



Fine-grained entity type classification with adaptive context

Jin Liu¹ · Lina Wang¹ · Mingji Zhou¹ · Jin Wang^{2,3} · Sungyoung Lee⁴

Published online: 25 January 2018
© Springer-Verlag GmbH Germany, part of Springer Nature 2018

Abstract

Natural language processing (NLP) is the technology that enables machine to process human language. Entity recognition is one of the most basic tasks in NLP. It aims to identify and classify the name of each object in the text. Traditional named entity recognition systems can only identify a small set of types such as person, location, organization or miscellaneous. In order to make machine exploit the meaning of the text better, it is necessary to classify the entities appearing in the text to fine-grained types. Previously reported work generally obtained the entity context information through a fixed window, so the external information for classifying the entity is not enough, which may lead to ambiguity. To solve the shortcomings of these methods, this paper presents a fine-grained entity type classification method for unstructured text based on global information and sliding window context. By combining those information with other features, we utilize a bidirectional long–short-term memory network to perform the classification work. With the proposed method, the experiment results of fine-grained entity type classification are optimized.

Keywords Entity · Fine-grained · Classification · LSTM

1 Introduction

To develop automatic tools that can deal with large amount of text information efficiently is a traditional research topic.

Communicated by J. Park.

✉ Jin Liu
jinliu@shmtu.edu.cn

Lina Wang
lnwang@shmtu.edu.cn

Mingji Zhou
mjzhou@shmtu.edu.cn

Jin Wang
jinwang@yzu.edu.cn

Sungyoung Lee
sylee@oslab.khu.ac.kr

- ¹ College of Information Engineering, Shanghai Maritime University, Shanghai, China
- ² Key Lab of Broadband Wireless Communication and Sensor Network Technology (Nanjing University of Posts and Telecommunications), Ministry of Education, Nanjing, China
- ³ College of Information Engineering, Yangzhou University, Yangzhou, China
- ⁴ Computer Engineering Department, Kyung Hee University, Suwon, Korea

Natural language processing (NLP) is such a technology that enables machine to process human language. NLP has been quickly evolved with the development of artificial intelligence technology. With the fast growing of Web, the scale of unstructured text online is also increasing with an astonishing speed. Thus, machine learning, an automatic learning method based on data, has been proposed to solve this problem. At the same time, some applications such as information extraction and machine translation have emerged. To implement such applications, many innovative technologies are proposed. Among them, named entity recognition is a key task to fulfill and it is the basis of text information comprehension and processing.

However, traditional named entity recognition only considers how to identify entities in a small set of types. Therefore, the task of entity type classification is becoming more popular with the growing expectation of the researcher. This task aims to assign respective semantic types to entity mentions in their context. MUC-7 (Krupka and Hausman 1998) defined the three common types: Person, Organization, and Location. CoNLL03 added a miscellaneous type (Sang and Meulder 2003). Recent work also suggests that we can use a larger set of fine-grained types to make improvement for the NLP applications.

Existing fine-grained type classification systems have used approaches obtaining the entity local context information through a fixed window. However, local context with fixed window may lead to ambiguity because of the inadequate external information. To solve this problem, this paper proposed a novel fine-grained entity type classification method for unstructured text based on adaptive context information which not only includes global information, the local context in a sliding window is also used. The proposed method can locate the context information by finding the structure of the sentence structure, and the context information of entities can be accessed more efficiently. Therefore, by mining the text of the paragraph where the entity is located and using automatic summarization technique to obtain the global information of the text, our proposed method can reduce the ambiguity of entity classification, and the accuracy of entity classification can be improved. Experiments on the proposed methods have been taken on two public datasets: FIGER and OntoNotes. The validity of the proposed method can be proved by comparing the experimental results with the results of the related methods. The loose micro-F score for our method is 75.35% on FIGER dataset and 65.35% on OntoNotes dataset.

The remaining of this paper is organized as the follows: Sect. 2 presents the related researches of the fine-grained entity type classification in the literature, Sect. 3 describes the details of our proposed method, Sect. 4 presents the experiment results, and Sect. 5 draws some conclusion.

2 Related work

Most named entity recognition systems only support a small set of types. However, it is far from adequate for NLP tasks. For example, in the question answering task, we need to know the exact type of the candidate answers such as *Event*, *Tools* or *Product*. So the task of fine-grained type classification is widely researched in the literature.

Recently, many researches have focused on classifying the entity mentions in the text to fine-grained types. Although recently researchers proposed ways to deal with vague knowledge (Singh and Kumar 2015), most of current studies still assume the knowledge is well defined. Fleischman and Hovy (2002) classified mentions into eight subtypes of *Person* type by a decision tree based on local contextual word features and WordNet synonyms. Giuliano and Gliozzo (2008) proposed a method to further classifying the entity into 21 subtypes of *Person* type. To the best of our knowledge, the first one to do the fine-grained entity type classification was Lee et al. (2006). They defined 147 fine-grained types. Their main purpose was to apply the types into question answering systems and they trained and evaluated a conditional random field model on a manually annotated Korean dataset. Sekine (2009) defined 200 coarse types which could

serve as primitives for fine-grained types. And they emphasized the necessity of a large set of types for entity type classification. Rahman and Ng (2010) defined a type system which contained 29 types and 92 subtypes.

Xiao and Weld (2012b) derived 112 types from Freebase and automatically created the training data from Wikipedia by distant supervision method (Mintz et al. 2009). In addition, they created both a training and evaluation dataset FIGER of newspaper articles. And then, they demonstrated that their system could improve the accuracy of relation extraction system by fine-grained types. Nevertheless, there is an argument that fine-grained types should be classified in a hierarchical taxonomy. So Yosef et al. (2015) organized 505 types from YAGO (Hoffart et al. 2013) in such a hierarchical taxonomy, and the deepest one reached 9 layers. The results could be improved by using this set of types on FIGER dataset. In addition, they developed a multi-label hierarchical classification system. Corro et al. (2015) used a similar method to introduce a system which is the most fine-grained entity type classification system until now, and it covered more than 16,000 types in the WordNet hierarchy.

Most of the above methods assumed that type classification could be done independently without context information of entity mention. (Gillick et al. 2016) first introduced the fine-grained classification with context information. The type labels were limited to what could be deduced from entity mention context. Moreover, they introduced a new OntoNotes-derived manually annotated evaluation dataset and addressed the label noise problem which is induced by distant supervision. Ren et al. (2016) proposed a method to further reducing the label noise, and the performance on the FIGER dataset and OntoNotes dataset was improved. Moreover, Yogatama et al. (2015) had proposed a method to map the manually crafted features and type labels to embeddings so that the information can be shared between both related types and features. (Munkhdalai et al. 2015) proposed an Active Co-Training Algorithm for Biomedical Named-Entity Recognition. The proposed method tends to efficiently exploit a large amount of unlabeled data by selecting a small number of examples that have useful information and comprehensive pattern. Viswanathan and Krishnamurthi (2012) presented a Modified bidirectional breadth-first search algorithm for finding paths between two entities which pass through other intermediate entities and the paths are ranked according to the users' needs. Vijayarajan et al. (2016) proposed a generic framework for ontology-based information retrieval and image retrieval in web data.

Different from previous models that relied on manually crafted features, Dong et al. (2015) first introduced a hybrid neural model without which consisted of two parts to classify entity mentions to a wide-coverage set of 22 types derived from DBpedia. They used recurrent neural networks to recursively obtain the entity mention representation and used

multi-layers perceptron to obtain the context representation. This model didn't use any external resources. After that, Shimaoka et al. (2016, 2017) used recursive neural networks to compose context representations and employed an attention mechanism to allow the model to focus on relevant expressions. On this basis, they (Shimaoka et al. 2017) combined learnt and manually crafted features and used a hierarchical encoding of labels that enabled us to share parameters between labels in the same hierarchy. Recently, Gotti and Langlais (2016) described a recall-oriented open information extraction system designed to extract knowledge from French corpora. Their research is the first one that focus on such a cross-domain, recall-oriented approach in open information extraction. Cui et al. (2017) proposed a hybrid neural network model for type classification of entity mentions with a fine-grained taxonomy. Experimental results demonstrate that our model achieves state-of-the-art performance on the FIGER dataset. Barua and Patel (2017) proposed to use Search Engine's Query suggestion as external knowledge source instead of gazetteers for named entity classification in NER systems. The experiments on MSM Challenge dataset demonstrate that QS-NEC is efficient in classification of entity mentions and can be effectively used in NER systems.

3 The model for fine-grained entity type classification with adaptive context

3.1 Overall model

We proposed a novel LSTM based model to achieve the objective of fine-grained type classification. We first define entity mention as follows:

$$E_i \in E \quad (1 < i < E_{\text{num}}) \tag{1}$$

where E represents a set of entity and E_{num} is the size of the set. Then we need to get the local context of the entity mention through the sentence structure. So the context can be defined as follows:

$$l_i \in L \quad (1 < i < K) \tag{2}$$

$$r_i \in R \quad (1 < i < R_{\text{num}}) \tag{3}$$

where L represents the word set of left context of entity mention, K represents the size of L , R represents the word set of right context of entity mention, and T represents the size of R .

After that, we find the global context with sentence and obtain the abstract through automatic summarization technique. Each word in the abstract can be defined as follows:

$$g_i \in G \quad (1 < i < G_{\text{num}}) \tag{4}$$

where G represents the set of words in the abstract, G_{num} is the size of the set. In addition, we should obtain the manually crafted features of each entity mention. We will describe it in detail in the following section.

Finally, we merge the following four parts and feed them into two dense layers, then input the outputs from dense layers into the Softmax Layer to compute the probability:

- entity mention representation r_e ,
- local context representation r_c ,
- global context representation r_g and
- manually crafted feature representation r_f .

For each input x_i has a unique label y_i :

$$\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\} \quad y_i \in \{1, 2, \dots, D\} \tag{5}$$

$$x = [r_e \ r_c \ r_g \ r_f] \tag{6}$$

where n is the number of input and D is the size of the set of label. Given an input x , we can compute the probability $p(y = d | e)$ for each type d :

$$h(x_i) = \begin{bmatrix} p(y_i = 1 | x_i; W_y) \\ p(y_i = 2 | x_i; W_y) \\ \dots \\ p(y_i = D | x_i; W_y) \end{bmatrix} = \frac{1}{\sum_{d=1}^D e^{W_{yd}x_i}} \begin{bmatrix} e^{(W_{y1}x_i)} \\ e^{(W_{y2}x_i)} \\ \dots \\ e^{(W_{yD}x_i)} \end{bmatrix} \tag{7}$$

We first assign the type d to the entity if y_d is the maximum. Then we assign the additional types d that y_d is greater than a threshold γ so that the type d can be predicted. The overview of our model is shown in Fig. 1.

3.2 Manually crafted features

For each entity mention, there will be a binary feature indicator vector $f(e) = \{0, 1\}^{D_f \times 1}$. We can feed it to the Softmax Layer with the other three representations. The manually crafted features are shown in Table 1. The example that used in Table 1 to extract the manually crafted features is "... the person who [Brack H.Obama] first picked ...". The features we used are similar with Gillick (Dan et al. 2016) and Yogatama (Yogatama et al. 2015), and the same as Shimaoka (Shimaoka et al. 2017). In this paper, we use the clustering method (Brown et al. 1992) which is more widely that can make the clusters publicly available. In addition, we use LDA (Blei et al. 2003) to learn a set of 15 topics.

We map the vector $f(e)$ to a low-dimensional projection to compute the manually crafted feature representation $r_f \in R^{D_f \times 1}$:

$$r_f = W_f f(e) \tag{8}$$

where $W_f \in R^{D_f \times D_f}$ is the mapping matrix.

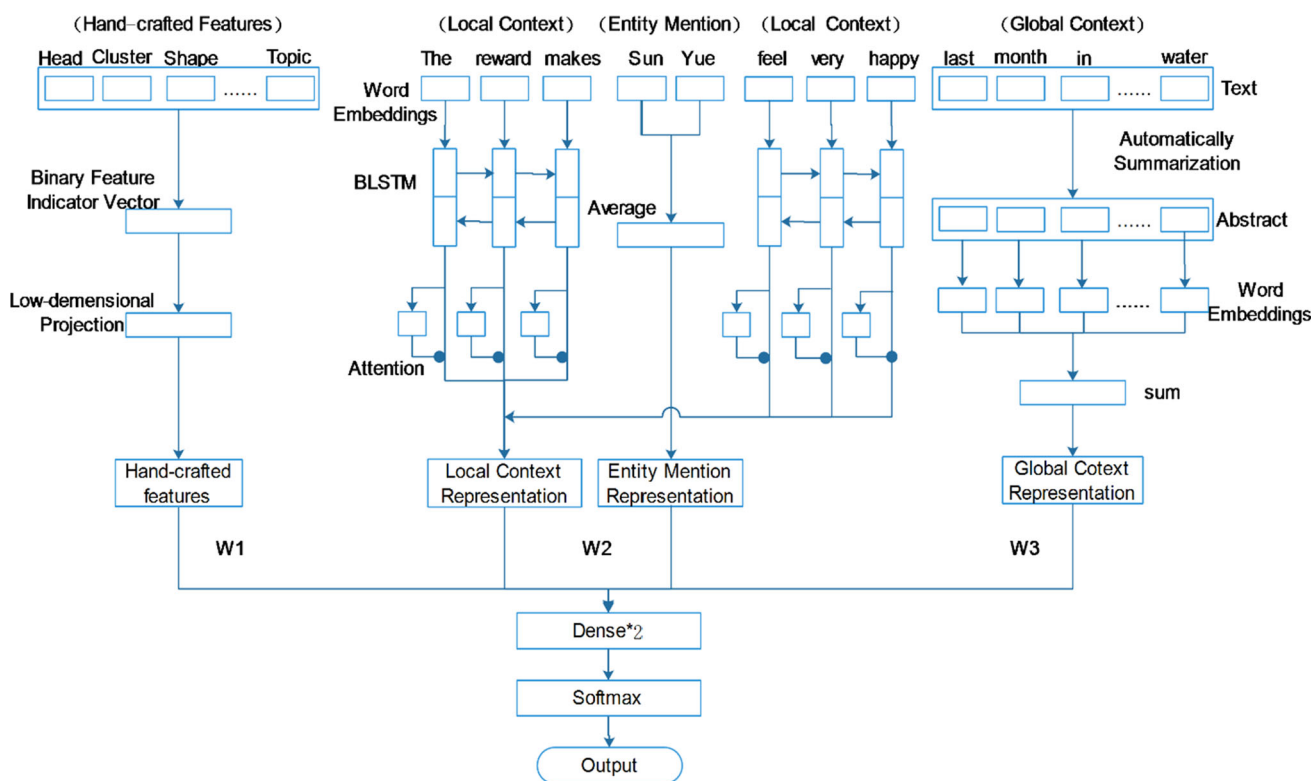


Fig. 1 Model of fine-grained entity type classification

Table 1 Manually crafted features

Feature	Description	Example
Head	Syntactic head of the mention	Obama
Non-head	Non-head words of the mention	Barack,H.
Cluster	Brown cluster for the head token	1110, ...
Characters	Character trigrams for the mention head	ob, oba, ...
Shape	Word shape of the mention phrase	Aa A.Aa
Role	Dependency label on the mention head	Subj
Context	Words before and after the mention	B: who, A: first
Parent	The head's lexical parent	Picked
Topic	The LDA topic of the document	LDA:13

3.3 Entity mention representation

Given an entity mention, we need to compute the average of all the embeddings of the words in the entity mention. Because an entity cannot just be composed by one word. For example, “New York” is an entity mention which consists of two words. Formally speaking, the word of entity mention is $e_i (1 \leq i \leq e_{num})$, e_{num} is the size of the entity mention. Then we compute the entity mention representation as follows:

$$r_e = \frac{1}{e_{num}} \sum_{i=1}^{e_{num}} u(e_i) \tag{9}$$

where u is a mapping from word to an embedding. $r_e \in R^{D_e \times 1}$ is the mention representation. We compute the representation in such a way since the previous method may lead to overfitting.

3.4 Local context representation

Given an entity mention, we need to obtain its local context information to predict its type. In previous work, most type classification methods used fixed window size on both left and right of entity mention to get context. However, the local contextual information obtained in such a way may lead to missing of key information if the sentence length is too long.

To solve this problem, in our method we employ a sliding window mechanism that can change the window size adaptively. Specifically, we determine the window size to obtain the local context by determining the boundary of sentence. Formally speaking, assume the left side context is l_1, \dots, l_K and right side context is r_1, \dots, r_T , where K is the window size of the left context while T is the windows size of the right context in the sentence. The specific steps to get adaptive context are shown in Algorithm 1.

Algorithm 1: Process to Obtain Context

```

Input: sentence, entity
Output: context
Begin
  ContextObtain(entity, sentence){
    flag = 0;
    For each in sentence {
      if ( != entity && flag == 0){
        left.append();
      }
      else if ( ==entity){
        flag = 1;
      }
      else if ( != entity && flag == 1){
        right.append ();
      }
    }
    Return left, right
  }
  }
    
```

We use the combination of bidirectional LSTMs (Graves 2012) and attention mechanism to compute the local context representation. Compared with traditional LSTM, bidirectional LSTMs can get much more information for it considers the words order in the sentence. And attention mechanism can let the model pay more attention to relevant expressions. The computation is as follows:

First, the outputs of the bidirectional LSTMs are $\vec{h}_1^l, \overleftarrow{h}_1^l, \dots, \vec{h}_K^l, \overleftarrow{h}_K^l$ (left context) and $\vec{h}_1^r, \overleftarrow{h}_1^r, \dots, \vec{h}_T^r, \overleftarrow{h}_T^r$ (right context). For each output layer, we use a two-layer feed-forward neural network $v_i \in R^{D_a \times 1}$ and weight matrices $W_d \in R^{D_a \times 2D_h}$ and $W_a \in R^{1 \times D_a}$ to compute a scalar value a_i^l :

$$v_i^l = \tanh \left(W \begin{bmatrix} \vec{h}_i^l \\ \overleftarrow{h}_i^l \end{bmatrix} \right) \tag{10}$$

$$a_i^l = \exp \left(W_a v_i^l \right) \tag{11}$$

Then, we normalize the scalar values so that the sum is 1:

$$a_i^l = \frac{a_i^l}{\sum_{i=1}^K a_i^l + \sum_{i=1}^T a_i^r} \tag{12}$$

The scalar values $a_i \in R$ are called attentions. Similarly, the right context is computed in the same way. Finally, we

compute the sum of output of the bidirectional LSTM as the local context representation:

$$r_c = \sum_{i=1}^K a_i^l \begin{bmatrix} \vec{h}_i^l \\ \overleftarrow{h}_i^l \end{bmatrix} + \sum_{i=1}^T a_i^r \begin{bmatrix} \vec{h}_i^r \\ \overleftarrow{h}_i^r \end{bmatrix} \tag{13}$$

3.5 Global context representation

Given an entity mention and the sentence that contains the entity, the document which contains those sentences is traditionally treated as global context. Nevertheless, a document always has redundant information or noise by itself. To avoid this problem, we apply automatic summarization for the objective document to get the most relevant information.

The automatic summarization algorithm we used in this paper added the semantic information and simplified steps for traditional algorithm based on rules to improve the accuracy and simplicity of the abstract. The flowchart of the algorithm is shown in Fig. 2.

Firstly, we need to create a word chart, the node of the word chart can be a word, a sentence or even a document. In this paper, we select the word to be the node and we use the vocabulary co-occurrence to construct the edge of the graph. If two words occur in the same sentence, we will connect these two word nodes. This method is called vocabulary co-occurrence. However, too many words in the document may lead to too many edges. And there has no relation between two distant words in the same sentence, so that it may result in a lot of interference edge. In order to address this problem, we optimize it by the introduction of dependency parser. The comparison of two methods is shown in Table 2. Then, we can construct the word graph. The example used in the word graph is ‘‘Alice, who had been reading about Spacy, saw Bob in the library.’’ And it is shown in Fig. 3. After completing the

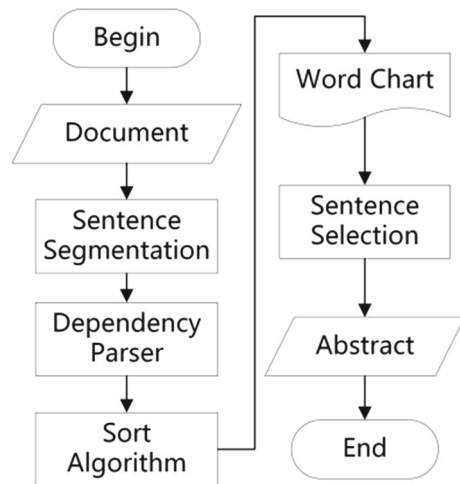


Fig. 2 Flowchart of the automatic summarization algorithm

Table 2 Comparison of co-occurrence method and dependency parser method to construct a word graph

Thailand attracts China tourists.	
Co-occurrence	Dependency parser
<Thailand, abstracts>	
<Thailand, tourists>	nsubj (abstracts, Thailand)
<Thailand, China>	dobj (abstracts, tourists)
<abstracts, China>	
<abstracts, tourists>	
<China, tourists>	

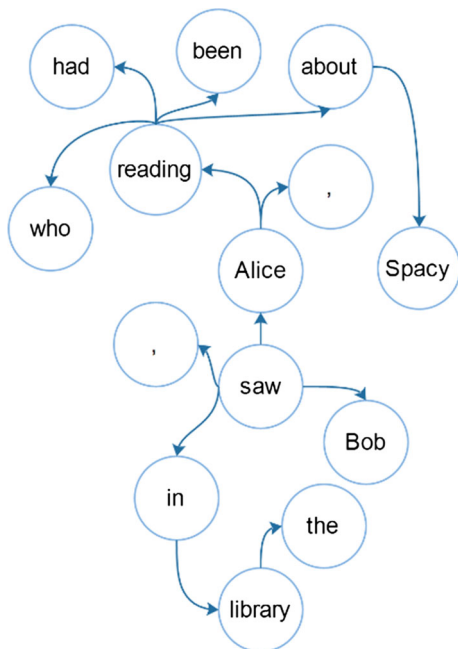


Fig. 3 An example of a word graph

word graph construction, we need to compute the importance of words. This paper use the classical graph model algorithm HITS (Kleinberg 1999). The basic idea of the algorithm is to enhance the relationship with each others, and the specific assumptions are as follows:

- Authority Score: A node is an important node if it point to many important nodes.
- Hub Score: A node is an important node if it is pointed to by many important nodes.

The formulas of the two assumptions are as follows:

$$HITS_A(V_i) = \sum_{V_j \in In(V_i)} HITS_H(V_j) \tag{14}$$

$$HITS_H(V_i) = \sum_{V_j \in Out(V_i)} HITS_A(V_j) \tag{15}$$

Let the sum of the authority score and hub score be the importance score of the word. And then, we compute the sum of the importance score of each word as the importance score of the sentence:

$$Score(S_i) = \sum_{w_i \in S_i} Score(w_i) \tag{16}$$

Finally, we select the top-ranked sentence as the abstract of the document.

3.6 Hierarchical label encoding

Fine-grained entity type classification has a big difference from traditional classification in that it tend to form a forest of type hierarchies. For example, teacher is a subtype of education, while education is a subtype of person. We can enable the parameter sharing by hierarchical label encoding, because some co-occurrence labels will be closer in this space. For instance, the candidate entity mention labels are person, artist and location. So the type person and artist are closer. Concretely, we compute the weight matrix W_y for the Softmax Layer with learnt weight matrix V_y and a constant sparse binary matrix S :

The illustration of hierarchical label encoding is shown in Fig. 4. Each type is mapped to a unique column in S . For example, the column for /person is encoded as [1, 0, 0, 0, ...], /person/education is encoded as [1, 1, 0, 0, ...], and /person/education/student is encoded as [1, 1, 1, 0, ...].

This method can make the parameters be shared between labels in the same hierarchy.

$$W_y^T = V_y S \tag{17}$$

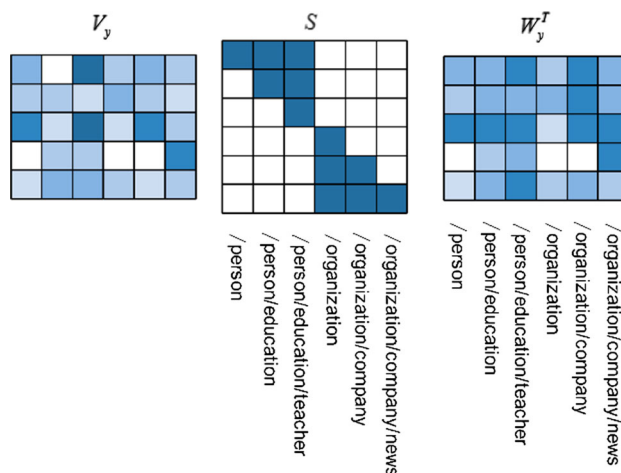


Fig. 4 An illustration of hierarchical label encoding

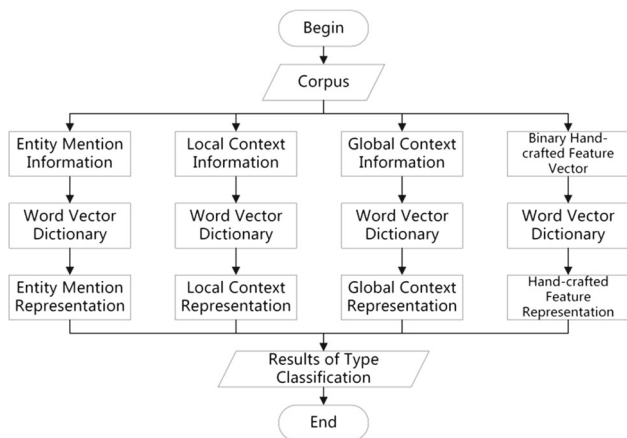


Fig. 5 Flowchart of the experiment

4 Experiment results and analysis

4.1 Overview

The overall flowchart of our experiment is shown in Fig. 5. The experiment can be divided into several parts as follows:

1. After dealing with the corpus, we can obtain the entity mention representation and local context representation easier.
2. Getting the global context representation by automatic summarization technique.
3. The four representations is fed into the model as input. Then, we can adjust the parameters to obtain the best model. Finally, we use the test set to evaluate our model.

In our experiment, we use Ubuntu 16.04 as experimental environment. And we use Python 2.7 as our development language. In addition, we use Keras 2.0.4 and TensorFlow 0.11 as our framework. We employ the lower version of TensorFlow for better compatibility with FIGER and OntoNotes Datasets.

4.2 Dataset and word embedding

To train and evaluate our model, we use two publicly available datasets. One is OntoNotes (Dan et al. 2016) which consists of 13,109 news documents where 77 test documents are manually annotated. The other is FIGER (Xiao and Weld 2012a) which contains 112 fine-grained types. The specific types of two datasets are shown in Figs. 6 and 7, respectively.

We use 300-dimensional cased word embeddings trained on 840 billion tokens by Glove algorithm (Pennington et al. 2014). We use the pre-trained word embeddings to converge more quickly, saving training time and improve the accuracy

of the model. For the absent words in the pre-trained word embeddings, we use the embeddings of the “unk” token.

4.3 Entity mention and context

Entity mention and local context can be obtained in one sentence, so we can get them jointly. First, we tag the location of the entity mention and manually crafted features. We obtain the entity mention by its location. Then, we label the word “BEG” at the beginning of the sentence and the word “END” at the end of the sentence to find the boundary of the sentence so that we can obtain the local context of the entity mention.

As to the global context, first, we find the document that contains the sentence occurs. Then, we construct the word graph with dependency parser for each sentence after segmenting the document text to sentences. After that, we obtain the abstract using the method mentioned before.

One of the most important tasks for processing text in English is to identify the punctuation marks. Full stop “.” is usually used in English to represent the abbreviated word. For example, the word “a.m” is the meaning of morning, but this “.” cannot represent the end of a sentence. Another similar example is the sentence “I like U.S.A.”. In this sentence, the last “.” represents not only the end of the sentence but also the abbreviated word “U.S.A.”. In this paper, we use Punkt Sentence Tokenizer of NLTK to complete the segmentation. It detects the sentence boundary without semantic information, so it can handle the abbreviation problem well.

Then, we use Stanford Dependency Parser tool in NLTK to do the task of dependency parser. An example for dependency parser is shown in Table 3.

After screening for various dependency relations, this paper selects several classical relations to construct graph model. The relations selected in this paper are shown in Table 4.

Then we use HITs algorithm to compute the weight of each sentence. After ranking the sentence by their weights, we select the top-two sentences as the abstract in this paper.

4.4 Model parameter settings

The parameter setting of our proposed model as shown in Fig. 1 is shown in Tables 5 and 6. Table 5 represents the parameter settings for the entire network. Table 6 represents the parameter settings for network layer details.

Loss function is a function which aims to evaluate the degree of inconsistency between the predicted value and the true value. In this paper, we adopt the cross entropy loss function as our loss function. Formally speaking, for each input x_i has a unique label y_i :

<p>PERSON</p> <div style="border: 1px solid black; padding: 2px;"> <p>artist actor author director music</p> </div> <div style="border: 1px solid black; padding: 2px;"> <p>education student teacher</p> </div> <p>athlete business coach doctor legal military political figure religious leader title</p>	<p>LOCATION</p> <div style="border: 1px solid black; padding: 2px;"> <p>structure airport government hospital hotel restaurant sports facility theatre</p> </div> <div style="border: 1px solid black; padding: 2px;"> <p>geography body of water island mountain</p> </div> <div style="border: 1px solid black; padding: 2px;"> <p>transit bridge railway road</p> </div> <p>figure religious leader title</p>	<p>ORGANIZATION</p> <div style="border: 1px solid black; padding: 2px;"> <p>company broadcast news</p> </div> <p>education government military music political party sports league sports team stock exchange transit</p>	<p>OTHER</p> <div style="border: 1px solid black; padding: 2px;"> <p>art broadcast film music stage writing</p> </div> <div style="border: 1px solid black; padding: 2px;"> <p>event accident election holiday natural disaster protest sports event violent conflict</p> </div> <div style="border: 1px solid black; padding: 2px;"> <p>award body part currency</p> </div> <div style="border: 1px solid black; padding: 2px;"> <p>language programming - language</p> </div> <div style="border: 1px solid black; padding: 2px;"> <p>living thing animal</p> </div> <div style="border: 1px solid black; padding: 2px;"> <p>product camera car computer mobile phone software weapon</p> </div> <p>food heritage internet legal religion scientific sports & leisure supernatural</p>
---	--	---	---

Fig. 6 Types in OntoNotes dataset

<p>person actor architect artist athlete author coach director</p>	<p>doctor engineer monarch musician politician religious_leader soldier terrorist</p>	<p>organization airline company educational_institution fraternity_sorority sports_league sports_team</p>	<p>terrorist_organization government_agency government political_party educational_department military news_agency</p>				
<p>location city country county province railway road bridge</p>	<p>body_of_water island mountain glacier astral_body cemetery park</p>	<p>product engine airplane car ship spacecraft train</p>	<table border="1"> <tr> <td style="vertical-align: top;"> <p>camera mobile_phone computer software game instrument weapon</p> </td> <td style="vertical-align: top;"> <p>art written_work newspaper music</p> </td> </tr> <tr> <td style="vertical-align: top;"> <p>event attack election protest</p> </td> <td style="vertical-align: top;"> <p>military_conflict natural_disaster sports_event terrorist_attack</p> </td> </tr> </table>	<p>camera mobile_phone computer software game instrument weapon</p>	<p>art written_work newspaper music</p>	<p>event attack election protest</p>	<p>military_conflict natural_disaster sports_event terrorist_attack</p>
<p>camera mobile_phone computer software game instrument weapon</p>	<p>art written_work newspaper music</p>						
<p>event attack election protest</p>	<p>military_conflict natural_disaster sports_event terrorist_attack</p>						
<p>building airport dam hospital hotel library power_station restaurant sports_facility theater</p>	<p>time color award educational_degree title law ethnicity language religion god</p>	<p>chemical_thing biological_thing medical_treatment disease symptom drug body_part living_thing animal food</p>	<p>website broadcast_network broadcast_program tv_channel currency stock_exchange algorithm programming_language transit_system transit_line</p>				

Fig. 7 Types in FIGER dataset

$$\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\} y_i \in \{1, 2, \dots, D\} \quad (18)$$

$$x = [r_e \ r_c \ r_g \ r_f] \quad (19)$$

Table 3 An example of dependency parser

Sentence	Dependency parser
The quick brown fox jumps over the lazy dog	(‘fox’, ‘NN’), ‘det’, (‘the’, ‘DT’) (‘fox’, ‘NN’), ‘amod’, (‘quick’, ‘JJ’) (‘fox’, ‘NN’), ‘amod’, (‘brown’, ‘JJ’) (‘fox’, ‘NN’), ‘dep’, (‘jumps’, ‘NNS’) (‘jumps’, ‘NNS’), ‘nmod’, (‘dog’, ‘NN’) (‘dog’, ‘NN’), ‘case’, (‘over’, ‘IN’) (‘dog’, ‘NN’), ‘det’, (‘the’, ‘DT’) (‘dog’, ‘NN’), ‘amod’, (‘lazy’, ‘JJ’)

Table 4 Dependency relations in this paper

Dependency	Explanation
Advmod	Aderb modifier
Dobj	Direct object
Nsubjpass	Passive nominal subject
Nsubj	Nominal subject

Table 5 Parameter settings 1

Parameter	Value
Embedding layer	1
Bidirectional layer	2
Attention layer	1
Dense layer	2
Softmax layer	1
Learning rate	0.001
Mini-batch	500
Loss function	Cross entropy loss function
Activation	ReLU
Dropout	0.5
Optimizer	Adam $\beta_1 = 0.9, \beta_2 = 0.999, \varepsilon = 1e - 8$

Table 6 Parameter settings 2

Description	Parameter	Value
Word embeddings	$D_e/D_c/D_g$	300
Hidden size of LSTM	D_h	100
Hidden layer size of attention model	D_a	100
Size of the projection of manually crafted features	D_l	50
Threshold	γ	0.5

where n is the number of input and D is the size of the set of label. Then we can compute the loss function L :

$$L = -\frac{1}{n} \left[\sum_{i=1}^n \sum_{j=1}^D 1\{y_i = d\} \log \frac{e^{W_{yd}x_i}}{\sum_{j=1}^D e^{W_{yj}x_i}} \right] \quad (20)$$

where $1\{y_i = d\}$ represents an indicator function. When $y_i = d$ is true, the result of the formula is 1, otherwise the result is 0.

In addition, the selection of optimizer is also an important task. This paper chooses the algorithm Adam (Graves 2012) which optimizes the algorithm SGD. Different parameters in Adam algorithm can adapt appropriate learning rate. It uses a momentum-like attenuation method.

Assuming that each parameter θ_i in the model uses the same learning rate η and gradient g_t of the objective function parameter θ_i :

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (21)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (22)$$

where β_1, β_2 are the decay rates, m_t represents weighted average variance, v_t represents weighted deviation variance. m_t and v_t are set to zero. However, they have always been close to zero during the process, especially when β_1 and β_2 are close to 1. To solve this problem, we have made a deviation correction to m_t and v_t :

$$m'_t = \frac{m_t}{1 - \beta_1^t} \quad (23)$$

$$v'_t = \frac{v_t}{1 - \beta_2^t} \quad (24)$$

The renewal equation is as follows:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{v'_t} + \varepsilon} m'_t \quad (25)$$

Kingma pointed out that Adam performed better in deviation correction (Kingma and Ba 2015), because it was much sparser in the convergence process.

Table 7 Results on FIGER

Models	Strict	Loose macro	Loose micro
FIGER	52.30	69.90	69.30
FIGER+PLE	59.90	76.30	74.90
K-WASABIE	–	–	72.35
Attentive+ Manually crafted	59.68	78.97	75.36
Ours	59.92	79.35	75.73

4.5 Evaluation criteria

The strict, loose macro, and loose micro are used in this section to evaluate performances. We denote the true set of types as T_i and the prediction set as T'_i . N is the number of instances. The three ways of computing P (precision)/ R (recall) are listed as follows:

(1) Strict

$$P = R = \frac{1}{N} \sum_{i=1}^N \delta(T'_i = T_i) \quad (26)$$

(2) Loose macro

$$P = \frac{1}{N} \sum_{i=1}^N \frac{|T'_i \cap T_i|}{|T'_i|} \quad (27)$$

$$R = \frac{1}{N} \sum_{i=1}^N \frac{|T'_i \cap T_i|}{|T_i|} \quad (28)$$

(3) Loose micro

$$P = \frac{\sum_{i=1}^N |T'_i \cap T_i|}{\sum_{i=1}^N |T'_i|} \quad (29)$$

$$R = \frac{\sum_{i=1}^N |T'_i \cap T_i|}{\sum_{i=1}^N |T_i|} \quad (30)$$

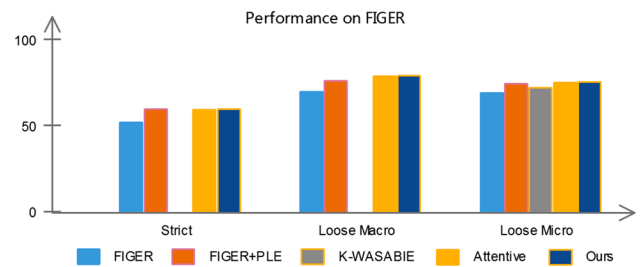
Then, we can compute the F score:

$$F = \frac{2 \times P \times R}{P + R} \quad (31)$$

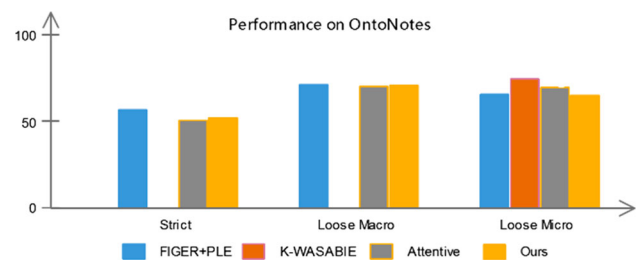
4.6 Experiment results

The experiment results of our model on the FIGER dataset are compared with the results of other baseline methods as shown in Table 7 and Fig. 8.

From Table 7 and Fig. 8, we observe that the results on FIGER are improved compared with the results on previous method mentioned in Chapter 2. Then, we can see the results of our model applied into OntoNotes which are shown in Table 8 and Fig. 9.

**Fig. 8** Results on FIGER**Table 8** Results on OntoNotes

Models	Strict	Loose macro	Loose micro
FIGER + PLE	57.20	71.50	66.10
K-WASABIE	–	–	70.01
Attentive+ Manually crafted	50.89	70.80	64.93
Ours	52.47	71.42	65.35

**Fig. 9** Results on OntoNotes

From Table 8 and Fig. 9, we can see that the results of our model on the OntoNotes are improved compared with the results of the model proposed by (Shimaoka et al. 2017). Combining with the results on FIGER, we can claim that the cause is due to the difference of datasets.

To sum up, experiment results verify the effectiveness of our model for fine-grained entity type classification.

5 Conclusions and future work

In this paper, existing type classification methods were analyzed, and we proposed a fine-grained type classification method with global information and sliding window context. We added dynamic global information on the basis of traditional type classification method. The global information is acquired by using the automatic summarization technique to remove the redundancy information. Then we fed it into neural networks to improve the accuracy of type prediction. In addition, different to the fixed window context-based methods in previous work, we proposed an adaptively window adjusting method which could locate the context information by finding the structure of the sentence structure. Finally, we

demonstrated in the experiment that the performance of our model is improved when compared with others. The strict, loose macro and loose micro of our model on OntoNotes reached 52.47, 71.42 and 65.35, respectively, which is better than FIGER+PLE, K-WASABIE and Attentive + Manually crafted by comprehensive comparison of the results.

In the future, there are several directions worth exploring. First, we hope to explore more entity types by distant supervision where open text can be used to complete the fine-grained type classification. Second, we would like to figure out whether our method can recognize more than one entity while they appear in the same one sentence.

Acknowledgements This work was supported by Shanghai Maritime University research fund project (20130469), Shanghai Science & Technology Innovation Plan Fund (14511107400), the State Oceanic Administration China research fund project (201305026), and by the open research fund of Key Lab of Broadband Wireless Communication and Sensor Network Technology (Nanjing University of Posts and Telecommunications), Ministry of Education. This work was also supported by the Industrial Core Technology Development Program (10049079, Development of Mining core technology exploiting personal big data) funded by the Ministry of Trade, Industry and Energy (MOTIE), Korea.

Compliance with ethical standards

Conflict of interest The authors declare that he has no conflict of interest.

Ethical approval This article does not contain any studies with human participants performed by any of the authors.

References

- Barua J, Patel D (2017) Named entity classification using search engine's query suggestions. In: European conference on information retrieval, pp 612–618
- Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. *J Mach Learn Res* 3(1):993–1022
- Brown PF, Desouza PV, Mercer RL, Pietra VJD, Lai JC (1992) Class-based n-gram models of natural language. *Comput Linguist* 18(4):467–479
- Corro LD, Abujabal A, Gemulla R, Weikum G (2015) FINET: context-aware fine-grained named entity typing. In: Conference on empirical methods in natural language processing, pp 868–878
- Cui KY, Ren PJ, Chen ZM, Lian T, Ma J (2017) Relation enhanced neural model for type classification of entity mentions with a fine-grained taxonomy. *J Comput Sci Technol* 32(4):814–827
- Dan G, Lazić N, Ganchev K, Kirchner J, Huynh D (2016) Context-dependent fine-grained entity type tagging. arXiv preprint [arXiv:1412.1820](https://arxiv.org/abs/1412.1820), 2014
- Dong L, Wei F, Sun H, Zhou M, Xu K (2015) A hybrid neural model for type classification of entity mentions. In: International conference on artificial intelligence, pp 1243–1249
- Fleischman M, Hovy E (2002) Fine grained classification of named entities. In: Proceeding of the 19th international conference on computational linguistics, pp 1–7
- Giuliano C, Gliozzo A (2008) Instance-based ontology population exploiting named-entity substitution, COLING 2008. In: Proceeding of the 22th international conference on computational linguistics, Manchester, UK, pp 265–272
- Gotti F, Langlais P (2016) Harnessing open information extraction for entity classification in a French corpus. In: Canadian conference on artificial intelligence, pp 150–161
- Graves A (2012) Supervised sequence labelling. In: Supervised sequence labelling with recurrent neural networks. Studies in computational intelligence, vol 385. Springer, Berlin, Heidelberg
- Hoffart J, Suchanek FM, Berberich K, Weikum G (2013) Yago2: a spatially and temporally enhanced knowledge base from wikipedia. *Artif Intell* 194(1):28–61
- Kingma DP, Ba J (2015) Adam: a method for stochastic optimization. In: The 3rd international conference for learning representations, San Diego. [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)
- Kleinberg JM (1999) Authoritative sources in a hyperlinked environment. *J ACM* 46(5):604–632
- Krupka GR, Hausman K (1998) Isoquest inc.: description of the netowltm extractor system as used for muc-7. In: Proceedings of MUC, vol 7
- Lee C, Hwang YG, Oh HJ, Lim S, Heo J, Lee CH et al (2006) Fine-grained named entity recognition using conditional random fields for question answering. In: Proceeding of third Asia information retrieval symposium, pp 581–587
- Mintz M, Bills S, Snow R, Dan J (2009) Distant supervision for relation extraction without labeled data. In: Proceedings of the joint conference of the 47th annual meeting of the ACL and the 4th international joint conference on natural language processing of the AFNLP, vol 2, pp 1003–1011
- Pennington J, Socher R, Manning C (2014) Glove: global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1532–1543
- Rahman A, Ng V (2010) Inducing fine-grained semantic classes via hierarchical and collective classification. In: Proceedings of the 23rd international conference on computational linguistics, pp 931–939
- Ren X, He W, Qu M, Voss CR, Ji H, Han J (2016) Label noise reduction in entity typing by heterogeneous partial-label embedding: [arXiv:1602.05307](https://arxiv.org/abs/1602.05307)
- Sang EFTK, Meulder FD (2003) Introduction to the conll-2003 shared task: language-independent named entity recognition. In: Proceedings of the seventh conference on natural language learning at HLT-NAACL 2003, vol 4, pp 142–147
- Sekine S (2009) Extended named entity ontology with attribute information. In: Proceeding of international conference on language resources and evaluation, Lrec 2008, 26 May–1 June 2008, Marrakech, Morocco, vol 24, pp 52–57
- Shimaoka S, Stenetorp P, Inui K, Riedel S (2016) An attentive neural architecture for fine-grained entity type classification. arXiv preprint [arXiv:1604.05525](https://arxiv.org/abs/1604.05525)
- Shimaoka S, Stenetorp P, Inui K, Riedel S (2017) Neural architectures for fine-grained entity type classification. arXiv preprint [arXiv:1606.01341](https://arxiv.org/abs/1606.01341)
- Singh PK, Kumar C (2015) A note on computing the crisp order context of a fuzzy formal context for knowledge reduction. *J Inf Process Syst* 11(2):184–204
- Vijayarajan V, Dinakaran M, Tejaswin P, Lohani M (2016) A generic framework for ontology-based information retrieval and image retrieval in web data. *Human Centric Comput Inf Sci* 6(1):18
- Viswanathan V, Krishnamurthi I (2012) Finding relevant semantic association paths through user-specific intermediate entities. *Human Centric Comput Inf Sci* 2(1):1–11

- Xiao L, Weld DS (2012a) Entity linking with a knowledge base: issues, techniques, and solutions. *IEEE Trans Knowl Data Eng* 27(2):443–460
- Xiao L, Weld DS (2012b) Fine-grained entity recognition. In: *Proceedings of the 26th conference on artificial intelligence (AAAI)*
- Yogatama D, Gillick D, Lazić N (2015) Embedding methods for fine-grained entity type classification. In: *Meeting of the association for computational linguistics and the international joint conference on natural language processing*, pp 291–296
- Yosef MA, Bauer S, Hoffart J, Spaniol M, Weikum G (2015) HYENA: hierarchical type classification for entity names. In: *Processing of COLING 2012: posters*, pp 1361–1370