. 427–431.
25. Chen, Q.; Zhu, X.; Ling, Z.; Wei, S.; Jiang, H. Enhancing and Combining Sequential and Tree LSTM for Natural Language Inference. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017; pp. 1657–1668.
26. Chen, D.; Manning, C. A Fast and Accurate Dependency Parser using Neural Networks. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 740–750.
27. Greg, D.; Dan, K. Neural CRF Parsing. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, China, 26–31 July 2015; pp. 302–312.
28. Che, W.; Li, Z.; Liu, T. LTP: A Chinese Language Technology Platform. In Proceedings of the 23rd International Conference on Computational Linguistics, Beijing, China, 23–27 August 2010.
29. Ji, F.; Gao, W.; Qiu, X.; Huang, X. FudanNLP: A Toolkit for Chinese Natural Language Processing with Online Learning Algorithms. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Sofia, Bulgaria, 4–9 August 2013; pp. 49–54.
30. Yang, J.; Leskovec, J. Patterns of Temporal Variation in Online Media. In Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, Hong Kong, China, 9–12 February 2011; pp. 177–186. [CrossRef]
31. Gong, Y.; Zhang, Q.; Huang, X. Hashtag recommendation for multimodal microblog posts. *Neurocomputing* **2018**, *272*, 170–177. [CrossRef]
32. Zhihu. Available online: https://www.zhihu.com/topics (accessed on 4 April 2019).
33. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
34. Ding, Z.; Zhang, Q.; Huang, X. Automatic hashtag recommendation for microblogs using topic-specific translation model. In Proceedings of the 24th International Conference on Computational Linguistics, COLING 2012, Mumbai, India, 8–15 December 2012; pp. 265–274.
35. Kywe, S.M.; Hoang, T.A.; Lim, E.P.; Zhu, F. On Recommending Hashtags in Twitter Networks. In *Social Informatics, Proceedings of the 4th International Conference, SocInfo 2012, Lausanne, Switzerland, 5–7 December 2012*; Aberer, K., Flache, A., Jager, W., Liu, L., Tang, J., Guéret, C., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7710, pp. 337–350.
36. Wang, Y.; Qu, J.; Liu, J.; Chen, J.; Huang, Y. What to Tag Your Microblog: Hashtag Recommendation Based on Topic Analysis and Collaborative Filtering. In *Web Technologies and Applications, Proceedings of the 16th Asia-Pacific Web Conference, APWeb 2014, Changsha, China, 5–7 September 2014*; Chen, L., Jia, Y., Sellis, T., Liu, G., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2014; Volume 8709, pp. 610–618.
37. Zhao, F.; Zhu, Y.; Jin, H.; Yang, L.T. A personalized hashtag recommendation approach using LDA-based topic model in microblog environment. *Future Gener. Comput. Syst.* **2016**, *65*, 196–206. [CrossRef]
38. Li, Q.; Shah, S.; Nourbakhsh, A.; Liu, X.; Fang, R. Hashtag Recommendation Based on Topic Enhanced Embedding, Tweet Entity Data and Learning to Rank. In Proceedings of the 25th ACM International Conference on Information and Knowledge Management (CIKM '16), Indianapolis, IN, USA, 24–28 October 2016; pp. 2085–2088. [CrossRef]
39. Ma, Z.; Sun, A.; Yuan, Q.; Cong, G. Tagging Your Tweets: A Probabilistic Modeling of Hashtag Annotation in Twitter. In Proceedings of the 23rd ACM International Conference on Information and Knowledge Management (CIKM '14), Shanghai, China, 3–7 November 2014; pp. 999–1008. [CrossRef]
40. Lu, H.; Lee, C. A Twitter Hashtag Recommendation Model that Accommodates for Temporal Clustering Effects. *IEEE Intell. Syst.* **2015**, *30*, 18–25. [CrossRef]

41. Kowald, D.; Pujari, S.C.; Lex, E. Temporal Effects on Hashtag Reuse in Twitter: A Cognitive-Inspired Hashtag Recommendation Approach. In Proceedings of the 26th International Conference on World Wide Web Companion (WWW'17), Perth, Australia, 3–7 April 2017; pp. 1401–1410. [CrossRef]
42. Zhang, Q.; Wang, J.; Huang, H.; Huang, X.; Gong, Y. Hashtag Recommendation for Multimodal Microblog Using Co-Attention Network. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17), Melbourne, Australia, 19–25 August 2017.