Iterative Strategy for Named Entity Recognition with Imperfect Annotations

Huimin Xu, Yunian Chen, Jian Sun, Xuezhi Cao, and Rui Xie(✉)

Meituan-Dianping Group, Shanghai, China
{xuhuimin04,chenyunian,sunjian20,caoxuezhi,rui.xie}@meituan.com

Abstract. Named entity recognition (NER) systems have been widely researched and applied for decades. Most NER systems rely on high quality annotations, but in some specific domains, annotated data is usually imperfect, typically including incomplete annotations and non-annotations. Although related studies have achieved good results on specific types of annotations, to build a more robust NER system, it is necessary to consider complex scenarios that simultaneously contain complete annotations, incomplete annotations, non-annotations, etc. In this paper, we propose a novel NER system, which could use different strategies to process different types of annotations, rather than simply adopts the same strategy. Specifically, we perform multiple iterations. In each iteration, we first train the model based on incomplete annotations, and then use the model to re-annotate imperfect annotations and update their weights, which could generate and filter out high quality annotations. In addition, we fine-tune models through high quality annotations and its augmentations, and finally integrate multiple models to generate reliable prediction results. Comprehensive experiments are conducted to demonstrate the effectiveness of our system. Moreover, the system is ranked first and second respectively in two leaderboards of NLPCC 2020 Shared Task: Auto Information Extraction (https://github.com/ZhuiyiTechnology/AutoIE).

Keywords: NER · Imperfect annotations · Iterative strategy

1 Introduction

NER is one of the most important tasks in natural language processing (NLP). NER systems can identify named entities like person, TV, location, organization, etc. in texts, which can be applied to other NLP tasks, including information extraction, question answering, information retrieval, etc. Most NER algorithms focus on supervised learning approaches, which rely on high quality annotated corpus. However, high quality annotated data with ground-truth is usually difficult to obtain for some specific domains due to their complexities, such as word sense disambiguation, grammatical, professional word, or even typos.

In a real business scenario, There may be complete annotations and incomplete annotations, and non-annotations in a corpus. We refer to the latter two

X. Zhu et al. (Eds.): NLPCC 2020, LNAI 12431, pp. 512–523, 2020.
https://doi.org/10.1007/978-3-030-60457-8_42

types of annotations as imperfect annotations in this paper. **Complete annotations** represent the sequences that are verified and labeled completely correct. **Incomplete annotations** represent that sequences are labeled, but there may be missing or error caused by manual annotation or supervision. **Non-annotations** are sequences without any labels, which may be newly generated and not annotated yet, or really have no entities. Figure 1 shows an example sequence with three annotations.

	雪	山	飞	狐	金	庸	武	侠	剧
Complete annotations	B-TV	I-TV	I-TV	E-TV	B-PER	E-PER	O	O	O
Incomplete annotations	O	O	O	O	B-PER	E-PER	O	O	O
Non-annotations	O	O	O	O	O	O	O	O	O

Fig. 1. Examples of different annotations.

There is a lot of literature studying these annotations. For complete annotations, previous works focus on feature-engineered supervised systems and feature-inferring neural network systems [29]. The former systems focus on extracting features that have a good ability to distinguish entities [17,20,21], while the latter systems can automatically infer useful features for entity classification by using deep learning models [3,12,18]. For incomplete annotations, some works focus on modifying the model structure to learn from inexpensive partially annotated sequences [7,19], while the other work focuses on using iterative training strategy to relabel entities and update their weights, to improve weights of the high quality labeled entities and reduce weights of the unlabeled or mislabeled entities in a sequence [13]. For non-annotations, previous work focus on rule-based systems and unsupervised systems [16]. The former systems rely on lexicon resources and domain-specific knowledge [9,14], while the latter systems use lexical resources, lexical patterns, and statistics computed on a large corpus to infer mentions of named entities [4,31]. Although these works have achieved satisfactory results for a specific type of annotations, to our best knowledge, few papers have taken into account the differences between different types of annotations.

Different annotations cannot be simply processed by the same strategy. Using complete annotations can help NER algorithms quickly learn a high available model, while identifying and using incomplete annotations and non-annotations could help NER algorithms improve fault tolerance and cover more entity types, thereby improving the generalization of the algorithm. In order to work better in a real business environment, it is necessary to be able to process various annotations flexibly in a NER task. Thus, how to build a flexible and high accurate NER system based on various and complex annotations is the focus of this paper.

In this paper, we use iterative strategy to build robust models and propose flexible and efficient strategies to deal with different types of annotations.

We first use complete and incomplete annotations to train a base model. During the training process of each iteration, the base model will be used to relabel incomplete annotations and non-annotations, and then we could generate and filter out high confidence annotations for the next iteration. In addition, we use high quality annotations and its augmentations to fine-tune the base model to achieve higher performance. Finally, with ensembles of different models, we could build a more reliable system.

Comprehensive experiments are conducted to demonstrate the effectiveness of our system. We evaluate and verify the system on a complex corpus released by NLPCC 2020 Shared Task: Auto Information Extraction. The experimental results show that our system can effectively deal with different types of annotations and won first and second place respectively in two leaderboards of the NLPCC 2020 competition.

The rest of this paper is organized as follows: Sect. 2 introduces the related work. We describe our algorithm in Sect. 3. Sect. 4 shows the experimental results and analysis. Finally, we conclude this paper in Sect. 5.

2 Related Work

For NER task, HMM [23], MEMN [2] and CRF [15] are some traditional methods. Recently, neural network based embedding layers and conditional random fields (CRF) are often used in end-to-end model. Embedding layer can extract features of sentences. For example, word2vec [22], ELMo [26], BERT [6], Bidirectional LSTM (BiLSTM) and convolutional neural network (CNN) based models are used to obtain character-level or word-level representations. CRF often in the last layer of a model, can learn label constraints, such as tag "E" appears after tag "B" in "BIOE" annotation system.

Many researchers study the NER task with fully annotated data, however, obtaining a fully annotated dataset is expensive. Most of data is incomplete. The entity is not correctly labeled, but wrongly labeled as "O" which will disturb the training process. Some previous works [1,7] try to make assumptions on the data with "O" labels. However, there also are partly annotated entities or words with "O" labels in their assumptions which is unrealistic. Thus, Jie etc. [13] propose to regard the missing labels as latent variables and using classifier stacking technique to model them. Latent-variable CRF is also utilized in Chinese NER which is explored by Yang etc. [30] and in a biomedical NER task by Greenberg etc. [8].

Distant supervision is also a popular method in an incomplete annotation scenario, which can generate amounts of labeled data for new entities automatically. It assumes that if a string appears in a predefined entity dictionary, the string is likely to be an entity. Yang etc. [30] propose a distantly supervised approach to address both incomplete annotation problem and noisy annotation problem. Peng etc. [25] formulate the NER task with only unlabeled data and named entity dictionaries as a positive-unlabeled (PU) learning problem. Their model is also distantly supervised.

Other models like large margin learning framework [1], a modified structured perceptron framework [7,19] and CrossWeight [28] try to solve incomplete annotation problem from model structure aspect or data cleaning aspect. Some works [24,27] also study weakly supervised methods, but these methods usually perform worse on specific language or it's difficult to implement in a real-world scenario.

In addition, to combining multiple advantages in these works, we also consider some other aspects which can make our model perform better. Firstly, we design a more robust base model and propose an effective iterative strategy on an extremely incomplete dataset (only 30% entity labels appear in training data); secondly, we propose a data augmentation method to automatically generate more samples; Finally, we obtain more reliable prediction by integrating multiple model results.

3 Approach

We propose a novel and scalable system to deal with different types of annotations flexibly according to the characteristics of data. The system consists of three main modules, i.e., base model, iterative strategy and data augmentation, as shown in Fig. 2. Base model is a classic NER framework, including word representation layer, contextual embedding layer and output layer. Then, we propose an iterative strategy to reconstruct imperfect annotations. Finally, a specific data augmentation method is used to expand high quality annotated corpus. Next, we give a detailed description.

3.1 Base Model

Word Representation Layer. Given a word sequence $x = \{x_1, x_2, \cdots, x_t\}$ whose label sequence is $y = \{y_1, y_2, \cdots, y_t\}$, $y_i \epsilon [B, I, E, O]$. First, we map each word in the sequence to a high-dimensional vector space. Because the pre-trained language model (e.g., Bidirectional Encoder Representations from Transformers, BERT [6]) has shown marvelous improvements across various NLP tasks, we adopt Chinese BERT to encode word sequences to word embeddings.

In addition, word segmentation and part-of-speech (POS) tagging are useful for Chinese NER. Therefore, we utilize HanLP [10] to divide the sequence into words and tag the POS of each character. For each character, word embeddings generated by Chinese BERT [5] and POS embeddings are concatenated as final word embeddings $w = \{w_1, w_2, \cdots, w_t\}$.

Contextual Embedding Layer. Long-Short Term Memory (LSTM) Neural Network [11] addresses the vanishing gradient problems and is capable of modeling long-term contextual information along the sequence. BiLSTM captures the context from both past and future time steps jointly while vanilla LSTM only considers the contextual information from the past. So, we use BiLSTM to get hidden states as contextual representation of word sequences $H = \{h_1, h_2, \cdots, h_t\}$.

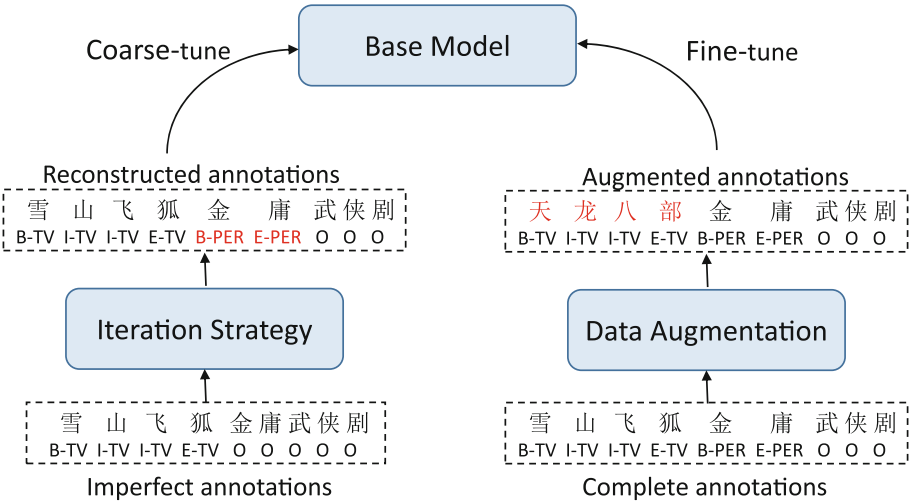


Fig. 2. Architecture of the base model.

Output Layer. The goal of base model is to predict a label sequence that marks the positions of entities. CRF is often used in the sequence tagging model because it captures dependency between the output tags in a neighborhood. During this training, the loss function is formalized as below.

$$\jmath = \sum_{i=1}^{n} l(CRF(H^{(i)}), y^{(i)})$$

where $l(CRF(H^{(i)}), y^{(i)})$ is the negative log-likelihood of the model's prediction $CRF(H^{(i)})$ compared to label sequence $y^{(i)}$.

3.2 Training Process

High quality annotated corpus is very valuable and difficult to obtain, especially in some specific fields, such as finance, mother-infant, healthcare, etc. Most of imperfect annotations suffer from low accuracy, and the performance of the model will be affected when using them directly. Therefore, we propose an iterative strategy and data augmentation method to improve the diversity of data and enrich the entities information.

Iterative Strategy. Since there are lots of unlabeled entities in imperfect annotations that seriously damage the performance, we propose an iterative strategy to reconstruct them to contain more entity information.

Algorithm 1 shows the iterative strategy of reconstructing imperfect annotations. Firstly, considering the imbalanced credibility of labels, we assign different weights W to each character of each sample. Specifically, the weight of each label

of the complete annotations is 1, the weight of the "O" labels in imperfect annotations are 0.6, and the rest are 0.95.

Then, we use incomplete annotations to train the model, and the number of epochs increases with the number of iterations. This is because we find that the precision of the first few epochs of the model is relatively high, the recall is slightly low, so we can obtain reconstructed annotations with high confidence.

Finally, the trained model is used to predict imperfect annotations. According to the prediction results, imperfect annotations are relabeled and the weights of labels are reset to the predicted confidences. In order to ensure the accuracy of relabeling, we only relabel the positions which meet the following requirements:

- The original label is "O";
- The predicted labels are complete entity;
- The confidence of the predicted labels is greater than 0.7.

In addition, in order to avoid the error accumulation of relabeling, reset the model's parameters before each iteration. After K iterations, we obtain the reconstructed annotations.

Algorithm 1. Iterative Strategy.

Input: K: number of Iterations; W: weights of samples; M: base model; D_{ic}: incomplete annotations; D_{non}: non-annotations.
Output: D_{re}: reconstructed annotations.

```
Save initial parameters of model M as M_init;
Set weights W to each sample;
for k = 1 → K do
    Reset the parameters of the model M to M_init;
    Train model M with D_ic for k epochs, get model M_k;
    Use model M_k to predict the D_ic and D_non;
    Update the weights W and relabel D_ic and D_non according to the prediction
    results, get reconstructed annotations D_re.
    Reclassify D_re to get D_ic and D_non according to whether the sequences contain
    any labels.
end for
```

Data Augmentation. Since the number of high quality annotated corpus is so limited, we adopt a specific data augmentation method to expand complete annotations.

Firstly, we get an entity dictionary from complete annotations whose entities are absolutely right. In detail, for a randomly selected (with a probability of 5%) sequence from complete annotations, we replace the entity in the sequence with the other from the entity dictionary then generating a new sample. There are three kinds of entity types, i.e., TV, person and serial. These three types of entities are unevenly distributed, thus we take different replacement-probability (i.e., 10%, 20%, 100%) for three types. All new samples form the augmented

annotations. During the training phase, we use data augmentation technique in each epoch.

Training. Since noises are inevitably introduced by iterative strategy, and augmented annotations are relatively correct. Therefore, the reconstructed annotations are first used to train the model, and then the augmented annotations are used to fine-tune. The weighted cross entropy loss function is used in the training. Algorithm 2 shows the training process.

Algorithm 2. Pipeline of training.

Input: M: base model; D_{re}: reconstructed annotations; D_{cp}: complete annotations;
Output: Trained model.

```
Train model M with D_re for K epochs;
for l = 1 → L do
    Get augmented annotations D_aug from D_cp;
    Train model M with D_aug for one epoch.
end for
```

Ensemble. In order to improve the robustness of the model, we run S times with different random seeds and get S models. Then we propose two ensemble processes: (E1) S models vote for each character in each sequence and choose the label with the highest number of votes; (E2) For each character in each sequence, choose the label with the highest confidence in S models.

4 Experiments

4.1 Setting

Data and Metrics. The corpus is from the caption texts of YouKu video. Three types of entities (TV, person and serial) are considered in this task. This dataset is split into three subsets, 10,000 samples for training, 1,000 samples for developing and 2,000 samples for testing. In the training set, 5,670 samples are not labeled, and 4,330 samples are incompletely annotated. For training data, entities are labeled by matching a given entity list. The entity list is made up of specific categories, which may cover around 30% of entities appearing in the full corpus. For developing and testing data, samples are fully annotated. Just like the other works, we adopt precision, recall and F-Score as metrics.

Experimental Details. The experimental details are introduced below, including settings of hyper-parameters and model details.

1) We use HanLP [10] to get the POS embedding for each sentence. The pretained BERT model is "chinese_wwm_ext"[1] released by Cui [5]. During the whole training process, parameters of BERT module are fixed.

[1] https://github.com/ymcui/Chinese-BERT-wwm.

2) We set learning rate as 0.001, batch size as 256 and we use the RMSProp optimizer for the whole training process. We set $K = 10$ for coarse-tuning stage and $L = 20$ for fine-tuning stage.

4.2 Results

Our experimental results include four parts: (1) comparing with baselines, (2) fine-tune, (3) model ensemble strategies, (4) results on NLPCC2020 shared task: Auto Information Extraction. All metrics are computed on testing data. The following is a detailed introduction for each part.

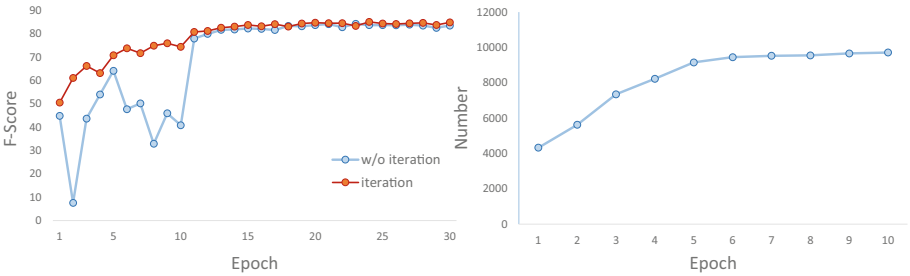
Comparing with Baselines. Firstly, we make comparisons among our coarse-tuning models and baseline models. The BERT+CRF model is released by the organizer and it is adapted from HardLatentCRF [13]. In our work, we use BERT/POS+BiLSTM+CRF as base model. The iterative model is the base model trained with iterative strategy. As shown in the first two rows of Table 1, the two baseline models and our base model perform poorly when trained directly on the imperfect 10,000 training data. However, our base model outperforms two baseline models. When using our well-designed iterative strategy, we make a comparison between our base model and the iterative model. We can see that the iterative model gets a growth of 10.85% compared with our base model.

Table 1. Performance comparison between different baseline models and our models with different strategies.

	Model	Precision	Recall	F-Score
Baselines (w/o Dev)	HardLatentCRF [13]	65.69	36.30	46.76
	BERT+CRF [13]	63.51	64.45	63.98
Coarse-tune (w/o Dev)	Base model	68.24	65.31	66.74
	Iterative model	81.28	74.21	77.59
Fine-tune (with Dev)	Base model	86.62	80.55	83.47
	Iterative model	85.83	**83.93**	84.87
	Iterative model (data augmentation)	87.20	83.02	85.06
Ensemble	Ensemble model (E1)	**87.36**	82.47	84.85
	Ensemble model (E2)	87.27	83.06	**85.11**

As described in Sect. 3, we propose an iterative strategy to reconstruct imperfect annotations. To explore the further capabilities of iterative strategy, we draw the performance curve on the test data during the training process of the model. As shown in the sub-figure (a) of Fig. 3, the first ten epochs are in the coarse-tuning stage, and the rest twenty epochs are in the fine-tuning stage. The blue and red curves correspond to the fifth and sixth rows in Table 1, respectively. Without the iterative strategy, the training process is more unstable in the

coarse-tuning stage. In the sub-figure (b), we can see that the number of valid entities increases with the number of iterations until the training converges.



(a) (b)

Fig. 3. Sub-figure (a) is the model training curves. Sub-figure (b) represents the number of valid entity increases when training with relabeling the unlabeled sentences.

Fine-tune. The results trained on the imperfect 10,000 training data are not satisfactory, thus we propose to fine-tune on the developing data. In the third part of Table 1, models are firstly trained on the 10,000 training data and then fine-tuning on the developing data. On the metric of F-Score, the iterative model performs better than the base model by 1.40%. Compared with the iterative model without fine-tuning in the four row, our fine-tuned iterative model gets growth of 7.28%.

In order to get more fully annotated data when fine-tuning the iterative model on developing data, we use the data augmentation technique in our iterative model. We firstly obtain an entity dictionary from developing data, then we randomly replace some entities to generate new samples. As shown in the seven row of Table 1, our iterative model (with data augmentation) gets the best results on the metric of precision and F-Score.

In the coarse-tuning stage, we get a comparable iterative model, and we get a huge improvement in the fine-tuning stage. We can conclude that our iterative strategy and the idea of fine-tuning on developing data is effective.

Ensemble. During the full training process, we find the models in different training stages have different performance. Some models have better performance on the metric of recall, and others may have better precision. Thus, we try to integrate multiple models in different stages. There are two ensemble strategies as described in Sect. 3. The experimental results are shown in the last two rows of Table 1. By comparing the two strategies E1 and E2, we can see that The E2 strategy is more effective. Compared with the models without ensemble, our ensemble model (E2) has both higher precision and F-Score.

Results on NLPCC2020 Shared Task: Auto Information Extraction. There are two leaderboards in the final contest. The metric of F-Score is computed on testing data. The competition results are shown in Table 2 and Table 3.

We show the top 3 ranked models and the baseline model in each leaderboard. The baseline model is released by the organizer. Table 2 is a ranking of model performance without external data and developing data. Our model performs best among the ranked models, especially outperforming the second place by 5.46% on the F-Score metric. The other leaderboard is the ranking of performance when using developing data, augmented data and integrating models from different training stages. As shown in Table 3, with all the data and ensemble considered, our overall performance is competitive and outperforms the baseline model by 4.00%.

Table 2. Leaderboard1.

Model	F-Score
Rank1 (ours)	77.32
Rank2	71.96
Rank3	71.86
Baseline	63.98

Table 3. Leaderboard2.

Model	F-Score
Rank1	85.00
Rank2 (ours)	84.87
Rank3	84.75
Baseline	80.87

5 Conclusion

In this paper, we considered a complex corpus that contains complete annotations, incomplete annotations, and non-annotations. Unlike most NER systems, only a single strategy is used to process an annotated corpus. We use specific strategies for processing different types of annotations and integrate these strategies to obtain reliable prediction results. To further improve the performance of base models, we use high quality corpus to fine-tune models. In addition, considering the robustness of the system, we also support data augmentation to enhance the diversity of the corpus. These strategies make the system more applicable to real business scenarios. We verify the effectiveness of our approach through comprehensive experiments, and won first and second place respectively in two scenarios provided by NLPCC 2020 Shared Task: Auto Information Extraction. Although our work is evaluated in NER tasks, we believe that the idea of this paper can be well applied to other fields with imperfect labeled sequences.

References

1. Carlson, A., Gaffney, S., Vasile, F.: Learning a named entity tagger from gazetteers with the partial perceptron. In: Learning by Reading and Learning to Read, Papers from the 2009 AAAI Spring Symposium, Technical Report SS-09-07, Stanford, California, USA, 23–25 March 2009, pp. 7–13. AAAI (2009)

2. Chieu, H.L., Ng, H.T.: Named entity recognition: a maximum entropy approach using global information. In: Proceedings of the 19th International Conference on Computational Linguistics-Volume 1, pp. 1–7. Association for Computational Linguistics (2002)
3. Chiu, J.P.C., Nichols, E.: Named entity recognition with bidirectional LSTM-CNNs. Trans. Assoc. Comput. Linguist. **4**, 357–370 (2016)
4. Collins, M., Singer, Y.: Unsupervised models for named entity classification. In: Fung, P., Zhou, J. (eds.) Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, EMNLP 1999, College Park, MD, USA, 21–22 June 1999. Association for Computational Linguistics (1999)
5. Cui, Y., et al.: Pre-training with whole word masking for Chinese BERT. arXiv preprint arXiv:1906.08101 (2019)
6. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (eds.) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019, Volume 1 (Long and Short Papers), pp. 4171–4186. Association for Computational Linguistics (2019)
7. Fernandes, E.R., Brefeld, U.: Learning from partially annotated sequences. In: Gunopulos, D., Hofmann, T., Malerba, D., Vazirgiannis, M. (eds.) ECML PKDD 2011. LNCS (LNAI), vol. 6911, pp. 407–422. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23780-5_36
8. Greenberg, N., Bansal, T., Verga, P., McCallum, A.: Marginal likelihood training of BiLSTM-CRF for biomedical named entity recognition from disjoint label sets. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 2824–2829 (2018)
9. Hanisch, D., Fundel, K., Mevissen, H., Zimmer, R., Fluck, J.: ProMiner: rule-based protein and gene entity recognition. BMC Bioinform. **6**(S-1) (2005)
10. He, H.: HanLP: Han Language Processing (2020). https://github.com/hankcs/HanLP
11. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
12. Jiang, Y., Hu, C., Xiao, T., Zhang, C., Zhu, J.: Improved differentiable architecture search for language modeling and named entity recognition. In: Inui, K., Jiang, J., Ng, V., Wan, X. (eds.) Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, 3–7 November 2019, pp. 3583–3588. Association for Computational Linguistics (2019)
13. Jie, Z., Xie, P., Lu, W., Ding, R., Li, L.: Better modeling of incomplete annotations for named entity recognition. In: Burstein, J., Doran, C., Solorio, T. (eds.) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019, Volume 1 (Long and Short Papers), pp. 729–734. Association for Computational Linguistics (2019)
14. Kim, J., Woodland, P.C.: A rule-based named entity recognition system for speech input. In: Sixth International Conference on Spoken Language Processing, ICSLP 2000/INTERSPEECH 2000, Beijing, China, 16–20 October 2000, pp. 528–531. ISCA (2000)
15. Lafferty, J., McCallum, A., Pereira, F.C.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data (2001)

16. Li, J., Sun, A., Han, J., Li, C.: A survey on deep learning for named entity recognition. IEEE Trans. Knowl. Data Eng., 1 (2020)
17. Liu, S., Tang, B., Chen, Q., Wang, X.: Effects of semantic features on machine learning-based drug name recognition systems: word embeddings vs. manually constructed dictionaries. Information **6**(4), 848–865 (2015)
18. Liu, Y., Meng, F., Zhang, J., Xu, J., Chen, Y., Zhou, J.: GCDT: a global context enhanced deep transition architecture for sequence labeling. In: Korhonen, A., Traum, D.R., Màrquez, L. (eds.) Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, 28 July–2 August 2019, Volume 1: Long Papers, pp. 2431–2441. Association for Computational Linguistics (2019)
19. Lou, X., Hamprecht, F.: Structured learning from partial annotations. arXiv preprint arXiv:1206.6421 (2012)
20. McCallum, A., Li, W.: Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In: Daelemans, W., Osborne, M. (eds.) Proceedings of the Seventh Conference on Natural Language Learning, CoNLL 2003, Held in cooperation with HLT-NAACL 2003, Edmonton, Canada, 31 May – 1 June 2003, pp. 188–191. ACL (2003)
21. McNamee, P., Mayfield, J.: Entity extraction without language-specific resources. In: Roth, D., van den Bosch, A. (eds.) Proceedings of the 6th Conference on Natural Language Learning, CoNLL 2002, Held in cooperation with COLING 2002, Taipei, Taiwan, 2002. ACL (2002)
22. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
23. Morwal, S., Jahan, N., Chopra, D.: Named entity recognition using hidden Markov model (HMM). Int. J. Nat. Lang. Comput. (IJNLC) **1**(4), 15–23 (2012)
24. Nadeau, D., Turney, P.D., Matwin, S.: Unsupervised named-entity recognition: generating gazetteers and resolving ambiguity. In: Lamontagne, L., Marchand, M. (eds.) AI 2006. LNCS (LNAI), vol. 4013, pp. 266–277. Springer, Heidelberg (2006). https://doi.org/10.1007/11766247_23
25. Peng, M., Xing, X., Zhang, Q., Fu, J., Huang, X.: Distantly supervised named entity recognition using positive-unlabeled learning. arXiv preprint arXiv:1906.01378 (2019)
26. Peters, M.E., et al.: Deep contextualized word representations. arXiv preprgreenberg2018marginalint arXiv:1802.05365 (2018)
27. Riloff, E., Jones, R., et al.: Learning dictionaries for information extraction by multi-level bootstrapping. In: AAAI/IAAI, pp. 474–479 (1999)
28. Wang, Z., Shang, J., Liu, L., Lu, L., Liu, J., Han, J.: CrossWeigh: training named entity tagger from imperfect annotations. arXiv preprint arXiv:1909.01441 (2019)
29. Yadav, V., Bethard, S.: A survey on recent advances in named entity recognition from deep learning models. In: Bender, E.M., Derczynski, L., Isabelle, P. (eds.) Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, 20–26 August 2018, pp. 2145–2158. Association for Computational Linguistics (2018)
30. Yang, Y., Chen, W., Li, Z., He, Z., Zhang, M.: Distantly supervised NER with partial annotation learning and reinforcement learning. In: Proceedings of the 27th International Conference on Computational Linguistics, pp. 2159–2169 (2018)
31. Zhang, S., Elhadad, N.: Unsupervised biomedical named entity recognition: experiments with clinical and biological texts. J. Biomed. Inform. **46**(6), 1088–1098 (2013)