CofeNet: Context and Former-Label Enhanced Net for Complicated Quotation Extraction

Yequan Wang[1*], Xiang Li[2*], Aixin Sun[3], Xuying Meng[4], Huaming Liao[4], Jiafeng Guo[4]
[1]Beijing Academy of Artificial Intelligence, Beijing, China
[2]Alibaba Group, China
[3]School of Computer Science and Engineering, Nanyang Technological University, Singapore
[4]Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
tshwangyequan@gmail.com, yuanye.lx@alibaba-inc.com, axsun@ntu.edu.sg,
{mengxuying, lhm, guojiafeng}@ict.ac.cn

Abstract

Quotation extraction aims to extract quotations from written text. There are three components in a quotation: *source* refers to the holder of the quotation, *cue* is the trigger word(s), and *content* is the main body. Existing solutions for quotation extraction mainly utilize rule-based approaches and sequence labeling models. While rule-based approaches often lead to low recalls, sequence labeling models cannot well handle quotations with complicated structures. In this paper, we propose the **Co**ntext and **F**ormer-Label **E**nhanced **Net** (CofeNet) for quotation extraction. CofeNet is able to extract complicated quotations with components of variable lengths and complicated structures. On two public datasets (*i.e.,* PolNeAR and Riqua) and one proprietary dataset (*i.e.,* PoliticsZH), we show that our CofeNet achieves state-of-the-art performance on complicated quotation extraction.

1 Introduction

Quotation extraction aims to extract quotations from written text (Pouliquen et al., 2007). For example, given one instance shown in Figure 1, we extract the quotation with *source*: *some democrats*, *cue*: *privately express*, and *content*: *reservations about ...*. As a point of view, quotations provide opinions of the speaker, which is important for analyzing the speaker's stand. In general, quotation extraction is the first step before any further analysis, *e.g.,* speaker stand detection. In this paper, we focus on the extraction of the three quotation components.

As illustrated in the above example, the extraction of *content* component in a quotation is complicated and difficult due to three reasons: variable length, unclear boundary, and indistinguishable components. Specifically, the length of *content* can be over 10, or even more than 50 tokens. Moreover, *content* does not come with a regular pattern, which not only leads to a more unclear boundary of itself, but also affects the estimation of *source* and *cue*. For example, *content* in a quotation can be a complete instance with subject, predicate, and object. It is therefore hard to distinguish a noun (subject or object) representing the *source* or a part of *content*. Difficulty also exists in recognition of *cue* when tackling with a predicate, *e.g.,* verb. Thus, as *content* may contain another quotation, such a nesting structure further increases the difficulty of extracting quotations.

Yet for all the symbolism and feel-good value of such an appointment, some democrats privately express reservations about entrusting a seat that could decide the balance of power in the closely divided senate to a candidate who has never won statewide, is considered less than dynamic and has been an anemic fundraiser.

Figure 1: An example of quotations. Text spans with orange, green and gray denote *source*, *cue* and *content* respectively.

Many existing solutions for quotation extraction are rule-based methods (Pouliquen et al., 2007; Krestel et al., 2008; Elson and McKeown, 2010; Vu et al., 2018). Generally, quotations include direct quotations and indirect quotations. Quotation marks and their variants are clear; thus *content* can be extracted by using regular expressions. However, not all quoted texts are quotations. Meanwhile, not all quotations are quoted. Another popular rule-based approach is to recognize *cue* words, *e.g., speak(s)*. Similarly, not all *cue* words are related to quotations and vice versa. For both approaches, after recognizing *content* or *cue*, they usually search for the nearby noun as *source*. In short, rule-based methods only cover limited cases, leading to serious low recall problems.

Quotation extraction has also been formulated

[*]Indicates equal contribution

Proceedings of the 29th International Conference on Computational Linguistics, pages 2438–2449
October 12–17, 2022.

as a sequence labeling task. Pareti et al. (2013); Lee et al. (2020) directly adopt sequence labeling for quotation extraction. However, these solutions ignore the traits of quotations where lengths of quotation components are variable and structures of *content* are complicated. In general, *source* and *cue* components are short, *e.g.,* ≤ 3 tokens. However, *content* usually is over 10 tokens, or even more. Further, the complicated structure of *content* greatly reduces the performance of *content* extraction for sequence-labeling-based solutions.

In this paper, we propose **Co**ntext and **F**ormer-Label **E**nhanced **Net** (CofeNet) for quotation extraction. CofeNet is a novel architecture to extract quotations with variable-length and complicated-structured components. Our model is also capable of extracting both direct and indirect quotations.

CofeNet extracts quotations by utilizing dependent relations between sequenced texts. The model contains three components, *i.e., Text Encoder*, *Enhanced Cell*, and *Label Assigner*. Given a piece of text, the encoder encodes the instance and outputs the encoded hidden vectors. We design the Enhanced Cell module to study semantic representations of variable-length components with the utilization of contextual information. Specifically, the enhanced cell (i) uses a composer layer to enhance the input with the former labels (which are predicted by the former cells), the former words, the current word, and the latter words encoded by the encoder; and (ii) uses a gate layer and an attention layer to control and attend the corresponding input when predicting the label of the current word, at the level of element and vector respectively. Experimental results on two public datasets (*i.e.,* PolNeAR and Riqua) and one proprietary dataset (*i.e.,* PoliticsZH) show that our CofeNet achieves state-of-the-art performance on complicated quotation extraction.

2 Related Work

At first glance, quotation detection is a kind of "triplet" extraction, making the task similar to another two tasks, open information extraction (Angeli et al., 2015; Gashteovski et al., 2017) and semantic role labeling (Exner and Nugues, 2011). However, these three tasks have different focuses. Arguments extracted by semantic role labeling are event-related factors. OpenIE aims to output a structured representation of an instance in the form of binary or n-ary tuples, each of which consists of a predicate and several arguments. The extracted text spans in both tasks are typically short and less complicated, compared to the *content* in quotations. Because *content* extraction is the key challenge in quotation extraction, we will not further elaborate on semantic role labeling and OpenIE. Prior work on quotation extraction can be grouped into rule-based and sequence labeling methods.

2.1 Rule-based Methods

Extracting indirect quotations without clear boundaries is a challenging task, so early studies focus on rule-based methods to extract direct quotations (Pouliquen et al., 2007; Krestel et al., 2008; Elson and McKeown, 2010). In fact, rule-based methods perform well for marked texts, especially for direct quotations.

Pattern matching is a popular method in early studies. Pouliquen et al. (2007); Elson and McKeown (2010) identify *content*, *cue* and *source* by known quote-marks, pre-defined vocabulary, and rules of pattern recognition. The difference is that Elson and McKeown (2010) add machine learning methods to the quote attribution judgment so that they can process complex text. O'Keefe et al. (2012) use regular expressions to recognize quote-marks to extract components, then use sequence labeling to recognize quotation triplets.

Hand-built grammar is another popular rule-based method. Krestel et al. (2008) design a system by combining common verbs corresponding to *cue* and hand-built grammar to detect constructions that match six general lexical patterns. PICTOR (Schneider et al., 2010) utilizes context-free grammar to extract components of quotations.

2.2 Sequence Labeling Methods

Due to the development of deep learning, sequence-labeling-based approaches have attracted attention (Pareti et al., 2013; Lee et al., 2020). To identify the beginning of a quotation, Fernandes et al. (2011) use sequence labeling with features including part-of-speech and entity features generated by a guided transformation learning algorithm. Then they use regular expressions to recognize the *content* within quotations. Pareti et al. (2013) follow a similar idea but use CRF to decode the label. Lee et al. (2020) further use BERT to encode the text and CRF to decode the label on a non-public Chinese news dataset. However, these models cannot well handle quotations with complicated structures.

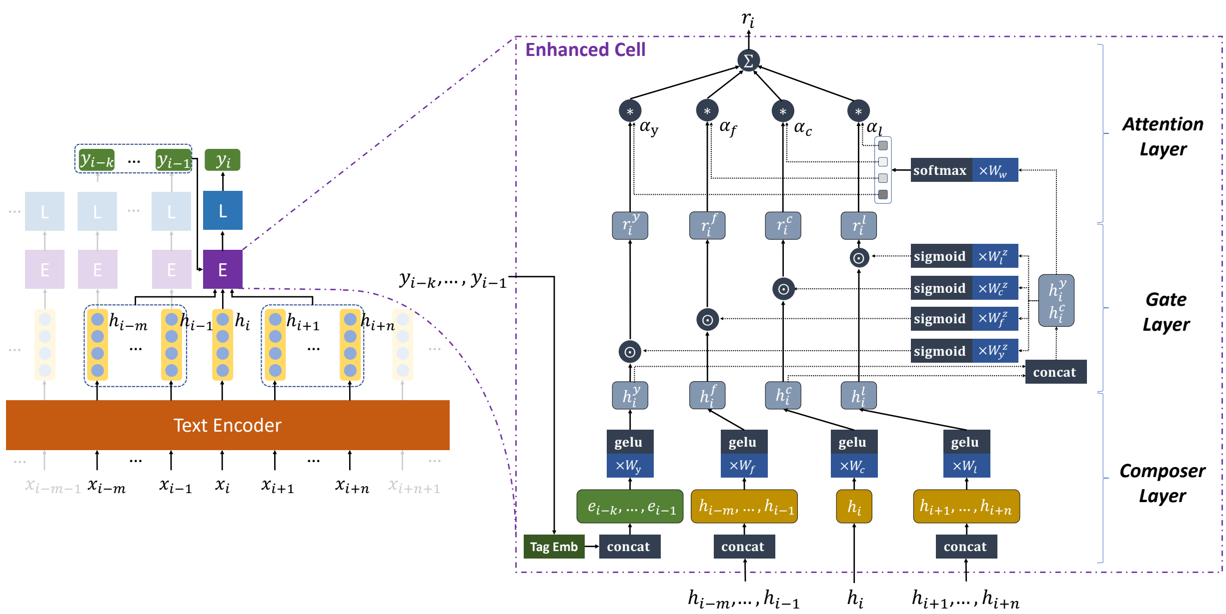

Figure 2: The architecture of CofeNet. Enhanced Cell is detailed on the right-hand side. (best viewed in color)

3 CofeNet Model

Figure 2 depicts the architecture of CofeNet. It consists of three modules: *Text Encoder*, *Enhanced Cell*, and *Label Assigner*. Text encoder is used to encode the input text to get hidden representations. Then, the enhanced cell is capable of building a representation considering the trait of quotations including variable-length and complicated-structured components. Last, the label assigner is to assign labels “B-source”, “B-cue”, “B-content”, “I-source”, “I-cue”, “I-content” and “O”, with BIO scheme.

3.1 Text Encoder

CofeNet is generic and can be realized by popular encoders such as LSTM (Hochreiter and Schmidhuber, 1997), CNN (Kim, 2014), Recursive Neural Network (Socher et al., 2011), and BERT (Devlin et al., 2019a). Unless otherwise specified, CofeNet denotes the model using BERT (Devlin et al., 2019b) as the encoder.

Given input text, hidden states of words are formulated by:

$$\{h_1, h_2, \ldots, h_N\} = \text{Encoder}(\{x_1, x_2, \ldots, x_N\}),$$

where, x_i is the i-th word of input, and Encoder denotes the Text Encoder. The hidden state h_i denotes the representation of i-th word x_i while encoding the preceding contexts of the position.

3.2 Enhanced Cell

As aforementioned, the challenge of quotation extraction is to extract the complicated-structured components with variable lengths. To this end, we design the enhanced cell with composer layer, gate layer, and attention layer, to study the semantic representations of variable-length components. At the same time, we also try to utilize contextual information and predicted labels.

Shown in Figure 2, the composer is used to reformat the input information to include the former labels $y_{i-k}, \ldots, y_{i-1}$, the former hidden states $h_{i-m}, \ldots, h_{i-1}$, the current state h_i, and the latter states $h_{i+1}, \ldots, h_{i+n}$. In this way, our model is able to consider a long span with different structures in a more coherent manner on top of encoded word representations. In general, the influence of different inflow information is different. To this end, we use a gate mechanism to control each element of input representations, and an attention mechanism to weigh the input representations at the vector level. Through the two mechanisms, we get a refined representation so that we could hold the complicated-structured and variable-length components of quotations. Next, we detail the workflow of the enhanced cell.

Composer Layer. The composer contains a label embedding unit and a linear unit to reformat the inflow information: the former labels $\{y_{i-k}, \ldots, y_{i-1}\}$, the former hidden states $\{h_{i-m}, \ldots, h_{i-1}\}$ of previous m words, the current state h_i of the current word x_i, and the latter states $\{h_{i+1}, \ldots, h_{i+n}\}$ of latter n words.

First, the enhanced cell contains a label embedding unit, which is able to select the embedding of the given label, formulated by:

$$e_i = \text{Emb}(y_i), \tag{1}$$

where Emb denotes the mentioned label embedding unit. The predicted label of word i is y_i and the embedding of y_i is e_i. Taking the former k predicted labels into consideration, we get the former labels' representations $[e_{i-k}, \dots, e_{i-1}]$ by concatenation, which is shown as a rectangle in green background, in the Enhanced Cell in Figure 2.

Intuitively, contextual information is important for us to predict the label of the current input word. We take the following context through simple but effective linear layers: the former predicted k labels, the former m words, the current word i, and the latter n words.

$$h_i^y = \text{GELU}([e_{i-k}, \dots, e_{i-1}]W_y + b_y) \tag{2}$$

$$h_i^f = \text{GELU}([h_{i-m}, \dots, h_{i-1}]W_f + b_f) \tag{3}$$

$$h_i^c = \text{GELU}(h_i W_c + b_c) \tag{4}$$

$$h_i^l = \text{GELU}([h_{i+1}, \dots, h_{i+n}]W_l + b_l) \tag{5}$$

In the above formulation, the hidden states $\{h_{i-m}, \dots, h_i, \dots, h_{i+n}\}$ and label embeddings $\{e_{i-k}, \dots, e_{i-1}\}$ are the input. W_y, W_f, W_c, W_l and b_y, b_f, b_c, b_l are the parameters of the linear layers. Here, we adopt GELU as the active function. $h_i^y, h_i^f, h_i^c, h_i^l$ denote the farther hidden states of the former labels, the former words, the current word and the latter words, respectively.

Gate Layer. The influence of different contexts is different. Hence, we use a gate mechanism to control the inflow hidden states at the element level. Inspired by Hochreiter and Schmidhuber (1997), we design a gate layer in the enhanced cell:

$$r_i^y = h_i^y \odot \text{sigmoid}([h_i^y, h_i^c]W_y^z + b_y^z) \tag{6}$$

$$r_i^f = h_i^f \odot \text{sigmoid}([h_i^y, h_i^c]W_f^z + b_f^z) \tag{7}$$

$$r_i^c = h_i^c \odot \text{sigmoid}([h_i^y, h_i^c]W_c^z + b_c^z) \tag{8}$$

$$r_i^l = h_i^l \odot \text{sigmoid}([h_i^y, h_i^c]W_l^z + b_l^z) \tag{9}$$

In the above formulation, r_i^y, r_i^f, r_i^c, and r_i^l denote the adjusted states of the former labels, the former words, the current word, and the latter word representation, respectively. The operation $\odot$ denotes element-wise product. W_y^z, W_f^z, W_c^z, W_l^z, and b_y^z, b_f^z, b_c^z, b_l^z are the parameters. We use sigmoid to adjust each element of the inflow representations.

Attention Layer. Inspired by Wang et al. (2016); Yang et al. (2016); Wang et al. (2018); Lin et al. (2019); Meng et al. (2022), we use an attention mechanism to attend the important part of r_i^y, r_i^f, r_i^c, and r_i^l. Since our target is to predict the label of the current word, we use the concatenation of h_i^y and h_i^c to attend the four vectors by

$$\alpha_y, \alpha_f, \alpha_c, \alpha_l = \text{softmax}([h_i^y, h_i^c]W_w + b_w), \tag{10}$$

where $\alpha_y, \alpha_f, \alpha_c$, and α_l are the weights for r_i^y, r_i^f, r_i^c, and r_i^l respectively. W_w and b_w are the parameters. In the attention layer, softmax function is used to calculate weights. Then, the current word representation r_i is obtained via:

$$r_i = \alpha_y r_i^y + \alpha_f r_i^f + \alpha_c r_i^c + \alpha_l r_i^l \tag{11}$$

To summarize, the Enhanced Cell uses the gate and attention layers with contextual information (*i.e.,* former labels, former words, current word, and latter words) to handle complicated-structured components with variable lengths. Specifically, to sense continuous span, we use attention layer by attending contextual information at the vector (macro) level, by using former labels, and the former, current, and latter word(s). Thus, the model avoids undesirable interruption within an instance. We also use the gate layer to control contextual information at the element (micro) level, especially former labels. Further, thanks to the ability of fine control, the gate layer is capable of avoiding illegal patterns, *e.g.,* "O" followed by "I-*".

3.3 Label Assigner

After getting the hidden representation of the current word, we use label assigner module to compute a probability distribution of the current label.

Briefly speaking, in label assigner, we use softmax classifier to calculate the distribution $\mathcal{P}_i$ of the current word i. Then argmax is used to assign a label of the current word. The two operations can be formulated as

$$\mathcal{P}_i = \text{softmax}(r_i W_p + b_p), \tag{12}$$

$$y_i = \text{argmax}(\mathcal{P}_i), \tag{13}$$

where W_p and b_p are the parameters.

3.4 Training Objective

The proposed CofeNet model could be trained in an end-to-end way by backpropagation. We adopt the cross-entropy objective function that has been used in many studies (Tang et al., 2015; Wang et al., 2016, 2019).

Sequence Labeling Objective. Similar to sequence labeling tasks, we evaluate the label of all words for each given training instance. Recall that our objective is to predict the label of each word in the given instance. The unregularized objective L can be formulated as cross-entropy loss:

$$L(\theta) = -\sum_i \sum_j l_i^j \log(\mathcal{P}_i^j) \quad (14)$$

For a given training instance, l_i^j is the ground truth of label j for word i. Correspondingly, $\mathcal{P}_i^j$ is the probability of label j for word i. θ is the parameter set.

4 Experiment

We now evaluate the proposed CofeNet on two public datasets (*i.e.,* PolNeAR and Riqua), and one proprietary dataset (*i.e.,* PoliticsZH) against baselines. The implementation details and parameter settings are presented in Appendix A. On all datasets, we train the model with the training set, tune hyperparameters on the validation set, and report performance on the test set.

4.1 Datasets

PolNeAR. Political News Attribution Relations Corpus (PolNeAR) (Newell et al., 2018) is a corpus of news articles in English, on political candidates during US Presidential Election in November 2016. PolNeAR annotations are univocal, meaning that each word has only one label (*source*, *cue*, *content*, or none). The average number of tokens is 46.

Riqua. RIch QUotation Annotations (Riqua) (Papay and Padó, 2020) provides quotations, including interpersonal structure (speakers and addressees) for English literary. This corpus comprises 11 works of 19th-century literature that are manually annotated for direct and indirect quotations. Each instance, typically a sentence, is annotated with its *source*, *cue*, and *content*. The average number of tokens in this corpus is 129, longer than PolNeAR.

PoliticsZH. Chinese Political Discourse (PoliticsZH) contains politics and economics news collected from mainstream online media of China including Xinhua Net[1]. The news are in Chinese and the average length of input is 69 tokens, longer than PolNeAR but shorter than Riqua.

[1] http://news.cn/

Table 1: The statistics of three datasets. "Ave. len." refers to "Average length".

Dataset	Number of sentences			Ave. len. in tokens		
	Train	Valid	Test	Source	Cue	Content
PolNeAR	17,397	1,925	1,814	3.27	1.88	14.49
Riqua	1,604	208	105	1.38	1.08	20.65
PoliticsZH	10,754	1,344	1,345	3.08	1.80	43.47

Table 1 presents the statistics of the three datasets. We observe that the numbers of instances of PolNeAR and PoliticsZH are at the order of $10k$, and the Riqua is at $1k$. The length of *source* and *cue* is less than 5 tokens. The length of *content* is greater than 10, even 40 tokens. Note that for all three datasets, the length of *content* is much longer than *source* and *cue*.

4.2 Compared Methods

To provide a comprehensive evaluation, we experiment on both deep learning (*i.e.,* CNN, GRU, (Bi)LSTM, BERT, and BERT-CRF), and traditional methods (*i.e.,* Rule and CRF).

Rule. O'Keefe et al. (2012) uses rules including entity dictionary, reported speech verbs, and special flag characters to extract components of quotations.

CoreNLP. CoreNLP (Vu et al., 2018) contains quote extraction pipeline which deterministically picks out *source* and *content* from a text while ignoring *cue*.

CRF. Lafferty et al. (2001) present CRF to label sequence by building probabilistic models.

CNN. CNN (LeCun et al., 1995), a simple and parallelized model, can be independently adopted for sequence labeling tasks (Xu et al., 2018).

(Bi)LSTM. LSTM (Hochreiter and Schmidhuber, 1997) is able to exhibit dynamic temporal behavior due to its well-designed structure. We use it and its variants, *i.e.,* Bidirectional LSTM (BiLSTM).

GRU. GRU is a slightly more dramatic variation of LSTM (Cho et al., 2014).

BERT(-CRF). BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right contexts (Devlin et al., 2019a).

4.3 Evaluation Metrics

The components of quotations are variable-length and complicated. As a result, it requires more spe-

Table 2: The $F1$ and J(accard) of methods on PolNeAR, Riqua and PoliticsZH datasets. The results marked with * are obtained by calling the CoreNLP toolkit package directly.

Dataset	Model	Source			Cue			Content		
		$F1$-E.	$F1$-B.	J	$F1$-E.	$F1$-B.	J	$F1$-E.	$F1$-B.	J
PolNeAR	Rule	10.7	13.0	8.8	22.8	25.3	14.4	5.6	10.5	6.1
	CoreNLP*	13.9	21.3	11.1	-	-	-	17.5	18.7	12.8
	CRF	50.6	56.2	42.1	53.4	63.3	44.1	28.6	50.9	42.3
	CNN	52.7	65.9	45.1	58.4	67.8	49.4	16.2	60.6	30.2
	GRU	46.5	58.2	36.7	59.1	68.1	48.8	51.3	65.0	51.3
	BiLSTM	64.1	74.4	56.8	63.3	72.6	55.1	53.4	67.3	53.7
	BERT	<u>81.1</u>	<u>86.2</u>	<u>74.8</u>	<u>74.0</u>	<u>81.1</u>	<u>67.4</u>	<u>68.9</u>	<u>78.7</u>	<u>70.0</u>
	CofeNet	**83.2**	**87.1**	**76.4**	**75.3**	**82.3**	**69.4**	**72.9**	**79.6**	**73.2**
Riqua	Rule	16.8	16.8	11.2	36.5	36.5	22.3	0.0	2.4	2.4
	CoreNLP*	22.8	22.8	17.9	-	-	-	63.8	63.8	46.9
	CRF	46.9	51.0	32.9	59.6	65.7	46.6	42.7	85.9	62.2
	CNN	52.7	59.1	39.6	85.2	85.2	74.2	45.2	95.4	58.5
	GRU	55.8	62.9	43.4	77.1	77.1	62.8	92.5	95.2	89.6
	BiLSTM	56.4	64.1	44.5	85.4	85.4	74.4	92.2	95.9	90.3
	BERT	<u>74.5</u>	<u>77.9</u>	<u>62.4</u>	<u>88.9</u>	<u>88.9</u>	<u>80.0</u>	<u>94.3</u>	<u>96.6</u>	<u>92.9</u>
	CofeNet	**81.8**	**84.3**	**72.6**	**89.2**	**89.2**	**80.4**	**94.4**	**97.1**	**94.1**
PoliticsZH	Rule	78.8	79.3	66.8	80.3	81.2	69.7	0.4	7.0	3.7
	CoreNLP*	38.1	39.5	24.3	-	-	-	0.2	2.2	4.3
	CRF	81.6	84.0	72.2	80.0	80.4	68.5	45.7	49.1	66.3
	CNN	82.5	87.8	76.5	81.4	83.6	72.1	35.0	74.5	46.7
	GRU	85.5	88.3	78.1	82.1	84.6	73.6	65.7	79.8	71.5
	BiLSTM	87.5	91.3	83.3	86.2	88.6	79.9	70.3	81.8	74.9
	BERT	<u>92.6</u>	<u>93.7</u>	<u>88.2</u>	<u>89.5</u>	<u>90.8</u>	<u>84.0</u>	<u>73.7</u>	<u>83.6</u>	<u>84.4</u>
	CofeNet	**93.7**	**94.4**	**89.8**	**90.3**	**91.1**	**85.4**	**78.0**	**86.9**	**88.7**

cific metrics. To this end, we evaluate the performance of models using our proposed "Jaccard", in addition to "Exact Match" and "Begin Match".

Exact Match. To measure the overall prediction at the instance level, we propose Exact Match index to quantify whether the multi-label prediction exactly matches the annotation. In the experiments, we use accuracy, precision, recall, and $F1$ to evaluate the exact match performance.

Begin Match. Exact match is harsh, especially long text span. Generally, the length of *source* and *cue* is short while the *content* is much longer. As a result, exact match is hard for *content*. To this end, we use begin match to evaluate only the beginning location for text span matching (Lee and Sun, 2019).

Jaccard. For text span matching, an important index is a ratio of the overlapping span over the total span. Thus we use "Jaccard" index to evaluate the performance of model in this aspect. Given the groundtruth text span $\mathcal{T}_g$ and its predicted text span $\mathcal{T}_p$, we can calculate the Jaccard index J through

$$J = \frac{|\mathcal{T}_p \cap \mathcal{T}_g|}{|\mathcal{T}_p \cup \mathcal{T}_g|}. \tag{15}$$

4.4 Main Results

Table 2 lists the $F1$ and J(accard) performance on the three datasets. In this table, the best results are in boldface and the second-best are underlined. We report results by exact match, begin match, and Jaccard, of all models for the three components of quotations. Here, $F1$-E. and $F1$-B. refer to the $F1$ based on exact match and begin match, respectively. The precision, recall and accuracy are shown in the page[2] due to space limitation. Our CofeNet model is listed in the last row of each dataset.

Table 2 shows that our CofeNet performs the best against all baselines. BERT achieves the second-best, followed by other deep-learning-based models. Note that due to the settled human-written rules, the performance of Rule and CoreNLP is not stable. For *source* and *cue*, on PoliticsZH, the performance is good due to more comprehensive rules. However, the rules on the other two datasets do not fit the domain well. As a comparison, *content* is on the opposite side. For *content*, the precision and recall of CoreNLP are 97.2 and 47.5 on Riqua dataset, which is better than PolNeAR. PoliticsZH dataset shows the worst performance. This is be-

[2] `https://thuwyq.github.io/docs/cofenet-detail-exp.pdf`

Table 3: The $F1$ and J of methods on PolNeAR. *B.L.* and *B.L.C.* denote BiLSTM and BiLSTM+CRF respectively.

Model	Source			Cue			Content		
	$F1$-E.	$F1$-B.	J	$F1$-E.	$F1$-B.	J	$F1$-E.	$F1$-B.	J
CNN	52.7	65.9	45.1	58.4	67.8	49.4	16.2	60.6	30.2
w. CRF	+8.3	**+4.1**	+8.0	**+4.3**	**+2.2**	**+3.6**	+25.8	+1.9	+19.3
w. Cofe	**+9.4**	+3.9	**+8.1**	+3.7	+2.1	+3.2	**+31.8**	**+3.1**	**+21.9**
GRU	46.5	58.2	36.7	59.1	68.1	48.8	51.3	65.0	51.3
w. CRF	+19.3	+13.7	+19.3	+6.2	+3.9	+6.8	+3.8	+0.8	**+6.2**
w. Cofe	**+20.5**	**+14.6**	**+19.7**	**+7.2**	**+4.6**	**+7.5**	**+6.9**	**+1.9**	+6.2
LSTM	46.1	56.4	35.7	58.6	67.5	47.9	50.4	65.5	50.8
w. CRF	+19.4	+14.7	+19.4	+6.4	+4.2	+6.7	+4.6	+0.3	+5.4
w. Cofe	**+21.8**	**+16.3**	**+20.9**	**+6.5**	**+4.3**	**+7.1**	**+7.6**	**+0.7**	**+6.0**
BiLSTM	64.1	74.4	56.8	63.3	72.6	55.1	53.4	67.3	53.7
w. CRF	+5.5	+1.3	+4.5	+3.4	+1.2	+2.6	+5.6	+2.1	+6.6
w. Cofe	**+7.1**	**+3.7**	**+7.0**	**+3.7**	**+1.3**	**+3.4**	**+8.8**	**+3.4**	**+9.1**
BERT	81.1	86.2	74.8	74.0	81.1	67.4	68.9	78.7	70.0
w. CRF	+1.1	+0.3	+0.8	+0.9	+0.9	+1.5	+2.1	+0.2	+2.8
w. CNN	-0.3	+0.6	+0.5	+0.0	+1.0	+1.2	+0.7	+0.3	+0.8
w. LSTM	+0.5	+0.4	+0.4	-0.3	0.0	+0.1	+2.0	+0.3	+1.0
w. *B.L.*	-0.6	-0.1	-0.5	-0.5	+0.7	+0.5	+0.7	-0.2	-0.6
w. *B.L.C.*	+1.4	+0.3	+1.2	**+1.4**	+0.9	+1.8	+2.9	+0.2	+2.4
w. Cofe	**+2.2**	**+0.9**	**+1.7**	+1.3	**+1.2**	**+2.0**	**+4.0**	**+1.0**	**+3.2**

cause CoreNLP uses quote marks to extract quotations. The number of direct quotations (*i.e.,* quoted *content*) on PolNeAR and Riqua is large, while the PoliticsZH is small. This shows that the rule-based methods cannot effectively identify indirect quotations.

The level of difficulty in extracting *source*, *cue*, and *content* is different. As a result, the performances of *source* and *cue* are better than the difficult *content*. This is expected because *content* is longer and complex in semantics. For example, the *content* may contain another *source*, *cue* and *content*. We design gate and attention mechanisms to fit those so that our model performs well.

4.5 Comparison with CRF and BERT

Comparison with CRF. CRF is a popular approach to handle sequence labeling problems, *e.g.,* NER (Ritter et al., 2011; Dong et al., 2016). We compare CofeNet with CRF by changing the encoder, *i.e., LSTM w. Cofe* denotes the Cofe using LSTM as text encoder. Recall that CofeNet specifically refers to the model using BERT as encoder, marked as *BERT w. Cofe* in Table 3. To make the comparison comprehensively and deeply, our comparisons between CRF and Cofe are based on various mainstream models including CNN, GRU, LSTM, BiLSTM, and BERT.

Table 3 details the comparison results on PolNeAR, and the results of the other two datasets are reported in the page[3]. (i) Results show that both Cofe and CRF perform better than basic models, and Cofe-based models perform better than CRF-based models. The comparison results suggest that our model architecture fits well with dependent sequence labeling tasks. As designed, the enhanced cell is capable of building the dependency relations of labels. (ii) Another interesting observation from the results is that if the basic model (*e.g.,* GRU) is simple, a larger improvement is achieved. On the contrary, the improvement over BERT is relatively small. It makes sense because the improvement is harder when the performance is already at a very high level. (iii) We also note that CofeNet performs better than CRF on all components of quotations.

Comparison with BERT. BERT based models are strong baselines for many tasks, particularly when there are clear patterns. The performance of models could be improved if we adopt a dependent encoding method based on BERT. To this end, based on BERT, we use decoders including CNN, LSTM, BiLSTM, BiLSTM+CRF in addition to CRF. The bottom area of Table 3 shows the results. Results show that the improvements of decoders including CNN, LSTM and BiLSTM are not significant than BiLSTM+CRF. Despite this, our CofeNet performs best. When meeting simple text span (*e.g.,*

[3] `https://thuwyq.github.io/docs/cofenet-detail-exp.pdf`

	B-source	I-source	B-cue	I-cue	B-content	I-content	O
<Start>	0.235	0.000	0.016	0.000	0.249	0.000	0.500
B-source	0.000	0.538	0.419	0.000	0.002	0.000	0.041
I-source	0.001	0.777	0.161	0.000	0.004	0.000	0.057
B-cue	0.054	0.000	0.000	0.380	0.342	0.000	0.225
I-cue	0.030	0.000	0.000	0.549	0.358	0.000	0.062
B-content	0.005	0.000	0.016	0.000	0.003	0.944	0.031
I-content	0.009	0.000	0.005	0.000	0.001	0.942	0.043
O	0.032	0.000	0.015	0.000	0.024	0.000	0.929

(a) The transition matrix of groundtruth

	B-source	I-source	B-cue	I-cue	B-content	I-content	O
<Start>	−0.006	0.000	−0.001	0.000	0.023	0.000	−0.016
B-source	0.000	0.022	−0.030	−0.001	0.002	0.000	0.006
I-source	−0.001	−0.006	−0.005	0.000	0.000	0.000	0.012
B-cue	0.006	0.000	0.000	−0.014	0.000	0.000	0.008
I-cue	0.003	0.000	0.000	0.025	−0.011	−0.003	−0.014
B-content	−0.001	0.000	0.002	0.000	0.002	−0.008	0.004
I-content	0.000	0.000	0.001	0.000	0.000	−0.001	0.000
O	0.000	0.000	0.003	0.000	0.003	0.000	−0.005

(b) The margin between groundtruth and CofeNet

Figure 3: The transition matrix and the margin of groundtruth and our model on PolNeAR.

Cue), the improvement of our proposed CofeNet is relatively small (1.3 point improvement, F1-Exact Match, on the Cue of PolNeAR dataset). When it comes to complex text span (*e.g.,* Content), our model shows large improvement over BERT model (4.0 points improvement, F1-Exact Match, on the Content of PolNeAR dataset).

From the comparisons, we demonstrate that our proposed CofeNet achieves the state-of-the-art performance on quotation extraction. To reveal the essence of CofeNet, we show the transition matrix of labels, the analysis on attention mechanism, and the ablation study in the next sections.

4.6 Label Transition Matrix

The probability transition matrix of labels reflects the particular features of *source*, *cue* and *content*. Thus we can use them to reveal the transition mechanism of labels. To this end, we calculate the label transition matrix of groundtruth, and the margin between groundtruth and CofeNet. Figure 3 depicts the detail on PolNeAR. In all subfigures, the column denotes the previous label and the row represents the current label. The value of Figure 3(a) denotes the transition probability of true labels, and the value of Figure 3(b) is the margin between the true and the predicted. As the word saying, "$\langle$Start$\rangle$" denotes the location before the first word, "B-" and "I-" denote the beginning and the inside of the *source*, *cue* and *content*, respectively. "O" refers to the other words.

The transition matrix of groundtruth shown in Figure 3(a) reveals the statistics of the PolNeAR dataset. Recall that the key for quotation extraction

Label		B-source	B-cue	I-cue	B-content	I-content	O
α_y		0.03	0.11	0.29	0.16	0.18	0.01
α_f		0.13	0.18	0.20	0.18	0.27	0.06
α_c		0.75	0.65	0.47	0.56	0.48	0.90
α_l		0.09	0.05	0.05	0.11	0.07	0.02
Word	<Start>	trump	has	denied	every	allegation	,

Label	O	B-cue	I-cue	B-content	I-content	I-content	I-content
α_y	0.03	0.17	0.28	0.15	0.10	0.10	0.13
α_f	0.06	0.17	0.21	0.19	0.31	0.31	0.27
α_c	0.89	0.61	0.45	0.57	0.48	0.51	0.49
α_l	0.02	0.05	0.05	0.10	0.11	0.08	0.12
Word	and	has	promised	to	fight	back	once

Label	I-content	I-content	I-content	I-content	O
α_y	0.13	0.12	0.13	0.14	0.01
α_f	0.27	0.27	0.26	0.27	0.05
α_c	0.51	0.53	0.53	0.52	0.92
α_l	0.09	0.07	0.08	0.07	0.02
Word	the	election	is	over	.

Figure 4: The attention weights of one test data from PolNeAR.

is the recognition of the "Begin". Hence, the margin of "Begin" is the compass for evaluating the performance. We find that the maximum absolute margin of "Begin" is -0.03, when the precious label is "B-source" and the current label is "B-cue". This is because the length of *source* is short, and *cue* word often follows *source* word closely. This proves that our model performs well even in difficult situations.

For BIO labeling scheme, the "I-source/cue/content" exists except the corresponding "B-*" exists. As a result, the transition value of "I-" could show the recognition ability of the model for those patterns. Also, Figure 3(b) shows almost all margins of those values are zeros. This reveals that our model could study those key patterns well.

4.7 Analysis on Attention Mechanism

In our design, the utilization of inflow information (*e.g.,* former labels, previous words, current word, and latter words) is the key for quotation extraction. Figure 4 shows the weights from the attention layer of one test instance in PolNeAR. To avoid the bias of a single case, we do a global prediction for all texts in the test dataset of PolNeAR attached in Appendix B. (i) The current word information has the largest weight, as expected. For the prediction of "I-source/cue/content", the former labels and former words information are the most important roles after the current word. It indicates that our model is capable of utilizing the former labels and sequence information as we designed. (ii) Another interesting observation is that the weights of the latter words' information for predicting "B/I-content" are about 0.1, which are greater than the other weights in α_l. As we mentioned before, the length of *content* is longer than *source* and *cue*, so the utilization of latter information improves the performance of long-span extraction more efficiently.

Table 4: Ablation study on PolNeAR dataset.

Model	Source			Cue			Content		
	$F1$-E.	$F1$-B.	J	$F1$-E.	$F1$-B.	J	$F1$-E.	$F1$-B.	J
CofeNet	83.2	87.1	76.4	75.3	82.3	69.4	72.9	79.6	73.2
w.o. g.m.	-1.0	-0.6	-0.9	-0.2	-0.2	-1.0	-0.8	-0.3	-1.2
w.o. a.m.	-0.9	-1.4	-1.5	-0.2	-1.0	-1.3	-1.2	-0.8	-1.3
w.o. f.l.	-2.4	-0.8	-1.5	-1.9	-0.5	-1.5	-2.5	-0.3	-2.7
w.o. f.w.	-0.9	-0.6	-1.1	-0.1	-0.3	-0.9	-1.3	-0.8	-1.1
w.o. c.w.	-2.0	-1.4	-2.0	-1.1	-1.0	-1.6	-1.4	-1.2	-1.2
w.o. l.w.	-1.0	-0.9	-1.2	-0.4	-0.4	-0.6	-1.7	-1.4	-1.0

4.8 Ablation Study

The CofeNet model uses gate mechanism $g.m.$ and attention mechanism $a.m.$ (see Section 3) to utilize information including former labels $f.l.$, former words $f.w.$, current word $c.w.$, and latter words $l.w.$. To study the effect of the two mechanisms and on the four information sources, we conduct ablation experiments on PolNeAR dataset.

Table 4 reports the results of this ablation study. (i) As expected, all mechanisms and information are useful for quotation extraction. For *content*, the Jaccard performance degrades at least 1.0 points after removing mechanisms or input information, which is similar to *source* and *cue*. As a comparison, the performance drop on $F1$-E. and $F1$-B. is significantly less than J. It is because the structure of *source* and *cue* is simpler than *content*. This phenomenon shows our CofeNet is particularly suitable for extracting quotations with long and complicated structures. (ii) When removing attention, larger drops on exact match are observed than removing gate. It reveals that attention is effective for begin match while gate prefers exact match. (iii) Further, we explore the performance of inflow information. The “w.o. f.w.” on Table 4 shows that the former words’ information is not so important for the prediction of *cue* because the *cue* is the shortest of all three components. The former label and the current word, the latter words are important for all of the components. It proves that the latter words’ information is key for the recognition of *content*. This fits with our observations in Section 4.7.

5 Conclusion and Future Work

In this study, we design the CofeNet model for quotation extraction with variable-length span and complicated structure. The key idea of CofeNet model is to use gate and attention mechanisms to control the important information including former labels, former words, current word and latter words at the element and vector levels. Experiments show that the proposed model achieves the state-of-the-art performance on two public datasets PolNeAR and Riqua and one proprietary dataset PoliticsZH.

For quotation analysis, the extraction of quotation components is the first step. In our study, we split a long text into short texts to ensure that one instance contains one *source*, one *cue* and one *content*. Thus the recognition of quotation triplets from long text (*e.g.,* across instance) is one important future work. Another important direction is to go deep into the nesting phenomenon, which makes the recognition harder.

Acknowledgments

This work was supported by the National Key R&D Program of China (2020AAA0105200) and the National Science Foundation of China (NSFC No. 62106249, 61902382, 61972381).

References

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. Leveraging linguistic structure for open domain information extraction. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 344–354, Beijing, China. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. BERT: pre-training of deep bidirectional transformers for language understanding. In 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, June 2-7, 2019, Volume 1 (Long and Short Papers), pages 4171–4186. Association for Computational Linguistics.

Chuanhai Dong, Jiajun Zhang, Chengqing Zong, Masanori Hattori, and Hui Di. 2016. Character-based LSTM-CRF with radical-level features for chinese named entity recognition. In Natural Language Understanding and Intelligent Applications - 5th CCF Conference on Natural Language Processing and Chinese Computing, NLPCC 2016, and 24th International Conference on Computer Processing of Oriental Languages, ICCPOL 2016, Kunming, China, December 2-6, 2016, Proceedings, volume 10102 of Lecture Notes in Computer Science, pages 239–250. Springer.

David K. Elson and Kathleen R. McKeown. 2010. Automatic attribution of quoted speech in literary narrative. In Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010. AAAI Press.

Peter Exner and Pierre Nugues. 2011. Using semantic role labeling to extract events from wikipedia. In Proceedings of the Workhop on Detection, Representation, and Exploitation of Events in the Semantic Web (DeRiVE 2011), Bonn, Germany, October 23, 2011, volume 779 of CEUR Workshop Proceedings, pages 38–47. CEUR-WS.org.

William Paulo Ducca Fernandes, Eduardo Motta, and Ruy Luiz Milidiú. 2011. Quotation extraction for Portuguese. In Proceedings of the 8th Brazilian Symposium in Information and Human Language Technology.

Kiril Gashteovski, Rainer Gemulla, and Luciano Del Corro. 2017. Minie: Minimizing facts in open information extraction. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017, pages 2630–2640. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. Neural Comput., 9(8):1735–1780.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.

Ralf Krestel, Sabine Bergler, and René Witte. 2008. Minding the source: Automatic tagging of reported speech in newspaper articles. In Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08), Marrakech, Morocco. European Language Resources Association (ELRA).

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001, pages 282–289. Morgan Kaufmann.

Yann LeCun, Yoshua Bengio, et al. 1995. Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks, 3361(10):1995.

Grace E. Lee and Aixin Sun. 2019. A study on agreement in PICO span annotations. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019, pages 1149–1152. ACM.

Kuan-Lin Lee, Yu-Chung Cheng, Pai-Lin Chen, and Hen-Hsen Huang. 2020. Keeping their words: Direct and indirect chinese quote attribution from newspapers. In Companion of The 2020 Web Conference 2020, Taipei, Taiwan, April 20-24, 2020, pages 98–99. ACM / IW3C2.

Xi Lin, Yequan Wang, Xiaokui Xiao, Zengxiang Li, and Sourav S. Bhowmick. 2019. Path travel time estimation using attribute-related hybrid trajectories network. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019, pages 1973–1982. ACM.

Xuying Meng, Yequan Wang, Runxin Ma, Haitong Luo, Xiang Li, and Yujun Zhang. 2022. Packet representation learning for traffic classification. In KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022, pages 3546–3554. ACM.

Edward Newell, Drew Margolin, and Derek Ruths. 2018. An attribution relations corpus for political news. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018. European Language Resources Association (ELRA).

Timothy O'Keefe, Silvia Pareti, James R. Curran, Irena Koprinska, and Matthew Honnibal. 2012. A sequence labelling approach to quote attribution. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2012, July 12-14, 2012, Jeju Island, Korea, pages 790–799. ACL.

Timothy O'Keefe, Silvia Pareti, James R. Curran, Irena Koprinska, and Matthew Honnibal. 2012. A sequence labelling approach to quote attribution. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 790–799, Jeju Island, Korea. Association for Computational Linguistics.

Sean Papay and Sebastian Padó. 2020. Riqua: A corpus of rich quotation annotation for english literary text. In Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020, pages 835–841. European Language Resources Association.

Silvia Pareti, Tim O'Keefe, Ioannis Konstas, James R. Curran, and Irena Koprinska. 2013. Automatically detecting and attributing indirect quotations. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 989–999, Seattle, Washington, USA. Association for Computational Linguistics.

Bruno Pouliquen, Ralf Steinberger, and Clive Best. 2007. Automatic detection of quotations in multilingual news. In Proceedings of Recent Advances in Natural Language Processing, pages 487–492.

Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. 2011. Named entity recognition in tweets: An experimental study. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 1524–1534, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Nathan Schneider, Rebecca Hwa, Philip Gianfortoni, Dipanjan Das, Michael Heilman, Alan W. Black, Frederick L. Crabbe, and Noah A. Smith. 2010. Visualizing topical quotations over time to understand news discourse.

Richard Socher, Cliff Chiung-Yu Lin, Andrew Y. Ng, and Christopher D. Manning. 2011. Parsing natural scenes and natural language with recursive neural networks. In Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011, pages 129–136. Omnipress.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 1422–1432, Lisbon, Portugal. Association for Computational Linguistics.

Thanh Vu, Dat Quoc Nguyen, Dai Quoc Nguyen, Mark Dras, and Mark Johnson. 2018. VnCoreNLP: A Vietnamese natural language processing toolkit. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations, pages 56–60, New Orleans, Louisiana. Association for Computational Linguistics.

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based LSTM for aspect-level sentiment classification. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 606–615, Austin, Texas. Association for Computational Linguistics.

Yequan Wang, Aixin Sun, Jialong Han, Ying Liu, and Xiaoyan Zhu. 2018. Sentiment analysis by capsules. In Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018, pages 1165–1174. ACM.

Yequan Wang, Aixin Sun, Minlie Huang, and Xiaoyan Zhu. 2019. Aspect-level sentiment analysis using as-capsules. In The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019, pages 2033–2044. ACM.

Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. 2018. Double embeddings and CNN-based sequence labeling for aspect extraction. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 592–598, Melbourne, Australia. Association for Computational Linguistics.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1480–1489, San Diego, California. Association for Computational Linguistics.

Appendix

A Implementation Details

We list the implementation details of CofeNet.

Table 5: CofeNet-BERT experimental configuration on PolNeAR, Riqua and PoliticsZH datasets. The sampling ratio is the value selection ratio of the former label during training. The three values represent the proportions of truth label, predict label and random label.

Training hyperparameters	
Optimizer	Adam
Learning rate except BERT	1e-3
Learning rate of BERT	5e-5
The hyperparameters of BERT	
Encoder layer	12
Attention head	12
Hidden size	768
Intermediate size	3,072
The hyperparameters of CofeNet	
Hidden size	100
Label embedding	100
Number of Former labels k	1
Number of Former words n	3
Number of Latter words m	3

Table 5 lists the same settings for the two public datasets (*i.e.,* PolNeAR and Riqua) and our proprietary dataset (*i.e.,* PoliticsZH). The learning rate for model parameters except BERT are $1e-3$, and $5e-5$ for BERT. We use typical 12-layers BERT (known as *bert-base-uncased* [4]) as a basic encoder for the two English datasets. For the Chinese dataset PoliticsZH, we use *bert-base-chinese* [5]. The middle part of Table 5 shows the important hyperparameters of BERT. There are other hyperparamters for CofeNet except BERT related. The hidden sizes of word representation and label embedding are 100. The number of former labels, former words, and latter words is 1, 3, and 3, respectively. The different hyperparameter for CofeNet is the batch size due to the GPU memory limitation. During training, we set the batch sizes for PolNeAR, Riqua and PoliticsZH to 15, 15 and 16, respectively.

We use Adam (Kingma and Ba, 2015) as our optimization method. CofeNet is implemented on Pytorch (version 1.2.0). NLTK is used to segment text. For BERT model, we invoke the pytorch-transformers package (version 1.2.0). To ensure the reliability of experimental results, we use the same transformer package with the same initialization parameters in BERT, BERT-CRF and CofeNet.

	B-source	I-source	B-cue	I-cue	B-content	I-content	O
<Start>	0.045	-	0.115	-	0.119	-	0.059
B-source	-	0.143	0.119	0.236	-	-	0.037
I-source	0.021	0.166	0.121	0.201	0.145	0.172	0.030
B-cue	0.044	-	-	0.244	0.152	-	0.035
I-cue	0.053	-	-	0.233	0.163	0.158	0.048
B-content	0.031	-	0.094	-	0.106	0.105	0.032
I-content	0.022	-	0.085	0.178	0.080	0.096	0.021
O	0.056	-	0.170	0.290	0.136	0.199	0.138

(a) The weight α_y for former labels r_i^y

	B-source	I-source	B-cue	I-cue	B-content	I-content	O
<Start>	0.134	-	0.165	-	0.190	-	0.093
B-source	-	0.171	0.194	0.201	-	-	0.095
I-source	0.090	0.148	0.170	0.172	0.166	0.214	0.070
B-cue	0.164	-	-	0.200	0.203	-	0.088
I-cue	0.158	-	-	0.201	0.201	0.252	0.117
B-content	0.148	-	0.198	-	0.200	0.295	0.107
I-content	0.150	-	0.198	0.236	0.180	0.297	0.078
O	0.113	-	0.159	0.177	0.151	0.203	0.110

(b) The weight α_f for former words r_i^f

	B-source	I-source	B-cue	I-cue	B-content	I-content	O
<Start>	0.720	-	0.651	-	0.600	-	0.794
B-source	-	0.638	0.639	0.522	-	-	0.829
I-source	0.792	0.622	0.653	0.568	0.596	0.532	0.864
B-cue	0.682	-	-	0.488	0.552	-	0.843
I-cue	0.690	-	-	0.488	0.530	0.486	0.779
B-content	0.714	-	0.602	-	0.586	0.488	0.809
I-content	0.731	-	0.650	0.488	0.632	0.527	0.869
O	0.754	-	0.620	0.473	0.637	0.512	0.705

(c) The weight α_c for current word r_i^c

	B-source	I-source	B-cue	I-cue	B-content	I-content	O
<Start>	0.100	-	0.069	-	0.091	-	0.053
B-source	-	0.048	0.048	0.041	-	-	0.039
I-source	0.097	0.064	0.057	0.059	0.093	0.082	0.035
B-cue	0.110	-	-	0.067	0.093	-	0.034
I-cue	0.099	-	-	0.077	0.107	0.103	0.055
B-content	0.107	-	0.106	-	0.108	0.112	0.052
I-content	0.097	-	0.067	0.097	0.108	0.080	0.032
O	0.077	-	0.052	0.061	0.076	0.086	0.048

(d) The weight α_l for latter words r_i^l

Figure 5: The weights for hidden states on PolNeAR.

B Global Analysis on Attention Mechanism

In our design, the utilization of inflow information is the key for quotation extraction. Recall that the information includes the former labels, the previous words, the current word and the latter words. Hence, we use the attention to reveal the operating principle of the model. Figure 4 has shown the weights from the attention layer of one individual case from test set of PolNeAR dataset. To avoid the bias of a single case, we do a global prediction for all texts in test set of PolNeAR shown in Figure 5. The observations from Figure 5 are similar to that reported in Section 4.7, so we will not repeat them.

[4] https://s3.amazonaws.com/models.huggingface.co/bert/bert-base-uncased-pytorch_model.bin

[5] https://s3.amazonaws.com/models.huggingface.co/bert/bert-base-chinese-pytorch_model.bin