# Entity Linking for Chinese Short Texts Based on BERT and Entity Name Embeddings

Jingwei Cheng, Chunguang Pan, Jinming Dang, Zhi Yang, Xuyang Guo, Linlin Zhang, and Fu Zhang

Northeastern University
chengjingwei@mail.neu.edu.cn

**Abstract.** Traditional entity linking tasks are primarily for long texts, which have full context information that can assist in the recognition and disambiguation of entities. In contrast, Chinese short text entity linking brings a great challenge due to serious colloquialism and insufficient contexts. The whole process of entity linking consists of two subtasks: entity recognition and entity disambiguation. For the entity recognition subtask, we use description text information in a knowledge base to enhance the performance of entity recognition, and propose the BERT-EntityNameEmbedding (BERT-ENE) model. Specifically, the vector embedding of a entity is first obtained by mining the description text of the entity in a knowledge base. Then, candidate entities in the short text are obtained by name dictionary matching technique. Finally, the results are filtered by the BERT-ENE model to fulfil the task of entity recognition. In addition, we further propose a combination of the BERT-ENE model and the BERT-CRF model, which significantly improves the recognition effect compared with traditional methods. For the entity disambiguation subtask, we regard it as a binary classificaiton problem. Candidate entities are predicted by a BERT-based binary classification model, and then predicted probabilities are sorted. The entity with highest probability is selected as the correct entity. Based on the proposed method in this paper, we achieved the first place in the CCKS2019 entity linking task for Chinese short text.

**Keywords:** Entity Linking · Entity Recognition · Entity Disambiguation · BERT

## 1 Introduction

In recent years, with the rapid development of the Internet, a large amount of Web data has been generated, and Web has become one of the largest databases in the world. The data on the Web is mostly presented in the form of Web texts, which contain a large number of named entities [18] (for example, people, places, and organizations), which are the basic elements of Web text. However, the meanings of these entities are very vague, especially in the case when named entities are frequently appearing. A named entity may have multiple names, and a single name may also represents several different named entities. In addition,

the emergence of knowledge sharing communities, such as Wikipedia, and the rapid development of information extraction technology have facilitated the automatic construction of large-scale knowledge bases. A knowledge base contains entities, information about entities, and information between entities. Automatic construction of a knowledge base involves extracting the relationship between entities from Web texts and adding them to the knowledge base. In this process, the ambiguities of extracted entities need to be eliminated. This process is called *Entity Linking*. An entity linking task maps an recognized entity reference to the correct entity object in an existing knowledge base. The whole process of entity linking includes two subtasks: entity recognition and entity disambiguation.

The method proposed in this paper is meant to solve CCKS2019 Challenge task 2: entity linking for Chinese short text. The training data, test data, and the knowledge base used in this task are provided by Baidu. The length for short texts is 50 characters at most and 27 characters on average. The data samples are shown in Table 1. In the first example, common words such as ' 人物' and '性格' also serve as mentions. In the second example, the sentence of 16 characters has 5 entities, among which is a single-character entity '诗'. From this we can have some idea of the difficulty of this task. Compared to longer texts and English texts, there are great challenges for entity linking for Chinese short texts. The main reasons are as follows: (1) Serious colloquialism leads to difficulties in entity recognition and entity ambiguity resolution. (2) Short texts have no sufficient contexts and require a precise understanding of contexts. (3) Compared with English, Chinese is more challenging in the short text link problem due to the characteristics of the language itself.

**Table 1.** Data examples

| text | mention |
|---|---|
| 求一些亦正亦邪的人物的性格描写 | 人物, 性格 |
| 《有所思》_萧衍的诗_南北朝诗人 | 有所思, 萧衍, 诗, 南北朝, 诗人 |

In this paper, we divide the entity linking task into two subtasks. For the entity recognition subtask, BERT-CRF [1] is a popular named entity recognition model. However, the BERT-CRF model can only use short text information and does not utilize the information of the knowledge base. Therefore, there will still be problems such as entity boundary recognition errors and incomplete recognition of entities. To amend these deficiencies and to fully utilize the information of the knowledge base, we propose the BERT-ENE model. For the subtask of entity disambiguation, we regard it as a binary classification problem, and use the BERT-based binary classification model to disambiguate candidate entities.

The main innovations of this paper are as follows:

1. The pre-training model BERT is skillfully deployed in the entity recognition and entity disambiguation of short texts, and the semantic information of short texts is fully extracted.

2. In the process of entity recognition, by introducing the vector embeddings of entity names, the text description information of the knowledge base is fully utilized, which in turn solves the problem that short texts always carry too little information.

3. A new model that combines the BERT-ENE model and the BERT-CRF model is proposed, which significantly improves the effectiveness of entity recognition.

## 2 Related works

We mainly use named entity recognition and name dictionary based matching technology in entity recognition subtask. In entity disambiguation subtask, we adopt the binary classification idea to achieve disambiguation. The related works of entity recogniton and entity disambiguation are described separately as follows.

In its early history, the research lines of named entity recognition can be classified as rule-based methods [11] [13] [7], statistical methods [8] [16] [19] [14] and a combination of rule-based and statistical methods [15] [20] [13]. Rule-based methods typically rely on a specific language environment, and once the corpus changes, the rules in the system must be reconstructed. In contrast, statistical methods do not require extensive linguistic knowledge and train language models through machine learning methods to automatically identify named entities. With the rapid development of neural networks, end-to-end solutions that do not rely on artificial features gradually dominate the mainstream. Considering the sequence modeling capabilities of LSTM, Lample et al. [9] proposed LSTM-CRF, which becomes one of the most popular infrastructures for named entity recognition. Later, additional network layers or features are introduced into LSTM-CRF to improve performance. For instance, Ma and Hovy [10] added convolution layers, and Ghaddar and Langlais [3] added additional vocabulary features. In the past two years, the emergence of pre-trained language models, such as ELMo [10] and BERT [1], further improved the performance of named entity recognition. The BERT-CRF model has also become a popular entity recognition model.

In view that named entity recognition strategy does not fully identify all candidate entities in the text, researchers use name dictionary-based matching techniques to improve performance, which maintain a name dictionary extracted from a knowledge base [5] [4]. Each name is a keyword in the dictionary and has a set of possible entities mapping to it. After building such a dictionary, we can use different methods to obtain candidate entities, many of which adopt a precise matching strategy. To improve the recall, Zheng et al. [21] used loose matching instead of precision matching, by means of string matching rules. To improve the precision, some studies use empirical probabilities to select candidate entities

[12]. Most existing methods for selecting matching results are based on rules or probabilities and do not take advantage of deep learning models.

At present, there are mainly three kinds of entity disambiguation methods, i.e., rank based methods, binary classification based methods, and graph model based methods. Next, we will focus on binary classification methods which are directly related to our work. In a binary classification based method, the correlation features of mentions and candidate entities are typically used to train the binary classification model, which is capable of determining if a candidate entity is a positive instance. For example, Pan et al. [12] extract features including vocabulary features, word categories, and named entity categories, and then uses SVM classifiers for classification. Traditional machine learning methods rely too much on manual constructed features. The quality of features seriously affects the performance of classifiers. Deep learning based methods automatically learn representations of features. For example, Sun et al. [17] proposed to use a deep learning approach to obtain semantic representations of mentions, contexts, and entities. Huang et al. [6] proposed a deep semantic association model based on deep neural network to measure entity semantic association. Ganea et al. [2] achieved entity disambiguation in virtue of entity embeddings and local context window attention mechanism.

## 3 Model

We will detail our model from three steps: data preprocessing, entity recognition, and entity disambiguation.

### 3.1 Data Preprocessing

The training data includes a `text` field and a `ment_data` field. The `ment_data` contains `mention` and `kb_id` fields. The knowledge base contains fields of `subject_id`, `subject`, `alias`, `data`, etc. The `data` fields contains multiple `predicate` and `object` fields.

*Introducing a new alias* After statistical analysis of the data set, we found 2.592% entity names in the training set have no matching in the entity library. Some errors are shown below:

1. 安妮‘海瑟薇: There is a special character with the text.
2. 新浪微薄: The entity name in the input text is incorrect.
3. 国家质检总局: The alias is not in the knowledge base.

In order to solve these problems, we introduce a new alias for the corresponding entity in the knowledge base. The steps are as follows:

1. For error 1, normalize special characters and add the processed name to the alias of the corresponding entity. For example, all Chinese punctuation marks are replaced with English punctuation marks.

2. For error 2 and error 3, count the number of times $E_{num}$ that an entity $E$ has no matching, all strings $M_1, M_2, M_3...M_i$ in the training set that entity

$E$ fails in matching, and the number of occurrences $M_{i_{num}}$ of each $M_i$. We set that the string $M_i$ is added to the alias of entity $E$, if $E_{num}$ is greater than 4 and $M_{i_{num}}$ is greater than 3.

*Construction of entity description text* Concatenate the `predicate` and `object` in `data` to obtain the entity description text. In order to facilitate later processing, we truncate the text according to the following rule: if the total length of the `predicate` and `object` is greater than 30, truncate proportionally, and no truncation otherwise.

*Name dictionary construction* The name dictionary is constructed according to entity names, entity aliases, lowercase of entity names, and the newly introduced aliases above. Each entity name corresponds to one or more entity ids after construction. For example, 'victory': ['10001', '19044', '37234', '38870', '40008', '85426', '86532', '140750'].

### 3.2 Entity recognition

**BERT-CRF** Figure 1 shows the BERT-CRF model with BIO tags, where the [CLS] and [SEP] locations of the BERT are represented by the tag TAG. The model consists of an input layer, a BERT layer, and a CRF layer.

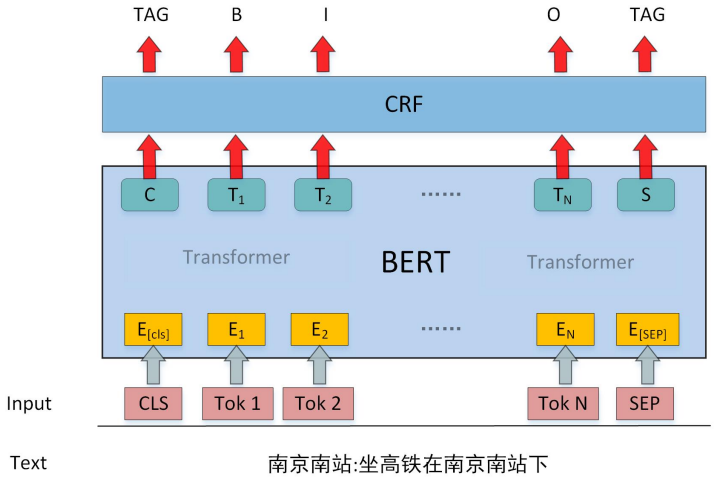


**Fig. 1.** BERT-CRF model

Input layer: The input to the BERT is the sum of word embedding, position embedding, and type embedding. The word embedding is the corresponding id of each word, and the type embedding is 0 or 1, where 0 denotes the first sentence and 1 the second sentence. For named entity recognition tasks, there is

only one sentence and the type embedding is always 0. To learn time sequential features, BERT uses positional embeddings to add time sequential information. The position embedding is shown in Equation (1) and (2):

$$PE(pos, 2i) = \sin\left(pos/10000^{2i/d \bmod d}\right) \tag{1}$$

$$PE(pos, 2i+i) = \cos\left(pos/10000^{2i/d \bmod d}\right) \tag{2}$$

BERT layer: The BERT layer contains a 12-layer Transformer encoder structure. The most important module of an encoder unit is the self-attention part, as shown in Equation (3).

$$(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3}$$

where $Q, K, V$ are the input word vector matrix and $d_k$ is the input vector dimension. The core is self-attention. In order to extend the model's ability to focus on different locations and increase the "representative subspace" of the attention unit, Transformer uses "multi-head" mode, as shown in Equation (4) and (5).

$$MultiHead(Q, K, V) = Concat(head_1, ...., head_h)W^o \tag{4}$$

$$head_i = Attention\left(QW_i^Q, KW_i^k, VW_i^V\right) \tag{5}$$

In addition, to solve the degradation problem in deep learning, the residual network and layer normalization are added to the Transformer encoder unit, as shown in Equation (6) and (7):

$$LN\left(x_i\right) = \alpha \times \frac{x_i - \mu_L}{\sqrt{\sigma_L^2 + \varepsilon}} + \beta \tag{6}$$

$$FFN = \max\left(0, xW_1 + b_1\right)W_2 + b_2 \tag{7}$$

CRF layer: The above BERT layer takes into account long-term context information, and does not consider the dependencies between tags. CRF is adopted to model the tag sequence, as it obtains a global optimal tag sequence by considering the adjacency relationship between tags.

### 3.3 BERT-EntityNameEmbedding（BERT-ENE）model

The BERT-ENE model is shown in Figure 2. The specific ideas are as follows: 1. Construct an entity name dictionary by using entity names of the knowledge base and the alias information of entities. 2. Using the BERT pre-training model and entity description text to select the vector output in the [CLS] position as the embedding of the entity name. 3. Obtain candidate entities in the short text by dictionary matching. 4. Filter matching results with the BERT-ENE model.

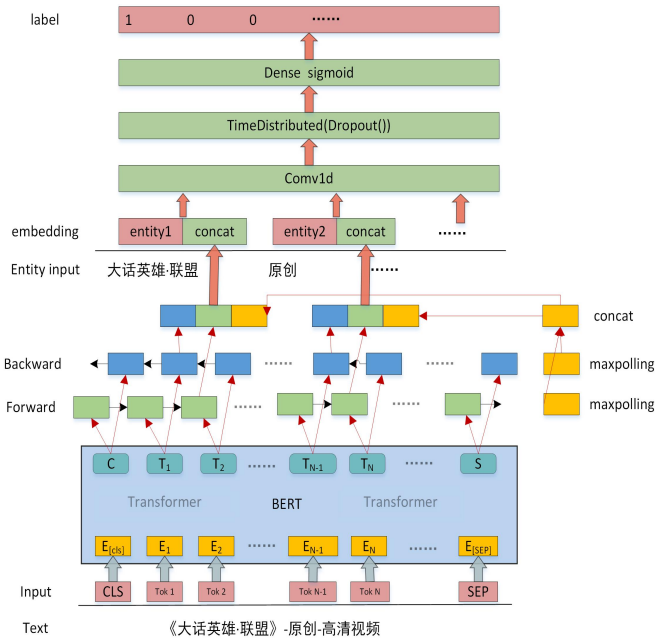


**Fig. 2.** BERT-ENE model

*Dictionary Matching* Use the constructed entity name dictionary and adopt the idea of maximum precision matching to match entities in the text. In order to improve the matching precision, leave single character entity names such as ‘ 诗’ in the name dictionary unmatched.

*Entity name embedding* Embeddings of entity names are obtained through the BERT model. The specific ideas and processes are as follows:

1. Each entity text description is input into the BERT model, and the output vector at the [CLS] position is extracted to express the meaning of this entity. We thus obtain the vector representation of each entity.

2. In case the entity name corresponds to only one entity, the entity corresponding vector is directly used as the embedding of the entity name, such as ‘无尽武道’: [’10007’].

3. In case where the entity name corresponds to multiple entities, use the average of vectors, such as ‘胜利’: [’10001’, ’19044’, ’37234’, ’38870’, ’40008’, ’85426’, ’86532’, ’ 140750’].

In this way, we obtain a 768-dimension embedding for each entity name.

*BERT-ENE model* The input of the BERT-ENE model has two parts, the short text and the entity name embedding. As that of BERT, the short text input also has three parts: word embedding, position embedding and type embedding. The short text input layer is followed by a BERT layer and a GRU layer.

GRU is a variant of the RNN network, and the specific calculation process is as shown in Equation (8)–(11):

$$z_t = \sigma\left(W_i * [h_{t-1}, x_t]\right) \tag{8}$$

$$r_t = \sigma\left(W_r * [h_{t-1}, x_t]\right) \tag{9}$$

$$\widetilde{h}_t = \tanh\left(W_c * [r_t \bullet h_{t-1}, x_t]\right) \tag{10}$$

$$h_t = (1 - z_t) \bullet c_{t-1} + z_t \bullet \tilde{h}_t \tag{11}$$

where $\sigma$ is the sigmoid function and $\bullet$ is the dot product. $x_t$ is the input vector of time $t$, $h_t$ is the hidden state and also the output vector, which contains all the information before time $t$. $Z_t$ is the update gate, which determines whether the information flows into the next moment, and $r_t$ is a reset gate to determine whether the information is to be lost, and the two jointly determine the output of the hidden state.

To take full advantage of contextual information, a two-way GRU is used here. The BERT output is firstly input into the forward GRU network and the backward GRU network, respectively. Then the matching entity name corresponding to the vector $V_{end}$ at the end position of the forward GRU, and the vector $V_{begin}$ corresponding to the start position of the backward GRU are extracted and concatenated to obtain $V_{con}$ , which servers as the semantic representation of the entity name. To learn the information of the whole text, the maximum pooling operation is performed on the output of the forward GRU and the backward GRU respectively to obtain the vector $V_{max}$, which can represent the semantics of the whole text. Finally, $V_{max}$ is concatenated with $V_{con}$ and the corresponding embedded entity name, then through the convolutional layer, the fully connected layer, and the sigmoid function activation, the predicted probability is obtained.

The BERT-ENE model is essentially a binary categorization model, which is designed to filter the matching entities. The loss function is defined in Equation (12).

$$loss = -\sum_{i=1}^{n} \hat{y}_i \log y_i + (1 - \hat{y}_i) \log\left(1 - \hat{y}_i\right) \tag{12}$$

### 3.4 Result fusion

As mentioned above, we use two models in entity recognition, a BERT-CRF model and a BERT-ENE model. Entities recognized by the BERT-CRF model may not match candidate entities due to boundary errors. The BERT-ENE model uses dictionary matching, each result of BERT-ENE thus can find candidate entities in the knowledge base and is immunized to boundary errors. The BERT-ENE model removes single character entities in dictionary matching, whereas the BERT-CRF model can predicts single character entities. The two solutions are combined to achieve better results. The fusion rule is that if there is a duplicate

of the two results at the same position, the result of BERT-ENE is selected, and the result of the BERT-CRF model is selected if there is a single character entity.

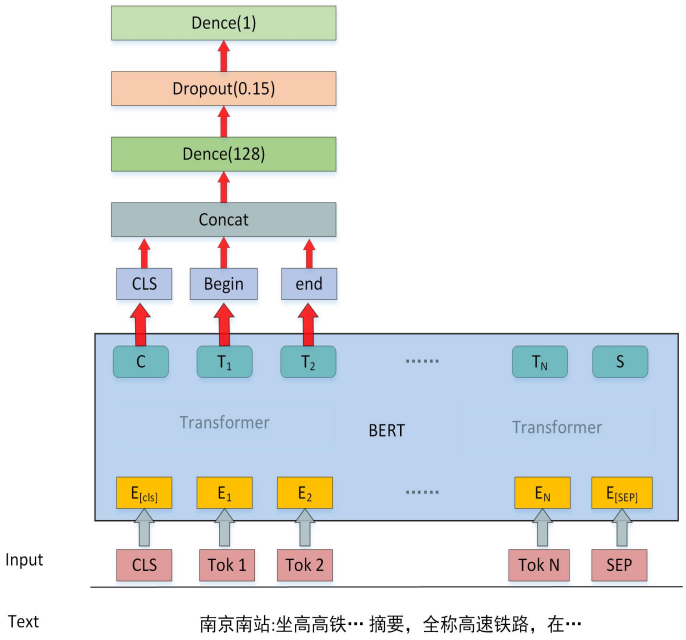


**Fig. 3.** BERT-binary model

### 3.5 Entity Disambiguation

Entity disambiguation is based on the idea of binary classification. When training, the matched entity is selected as a positive example, and two negative examples are selected in the candidate entities. The short text and the description text of the entity to be disambiguated are concatenated and input into the BERT model. The vector output at the [CLS] position is concatenated with the feature vector of the start and end positions of the candidate entity, then through the fully connected layer, the sigmoid function activation, the probabilities of candidate entities are obtained. The entity with highest probability is selected as the correct entity. The BERT-binary model is shown in Figure 3.

As in BERT, the input consists of word embedding, position embedding, and type embedding. The binary classification of two sentence demands the type embedding should have two values, where the embedding of first sentence is 0 and the embedding of the second sentence is 1.

**Table 2.** Entity Recognition Results Table

| model | Precision | Recall | F1-Measure |
|---|---|---|---|
| LSTM-CRF | 0.7862 | 0.7834 | 0.7848 |
| BERT-CRF | 0.8316 | 0.8121 | 0.8218 |
| BERT-ENE | 0.8224 | 0.8157 | 0.8191 |
| BERT-CRF & BERT-ENE | **0.8268** | **0.8534** | **0.8398** |

**Table 3.** Entity disambiguation results table

| model | Precision | Recall | F1-Measure |
|---|---|---|---|
| LSTM | 0.8657 | 0.8587 | 0.8621 |
| LSTM-CapsNet | 0.8716 | 0.8634 | 0.8674 |
| BERT-Binary | **0.8955** | **0.9019** | **0.8987** |

For the binary classification task, the loss function is shown in Equation (13):

$$loss = -\sum_{i=1}^{n} \hat{y}_i \log y_i + (1 - \hat{y}_i) \log (1 - \hat{y}_i) \tag{13}$$

## 4 Experiments

During the off-line model training, 90,000 training instances are divided into 80,000 training instances and 10,000 verification instances. We divide the entity linking task into two subtasks: entity identification and entity disambiguation. For each subtask, a number of different models are tried, and finally the best performing models are selected on the two subtasks. The results and process of the experiment are as follows:

The classic LSTM-CRF was selected as the baseline model in the entity identification stage. The experimental results are shown in Table 2. It can be seen that both the BERT-CRF and BERT-ENE models have a significant improvement over the LSTM-CRF model. The fusion of BERT-CRF and BERT-ENE improves 1.8 percentages compared to the BERT-CRF and 2.07 percentages compared to the BERT-ENE model.

The result of the disambiguation experiment is carried out in case all entities are correct. For comparison, experiments are also performed on models such as LSTM and LSTM-CapsNet. Table 3 shows a comparison of results of different disambiguation models.

After the above different attempts, we finally selected BERT-CRF & BERT-ENE & BERT-binary as our final solution. Results of experiments on the verification set and on the final test set are shown in Table 4.

**Table 4.** Entity disambiguation results table

| dataset | Precision | Recall | F1-Measure |
|---|---|---|---|
| dev | **0.7563** | **0.8020** | **0.7785** |
| test | **0.7868** | **0.8167** | **0.8014** |

## 5 Conclusion

In this paper, we provide a complete solution for entity linking of Chinese short texts, and propose a BER-ENE model to introduce vector embedding of entity names, making full use of the text description information of the knowledge base to solve the problem that the short texts carry too little information. Experiments show that the proposed method is effective, and our model ranks first in the fierce competition of CCKS2019 Challenge Task 2. However, there are still challenges in the short text entity linking task. In the following work, we will continue to try new models and new methods to improve the performance of short-text entity linking.

## Acknowledgements

The work is supported by the National Natural Science Foundation of China (61672139) and Project No. JCKY2018205C012.

## References

1. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
2. Ganea, O.E., Hofmann, T.: Deep joint entity disambiguation with local neural attention. arXiv preprint arXiv:1704.04920 (2017)
3. Ghaddar, A., Langlais, P.: Robust lexical features for improved neural network named-entity recognition. arXiv preprint arXiv:1806.03489 (2018)
4. Guo, Z., Barbosa, D.: Robust entity linking via random walks. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management. pp. 499–508. ACM (2014)
5. Han, X., Sun, L., Zhao, J.: Collective entity linking in web text: a graph-based method. In: Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval. pp. 765–774. ACM (2011)
6. Huang, H., Heck, L., Ji, H.: Leveraging deep neural networks and knowledge graphs for entity disambiguation. arXiv preprint arXiv:1504.07678 (2015)
7. Humphreys, K., Gaizauskas, R., Azzam, S., Huyck, C., Mitchell, B., Cunningham, H., Wilks, Y.: University of sheffield: Description of the lasie-ii system as used for muc-7. In: Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29-May 1, 1998 (1998)

8. Isozaki, H., Kazawa, H.: Efficient support vector classifiers for named entity recognition. In: Proceedings of the 19th international conference on Computational linguistics-Volume 1. pp. 1–7. Association for Computational Linguistics (2002)
9. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. arXiv preprint arXiv:1603.01360 (2016)
10. Ma, X., Hovy, E.: End-to-end sequence labeling via bi-directional lstm-cnns-crf. arXiv preprint arXiv:1603.01354 (2016)
11. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Lingvisticae Investigationes **30**(1), 3–26 (2007)
12. Pan, X., Cassidy, T., Hermjakob, U., Ji, H., Knight, K.: Unsupervised entity linking with abstract meaning representation. In: Proceedings of the 2015 conference of the north american chapter of the association for computational linguistics: Human language technologies. pp. 1130–1139 (2015)
13. Petasis, G., Vichot, F., Wolinski, F., Paliouras, G., Karkaletsis, V., Spyropoulos, C.D.: Using machine learning to maintain rule-based named-entity recognition and classification systems. In: Proceedings of the 39th Annual Meeting on Association for Computational Linguistics. pp. 426–433. Association for Computational Linguistics (2001)
14. Saha, S., Ekbal, A.: Combining multiple classifiers using vote based classifier ensemble technique for named entity recognition. Data & Knowledge Engineering **85**, 15–39 (2013)
15. Shaalan, K., Oudah, M.: A hybrid approach to arabic named entity recognition. Journal of Information Science **40**(1), 67–87 (2014)
16. Speck, R., Ngomo, A.C.N.: Ensemble learning for named entity recognition. In: International semantic web conference. pp. 519–534. Springer (2014)
17. Sun, Y., Lin, L., Tang, D., Yang, N., Ji, Z., Wang, X.: Modeling mention, context and entity with neural networks for entity disambiguation. In: Twenty-Fourth International Joint Conference on Artificial Intelligence (2015)
18. Sundheim, B.M.: Named entity task definition, version 2.1. In: Proc. Sixth Message Understanding Conf.(MUC-6), Nov. 1995. pp. 317–332 (1995)
19. Tang, Z., Jiang, L., Yang, L., Li, K., Li, K.: Crfs based parallel biomedical named entity recognition algorithm employing mapreduce framework. Cluster Computing **18**(2), 493–505 (2015)
20. Wu, Y., Zhao, J., Xu, B.: Chinese named entity recognition combining a statistical model with human knowledge. In: Proceedings of the ACL 2003 workshop on Multilingual and mixed-language named entity recognition-Volume 15. pp. 65–72. Association for Computational Linguistics (2003)
21. Zheng, J.G., Howsmon, D., Zhang, B., Hahn, J., McGuinness, D., Hendler, J., Ji, H.: Entity linking for biomedical literature. BMC medical informatics and decision making **15**(1), S4 (2015)