Jarvis Lab at SemEval-MeasEval Task 8: A Cascade Count and Measurement Extraction Tool for Scientific Discourse

Jiarun Cao, Yuejia Xiang, Yunyan Zhang, Zhiyuan Qi, Xi Chen[*], Yefeng Zheng
Jarvis Lab, Tencent
{jerrycjrcao,yuejiaxiang,yunyanzhang,jasonxchen,
yefengzheng}@tencent.com,qizhyuan@gmail.com

Abstract

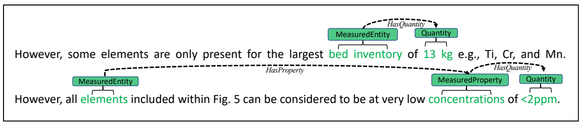
This paper presents our contribution to *SemEval 2021 Task 8: MeasEval*. The purpose of this task is identifying the counts and measurements from clinical scientific discourse, including quantities, entities, properties, qualifiers, units, modifiers, and their mutual relations. This task can be induced to a joint entity and relation extraction problem. Accordingly, we propose CONNER, a **c**ascade c**o**u**n**t a**n**d m**e**asu**r**ement extraction tool that can identify entities and the corresponding relations in a two-step pipeline model. We provide a detailed description of the proposed model hereinafter. Furthermore, the impact of the essential modules and our in-process technical schemes are also investigated. Our code is released and available at https://github.com/yuejiaxiang/CONNER.

1 Introduction

Clinicians are currently coping with a massive amount of information, both from raw experimental data and scientific publications recording their results. However, the ever-expanding information sources have exceeded the ability of clinicians to digest and utilize them properly (Botsis et al., 2011; Cao et al., 2018; Zhou et al., 2010). Clinical information extraction tools (Zhang et al., 2020; De Bruijn et al., 2011; Li and Huang, 2016; Yehia et al., 2019; Mulyar and McInnes, 2020) in the text-mining field make an effort to alleviate the clinician's burden according to exploit how to better utilize the knowledge contained in scientific discourse, accessible in the form of natural human language. Automating the process of understanding the relevant parts of the scientific literature allows for effective searching, and enabling inference of new information and hypothesis generation for clinical research.

Counts and measurements are an important part of information source from the scientific discourse (Harper et al., 2021). However, extracting these count and measurement entities and their interactions is challenging, since the way scientists write them can be ambiguous and inconsistent, and the location of this information relative to the measurement can vary greatly.

In this paper, we focus on the SemEval 2021 MeasEval task which is composed of five sub-tasks that cover span extraction, classification, and relation extraction. As shown in Figure 1 it firstly demands to identify all the quantity spans given a paragraph from a scientific text. For each identified quantity, we need to extract the measured entity, measured property and qualifier which are corresponded to identified quantity. Besides, the relationships between quantity, measured entity, measured property and qualifier are also required to be identified. Lastly, the unit and type of the quantity is also needed to be recognized if they exist.



However, some elements are only present for the largest bed inventory of 13 kg e.g., Ti, Cr, and Mn.

However, all elements included within Fig. 5 can be considered to be at very low concentrations of <2ppm.

Figure 1: A sample of annotated snippet of dataset.

To tackle this challenging task aforementioned, we propose CONNER, a **c**ascade c**o**u**n**t a**n**d m**e**asu**r**ement extraction tool, of which it primarily contains four components: (1) Model encoder gains the representation of both scientific paragraph text and the entities. (2) Quantity tagger extracts all the potential quantity entities within the paragraph. (3) Relation-specific object tagger recognizes the possible measured entities, measured properties and qualifiers, as well as their mutual relations. (4) Unit and modifier extractor identifies units and modifier by both rule-based approach and a simple classifier.

[*] Corresponding author

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 1239–1244
Bangkok, Thailand (online), August 5–6, 2021. ©2021 Association for Computational Linguistics

The rest of the paper is organized as follows. In Section 2, we elaborate on the whole workflow of CONNER and a detailed analysis of our experiments and results in Section 3. The paper is summarised and concluded in Section 4.

2 Model Description

2.1 Overview

The goal of CONNER is designed to identify all possible aspects of quantity items, including `Quantity`, `MeasuredEntity`, `MeasuredProperty`, `Qualifier`, `Unit` and `Modifier`, as well as their relations.

Inspired by (Wei et al., 2019), we assign a cascade framework that models relations as functions that map quantity to other objects in a sentence, which is contrary to the conventional prospective of relational triple extraction task (Gupta et al., 2016; Adel and Schütze, 2017; Zeng et al., 2018; Zheng et al., 2017) that firstly identity all the possible entities, then predict their relations. Based on our observation, there are roughly 90% entities overlapped in our tasks, which can be tackled smoothly by a cascade framework. Besides, our proposed method can just generate a limited amount of negative samples since there are only three types of relations in our tasks, which further prevent from the long-tail problem (Zhang et al., 2019; Li et al., 2020).

The basic idea of CONNER is a two-step pipeline model as shown in Figure 2. We firstly deploy a quantity tagger to identity all the possible `Quantity` from input paragraph in Section 2.2, for each predicted `Quantity`, we check all the potential relations to see if a relation can associate `MeasuredEntity`, `MeasuredProperty` and `Qualifier` with the `Quantity` in Section 2.4. In contrast of utilizing the proposed cascade framework, we adopt a rule-based method and a simple classifier in terms of extraction of unit and modifier in Section 2.5. We describe the detailed workflow below.

2.2 Model Encoder

The model encoder aims at gaining the semantic representations H of input paragraph text X, which will be further used in the following tagging module. In terms of the input paragraph that exceeds our pre-defined maximum length, we split them into pieces via full stop and encoder them separately. We experiment both ROBERTA (Liu et al., 2019) and BERT model (Devlin et al., 2018) to encode the context information. We adopt $H = \text{Encoder}(X)$ for brevity, and L denotes the length of the input paragraph.

2.3 Quantity Tagger

The lower level tagging model shown in Figure 2 is designed to predict the entire potential quantities in the input paragraph, which is an ensemble model incorporating a CRF layer and a PointerNet Layer. We illustrate the whole workflow as follows.

PointerNet layer. Driving from the PointerNetwork (Vinyals et al., 2015), two identical binary classifiers are adopted to detect the start and end position of quantities respectively. Each token is fed into the binary classifiers to predict whether the current token is aligned to a start or end position of a quantity span. Formally, given a contextual representation $h_i \in H$, we have:

$$p_i^{start} = \sigma(\boldsymbol{W}_{start} h_i + \boldsymbol{b}_{start}) \qquad (1)$$

$$p_i^{end} = \sigma(\boldsymbol{W}_{end} h_i + \boldsymbol{b}_{end}) \qquad (2)$$

where $\boldsymbol{b}$ is the bias matrix and σ is the sigmoid activation function. p_i^{start} and p_i^{end} denotes the probability of identifying the i-th token in the input paragraph as the start or end position of a quantity, respectively. We set up a threshold score as 0.7, of which the current token will be assigned to 1 if its probability surpasses the threshold score, otherwise assigned to 0. The loss function of the quantity tagger is the following:

$$\mathcal{L}_{qt} = \frac{1}{L^2} \sum_{i=1}^{L} y_i^{start,end} \log P_i^{start,end} \qquad (3)$$

where L denotes the length of the input paragraph, $y_i^{start,end}$ is the ground truth label. In terms of multiple quantities appeared in the same paragraph, We adopt the same strategy as (Wei et al., 2019) that we adopt the nearest start-end pair match principle to decide the span of any quantity based on the results.

CRF layer. In this layer, we consider quantity recognition as a sequence-labeling problem. We select BIOS(Beginning, Inside, Outside, Single) as our label schema. Accordingly, given the representation sequence $H = (h_1, h_2, ..., h_L)$we adapt a probability-based sequence detection conditional random field (CRF) model (Zheng et al., 2015),

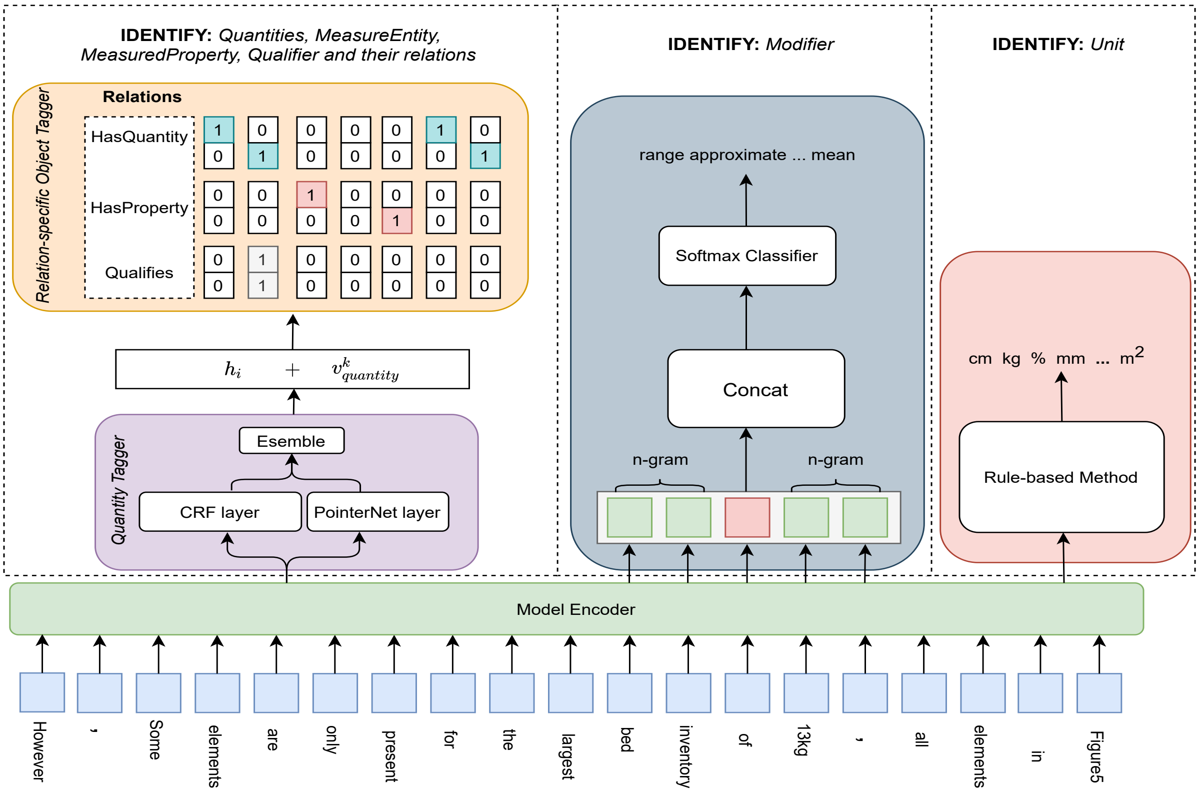


Figure 2: Overall model structure: the left dashed box corresponds to the Subsection 2.3 and 2.4, the right and middle ones correspond to Subsection 2.5.

which defines the conditional probability distribution $P(Y|H)$ of label sequence Y given contextual word representation H aforementioned. We maximize the log-probability during training. In decoding, we set transition costs as infinite if it is invalid. The expected label sequence $Y = (y_1, y_2, ..., y_L)$ is predicted based on maximum scores in the decoding.

Ensemble. The experimental results are relatively comparable in terms of CRF layer and pointerNet layer (See Table 2.3). However, an empirical observation on the predicted results suggest that CRF layer tends to extract shorter spans, while PointerNet layer does the opposite. Presumably, the ensemble model can gain better results since the distribution of entities in the dataset exists both long and short spans.

We deploy our model ensemble considering the predicted quantities that are partly overlapped. To be more specific, we firstly obtain the predicted quantities from PointerNet as our final result, besides, if a predicted quantity from CRF layer strictly does not exist in the PointerNet result, i.e., there is no overlap between the two quantities, we add it to our final result as well.

2.4 Relation-specific Object Tagger

The upper level tagging module identifies the `MeasuredEntity`, `MeasuredProperty` and `Qualifier` as well as the involved relations with respect to the quantities obtained in the previous section. It consists of a set of relationship-specific taggers with the same structure as the quantity tagger in the lower-level for all possible relations. All object taggers identify the corresponding object for each detected quantity simultaneously.

Distinguish from the quantity tagger using representation vector h_i as input, the relation-specific object tagger also takes the quantity semantic features into account. Given each contextual representation of the current token h_i. The detailed tagging operations are as follows:

$$\widetilde{p}_i^{start} = \sigma(\boldsymbol{W}^r_{start}(h_i + \boldsymbol{v}^k_{quantity}) + \boldsymbol{b}^r_{start}) \quad (4)$$

$$\widetilde{p}_i^{end} = \sigma(\boldsymbol{W}^r_{end}(h_i + \boldsymbol{v}^k_{quantity}) + \boldsymbol{b}^r_{end}) \quad (5)$$

where $\widetilde{p}_i^{start}$ and $\widetilde{p}_i^{end}$ denotes the probability of predicting the start and end position of current token. $\boldsymbol{v}^k_{quantity}$ represents the representation of k-th identified quantity via model encoder in Section 2.3.

We iterate all the possible relations across the given quantities. Accordingly, given the token representation h_i and quantity k, the loss function of relation-specific object tagger is as follows:

$$\mathcal{L}_{rot} = \frac{1}{L^2 \cdot R} \sum_{r=1}^{R} \sum_{i=1}^{L} y_{i,r}^{start,end} \log P_{i,r}^{start,end} \tag{6}$$

where $r \in R$ denotes the r-th relation. The rest of symbol keeps the same as Equation 3. Note that tags $y_i^{start,end} = 0$ if the objects are empty.

2.5 Unit and modifier extractor

Unit extraction. We design a set of rules to identify the units which are corresponding to each predicted quantity. The detailed description of schema is presented in Algorithm 1.

Algorithm 1 Unit Method

```
Require: Quantity $Q$, which is a string of $n$ characters.
The collection of all units that have appeared in train & trial: $V$
$p = n$
for $i = 1$ to $n$ do
    if $Q[i:n] \in V$ then
        return $Q[0:i]$
    end if
    if $Q[i]$ is a space then
        $p = i + 1$
    end if
end for
$s = p$
while $s > 0$ and $Q[s-1]$ is acharacter do
    $s = s - 1$
end while
if $s > 0$ then
    $p = s$
end if
return $Q[p:n]$
```

Modifier Classification. As none of the relations attached to the modifier in the input paragraph, we can not apply the relation-specific object tagger in terms of modifier extraction. Given a candidate quantity token x_i, we select its n-gram contextual tokens $\{x_{i-n}, x_{i-n+1}, ..., x_i, x_{i+1}, x_{i+n}\}$ and concatenate them as model input and then simply introduce a plain classifier to predict its labels:

$$c_i = \text{BERT}([x_{i-n}; ...; x_i; ...; x_{i+n}]) \tag{7}$$

$$y_i = \arg\max_{\theta}(\text{softmax}(c_i)) \tag{8}$$

where c_i denotes the representation of quantity and contextual tokens after BERT encoder and y_i is the predicted label of modifier in terms of current token x_i. The training loss is the conventional cross entropy loss, we will not elaborate on it due to the space limit.

3 Experimental Results

3.1 Dataset

This SemEval evaluation has released the dataset online[1], which includes a text file for each paragraph of scientific text along with annotations. As shown in Table 1, the overlapping entities are 9.3%, 0% and 90.7% in total of NEO, EPO and SEP in terms of train/dev/test set, respectively, which indicates the merit of applying CONNER to our tasks since it can naturally handle the overlapping entities.

	Train+Trial	Test
Sentence number	647	593
Avg. Sentence length	45	39
Max. Sentence length	200	304
Triples	2199	-
Cross Sentence Triples	65	-
NEO	203	-
EPO	0	-
SEO	1996	-

Table 1: Statistics of dataset, NEO represents none entity overlap, EPO represents entity pair overlap, SEP represents single entity overlap.

3.2 Experimental Settings

We adopt mini-batch mechanism to train our model with batch size as 8; the pretrained language model finetuing learning rate is set to 2e-5, crf decoder learning rate is set to 5e-3; the hyper-parameters are determined on the validation set. We also complete words with wrong boundaries by design rules, e.g., ”emain mostly neutral” in the raw text is corrected to ”remain mostly neutral”. The maximum length of sentence is set as 350. The number of n-gram is 0. We adopt Adam (Kingma and Ba, 2014) for optimization.

3.3 Main Experiments

The experimental results are conducted in test set, of which each entity category and relations are listed in Table 2. The result of extracting quantity outperforms the rest of entity categories by a large margin regarding named entity recognition. While the HasQuantity naturally achieves the best result in relation extraction task.

[1] The dataset is available at `https://github.com/harperco/MeasEval`

	Prec.(%)	Rec.(%)	F1(%)	F1 Over(%)
Entities				
Quantity	96.16	92.24	75.70	94.16
M.Entity	70.82	52.87	60.54	39.84
M.Prop.	72.79	56.71	63.75	43.66
Qualifier	20.00	12.32	15.25	0.0
Modifier	74.76	63.56	68.70	-
Unit	82.52	84.78	83.64	-
Relations				
HasQuan.	62.85	56.52	59.52	-
HasProp.	57.37	31.72	40.85	-
Qualifies	0	0	0	-
Overall	76.09	57.53	65.52	47.30

Table 2: Experimental results on test set, F1 Over represents F1 overlap. All results are produced by the official evaluation scripts.

3.4 Axuiliary Experiments

During the process of building our proposed system, we tested different schemes for each module of the our model and did relative experiments to compare their experiment results, the scheme with best performance is selected as our final modules consisting of CONNER. We present the in-depth analysis and experimental results listed below.

Model encoder. In Subsection 2.2, we separately adopt `BERT-based` and `ROBERTA-based` our model encoder. To examine the performance regarding different model encoder, we conduct experiments in the quantity identification stage for both identification of entities and relations. As we can notice in Table 3, ROBERTA-base all outperforms BERT-base so that it is selected as our final model encoder.

	Prec.(%)	Rec.(%)	F1(%)
Entities			
BERT-large	58.85	56.02	57.40
ROBERTA-large	60.37	57.68	58.99
Relations			
BERT-large	49.52	45.67	47.39
ROBERTA-large	49.52	52.94	51.17

Table 3: Experimental results of different model encoders

Settings of ensemble scheme. We tested the result of utilizing CRF layer and PointerNet layer independently, it shows comparable results as listed in Table 4. As we mentioned in in Subsection 2.3, combining the results of CRF and PointerNet can make the best use of both models, and results verified our assumption that ensemble models all outperform the singular models.

we also carried out two different ensemble approach for quantity tagger. The first one is as illustrate in Subsection 2.3. The second approach is simpler: we take the union of the predicted quantities of CRF layer and PointerNet layer, and remove duplicate as our final prediction result. The experimental results in Table 4 suggested the first ensemble model achieve the best result, so that it is selected as our final ensemble scheme.

	Prec.(%)	Rec.(%)	F1(%)
CRF layer	60.37	57.68	58.99
PointerNet layer	59.67	56.53	58.06
Union ensemble	58.54	59.78	59.15
ensemble in Section 2.3	60.47	59.02	59.73

Table 4: Experimental results of different model encoders

Settings of n-gram. Different number of n-gram can affect the model performance to some extent, we thus tested introducing different length of context regarding extraction of the modifier. As shown in Table 5, the model achieves best performance with 45.13% F1 score when n is 0, meanwhile, we speculate the underlying reason is that model is not capable of capturing valid semantics from contextual tokens due to the limited amount of the modifiers in the whole dataset.

n-gram	F1(%)
none context	**45.13**
window_char_5	42.22
window_char_10	41.84
window_word_1	44.63
window_word_3	44.08

Table 5: Experimental results of different model encoders

4 Conclusion

We proposed CONNER, a cascade count and measurement extraction tool to jointly identify the quantities and their attached items, as well as the corresponding relations for SemEval 2021 Task 8: MeasEval. Our model extracts these entities and relations in a two-step pipeline method. We also exploited various of technical schemes during the competition and select the one that gains the best performance in the experiments, which help us win second-place in the final ranking.

References

Heike Adel and Hinrich Schütze. 2017. Global normalization of convolutional neural networks for joint entity and relation classification. In *Proceedings of the 2017 Conference on Empirical Methods in*

Natural Language Processing, pages 1723–1729, Copenhagen, Denmark. Association for Computational Linguistics.

Taxiarchis Botsis, Michael D Nguyen, Emily Jane Woo, Marianthi Markatou, and Robert Ball. 2011. Text mining for the vaccine adverse event reporting system: medical text classification using informative feature selection. *Journal of the American Medical Informatics Association*, 18(5):631–638.

Jiarun Cao, Chongwen Wang, and Liming Gao. 2018. A joint model for text and image semantic feature extraction. In *Proceedings of the 2018 International Conference on Algorithms, Computing and Artificial Intelligence*, pages 1–8.

Berry De Bruijn, Colin Cherry, Svetlana Kiritchenko, Joel Martin, and Xiaodan Zhu. 2011. Machine-learned solutions for three stages of clinical information extraction: the state of the art at i2b2 2010. *Journal of the American Medical Informatics Association*, 18(5):557–562.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Pankaj Gupta, Hinrich Schütze, and Bernt Andrassy. 2016. Table filling multi-task recurrent neural network for joint entity and relation extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2537–2547, Osaka, Japan. The COLING 2016 Organizing Committee.

Corey Harper, Jessica Cox, Curt Kohler, Antony Scerri, Ron Daniel Jr., and Paul Groth. 2021. SemEval 2021 task 8: MeasEval – extracting counts and measurements and their related contexts. In *Proceedings of the Fifteenth Workshop on Semantic Evaluation (SemEval-2021)*, Bangkok, Thailand (online). Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Peng Li and Heng Huang. 2016. Clinical information extraction via convolutional neural network. *arXiv preprint arXiv:1603.09381*.

Yang Li, Tao Shen, Guodong Long, Jing Jiang, Tianyi Zhou, and Chengqi Zhang. 2020. Improving long-tail relation extraction with collaborating relation-augmented attention. *arXiv preprint arXiv:2010.03773*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Andriy Mulyar and Bridget T McInnes. 2020. Mt-clinical bert: Scaling clinical information extraction with multitask learning. *arXiv preprint arXiv:2004.10220*.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. *arXiv preprint arXiv:1506.03134*.

Zhepei Wei, Jianlin Su, Yue Wang, Yuan Tian, and Yi Chang. 2019. A novel cascade binary tagging framework for relational triple extraction. *arXiv preprint arXiv:1909.03227*.

Engy Yehia, Hussein Boshnak, Sayed AbdelGaber, Amany Abdo, and Doaa S Elzanfaly. 2019. Ontology-based clinical information extraction from physician's free-text notes. *Journal of biomedical informatics*, 98:103276.

Xiangrong Zeng, Daojian Zeng, Shizhu He, Kang Liu, and Jun Zhao. 2018. Extracting relational facts by an end-to-end neural model with copy mechanism. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 506–514, Melbourne, Australia. Association for Computational Linguistics.

Ningyu Zhang, Shumin Deng, Zhanlin Sun, Guanying Wang, Xi Chen, Wei Zhang, and Huajun Chen. 2019. Long-tail relation extraction via knowledge graph embeddings and graph convolution networks. *arXiv preprint arXiv:1903.01306*.

Yuanzhe Zhang, Zhongtao Jiang, Tao Zhang, Shiwan Liu, Jiarun Cao, Kang Liu, Shengping Liu, and Jun Zhao. 2020. Mie: A medical information extractor towards medical dialogues. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6460–6469.

Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. 2015. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1529–1537.

Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. 2017. Joint extraction of entities and relations based on a novel tagging scheme. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1227–1236, Vancouver, Canada. Association for Computational Linguistics.

Xuezhong Zhou, Yonghong Peng, and Baoyan Liu. 2010. Text mining for traditional chinese medical knowledge discovery: a survey. *Journal of biomedical informatics*, 43(4):650–660.