An Effective Transition-based Model for Discontinuous NER

Xiang Dai[1,2] **Sarvnaz Karimi**[1] **Ben Hachey**[3] **Cecile Paris**[1]
[1]CSIRO Data61, Sydney, Australia
[2]University of Sydney, Sydney, Australia
[3]Harrison.ai, Sydney, Australia
{dai.dai,sarvnaz.karimi,cecile.paris}@csiro.au
ben.hachey@gmail.com

Task

- Discontinuous mentions

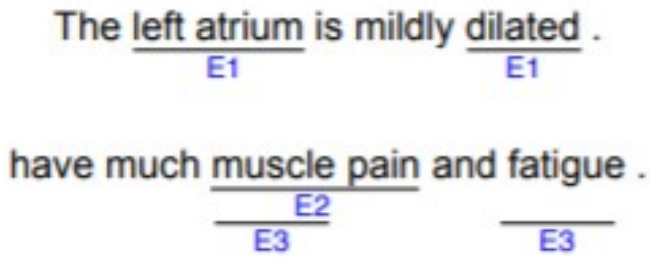

Figure 1: Examples involving discontinuous mentions, taken from the ShARe 13 (Pradhan et al., 2013) and CADEC (Karimi et al., 2015a) data sets, respectively. The first example contains a discontinuous mention *'left atrium dilated'*, the second example contains two mentions that overlap: *'muscle pain'* and *'muscle fatigue'* (discontinuous).

Model

- SHIFT moves the first token from the buffer to the stack; it implies this token is part of an entity mention.
- OUT pops the first token of the buffer, indicating it does not belong to any mention.
- COMPLETE pops the top span of the stack, outputting it as an entity mention. If we are interested in multiple entity types, we can extend this action to COMPLETE-y which labels the mention with entity type y.
- REDUCE pops the top two spans s_0 and s_1 from the stack and concatenates them as a new span which is then pushed back to the stack.
- LEFT-REDUCE is similar to the REDUCE action, except that the span s_1 is kept in the stack. This action indicates the span s_1 is involved in multiple mentions. In other words, several mentions share s_1 which could be a single token or several tokens.
- RIGHT-REDUCE is the same as LEFT-REDUCE, except that s_0 is kept in the stack.

Model

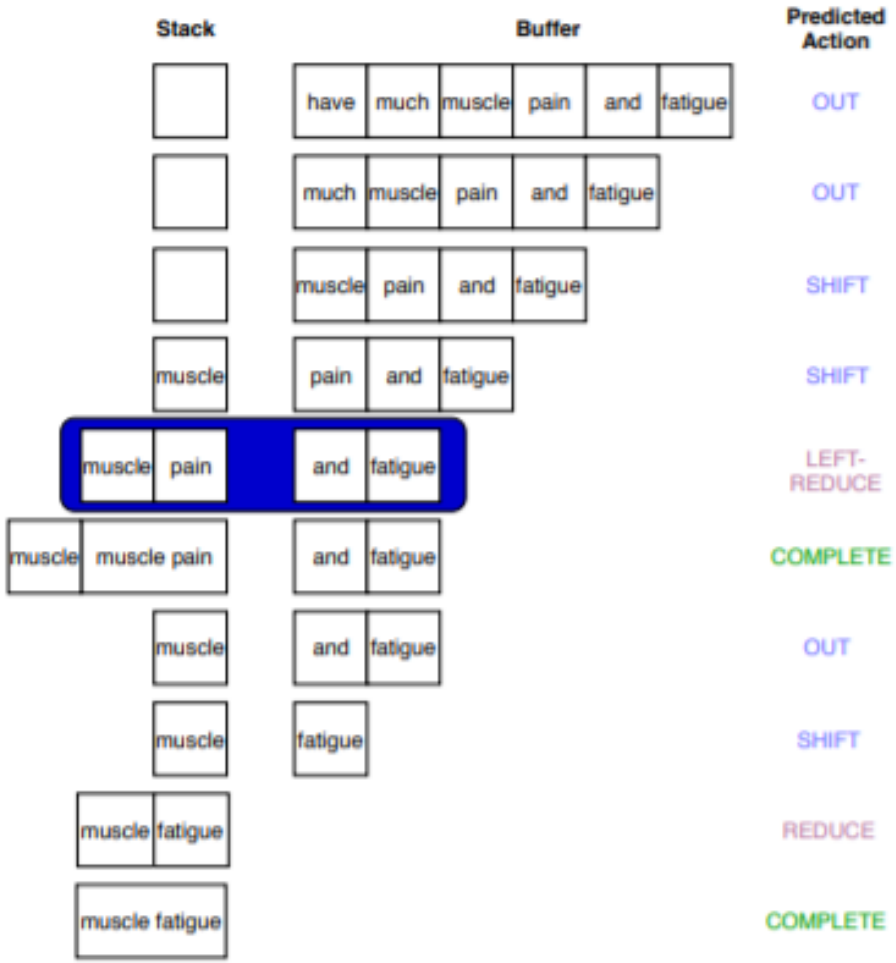

Figure 3: An example sequence of transitions. Given the states of stack and buffer (blue highlighted), as well as the previous actions, predict the next action (i.e., LEFT-REDUCE) which is then applied to change the states of stack and buffer.

Representation and Stack-lstm

ically, for the i-th token in the sequence, its representation can be denoted as:

$$\tilde{\mathbf{c}}_\mathbf{i} = \left[\overrightarrow{\text{LSTM}}(\mathbf{t_0}, \ldots, \mathbf{t_i}); \overleftarrow{\text{LSTM}}(\mathbf{t_i}, \ldots, \mathbf{t_{N-1}})\right],$$

where $\mathbf{t_i}$ is the concatenation of the embeddings for the i-th token, its character level representation learned using a CNN network (Ma and Hovy, 2016). Pretrained contextual word representations have shown its usefulness on improving various NLP tasks. Here, we can also concatenate pretrained contextual word representations using ELMo (Peters et al., 2018) with $\tilde{\mathbf{c}}_\mathbf{i}$, resulting in:

$$\mathbf{c_i} = [\tilde{\mathbf{c}}_\mathbf{i}; \text{ELMo}_i], \quad (1)$$

where ELMo_i is the output representation of pretrained ELMo models (frozen) for the i-th token.

Following the work in (Dyer et al., 2015), we use Stack-LSTM to represent spans in the stack. That is, if a token is moved from the buffer to the stack, its representation is learned using:

$$\mathbf{s_0} = \text{Stack-LSTM}(\mathbf{s_D} \ldots \mathbf{s_1}; \mathbf{c}_{\text{SHIFT}}),$$

where D is the number of spans in the stack. Once REDUCE related actions are applied, we use a multi-layer perceptron to learn the representation of the concatenated span. For example, the REDUCE action takes the representation of the top two spans in the stack: $\mathbf{s_0}$ and $\mathbf{s_1}$, and produces a new span representation:

$$\tilde{\mathbf{s}} = \mathbf{W}^T[\mathbf{s_0}; \mathbf{s_1}] + b,$$

where $\mathbf{W}$ and b denote the parameters for the composition function. The new span representation $\tilde{\mathbf{s}}$ is pushed back to the stack to replace the original two spans: $\mathbf{s_0}$ and $\mathbf{s_1}$.

Stack-lastm， Action selection

3.2 Capturing Discontinuous Dependencies

We hypothesize that the interactions between spans in the stack and tokens in the buffer are important factors in recognizing discontinuous mentions. Considering the example in Figure 3, a span in the stack (e.g., *'muscle'*) may need to combine with a future token in the buffer (e.g., *'fatigue'*). To capture this interaction, we use multiplicative attention (Luong et al., 2015) to let the span in the stack $\mathbf{s_i}$ learn which token in the buffer to attend, and thus a weighted sum of the representation of tokens in the buffer $\mathbf{B}$:

$$\mathbf{s_i^a} = \mathbf{softmax}(\mathbf{s_i^T W_i^a B})\mathbf{B}. \quad (2)$$

We use distinct $\mathbf{W_i^a}$ for $\mathbf{s_i}$ separately.

3.3 Selecting an Action

Finally, we build the parser representation as the concatenation of the representation of top three spans from the stack $(\mathbf{s_0}, \mathbf{s_1}, \mathbf{s_2})$ and its attended representation $(\mathbf{s_0^a}, \mathbf{s_1^a}, \mathbf{s_2^a})$, as well as the representation of the previous action $\mathbf{a}$, which is learned using a simple unidirectional LSTM. If there are less than 3 spans in the stack or no previous action, we use randomly initialized vectors $\mathbf{s_{empty}}$ or $\mathbf{a_{empty}}$ to replace the corresponding vector. This parser representation is used as input for the final softmax prediction layer to select the next action.

Dataset

	CADEC	ShARe 13	ShARe 14
Text type	online posts	clinical notes	clinical notes
Entity type	ADE	Disorder	Disorder
# Documents	1,250	298	433
# Tokens	121K	264K	494K
# Sentences	7,597	18,767	34,618
# Mentions	6,318	11,161	19,131
# Disc.M	675 (10.6)	1,090 (9.7)	1,710 (8.9)
Avg mention L.	2.7	1.8	1.7
Avg Disc.M L.	3.5	2.6	2.5
Avg interval L.	3.3	3.0	3.2
Discontinuous Mentions			
2 components	650 (95.7)	1,026 (94.3)	1,574 (95.3)
3 components	27 (3.9)	62 (5.6)	76 (4.6)
4 components	2 (0.2)	0 (0.0)	0 (0.0)
No overlap	82 (12.0)	582 (53.4)	820 (49.6)
Overlap at left	351 (51.6)	376 (34.5)	616 (37.3)
Overlap at right	152 (22.3)	102 (9.3)	170 (10.3)
Multiple overlaps	94 (13.8)	28 (2.5)	44 (2.6)
Continuous Mentions			
Overlap	326 (5.7)	157 (1.5)	228 (1.3)

Table 1: The descriptive statistics of the data sets. ADE: adverse drug events; Disc.M: discontinuous mentions; Disc.M L.: discontinuous mention length, where intervals are not counted. Numbers in parentheses are the percentage of each category.

Baseline

Flat model To train the flat model on our data sets, we use an off-the-shelf framework: Flair (Akbik et al., 2018), which achieves the state-of-the-art performance on CoNLL 03 data set. Recall that the flat model cannot be directly applied to data sets containing discontinuous mentions. Following the practice in (Stanovsky et al., 2017), we replace the discontinuous mention with the shortest span that fully covers it, and merge overlapping mentions into a single mention that covers both. Note that, different from (Stanovsky et al., 2017), we apply these changes only on the training set, but not on the development set and the test set.

BIO extension model The original implementation in (Metke-Jimenez and Karimi, 2016) used a CRF model with manually designed features. We report their results on CADEC in Table 2 and re-implement a BiLSTM-CRF-ELMo model using their tag schema (denoted as 'BIO Extension' in Table 2).

Graph-based model The original paper of (Muis and Lu, 2016) only reported the evaluation results on sentences which contain at least one discontinuous mention. We use their implementation to train the model and report evaluation results on the whole test set (denoted as 'Graph' in Table 2). We argue that it is important to see how a discontinuous NER model works not only on the discontinuous mentions but also on all the mentions, especially since, in real data sets, the ratio of discontinuous mentions cannot be made a priori.

Experiment

Model	CADEC			ShARe 13			ShARe 14		
	P	R	F	P	R	F	P	R	F
(Metke-Jimenez and Karimi, 2016)	64.4	56.5	60.2	–	–	–	–	–	–
(Tang et al., 2018)	67.8	64.9	66.3	–	–	–	–	–	–
(Tang et al., 2013b)	–	–	–	80.0	70.6	75.0	–	–	–
Flat	65.3	58.5	61.8	78.5	66.6	72.0	76.2	76.7	76.5
BIO Extension	68.7	66.1	67.4	77.0	72.9	74.9	74.9	78.5	76.6
Graph	**72.1**	48.4	58.0	**83.9**	60.4	70.3	**79.1**	70.7	74.7
Ours	68.9	**69.0**	**69.0**	80.5	**75.0**	**77.7**	78.1	**81.2**	**79.6**

Table 2: Evaluation results on the whole test set in terms of precision, recall and F_1 score. The original ShARe 14 task focuses on template filling of disorder attributes: that is, given a disorder mention, recognize the attribute from its context. In this work, we use its mention annotations and frame the task as a discontinuous NER task.

Experiment

	Sentences with discontinuous mentions									Discontinuous mentions only								
	CADEC			ShARe 13			ShARe 14			CADEC			ShARe 13			ShARe 14		
Model	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
Flat	50.2	36.7	42.4	43.5	28.1	34.2	41.5	31.9	36.0	0	0	0	0	0	0	0	0	0
BIO E.	63.8	52.0	57.3	51.8	39.5	44.8	37.5	38.4	37.9	5.8	1.0	1.8	39.7	12.3	18.8	8.8	4.5	6.0
Graph	**69.5**	43.2	53.3	**82.3**	47.4	60.2	60.0	52.8	56.2	**60.8**	14.8	23.9	78.4	36.6	50.0	42.7	39.5	41.1
Ours	66.5	**64.3**	**65.4**	70.5	**56.8**	**62.9**	**61.9**	**64.5**	**63.1**	41.2	**35.1**	**37.9**	**78.5**	**39.4**	**52.5**	**56.1**	**43.8**	**49.2**

Table 3: Evaluation results on sentences that contain at least one discontinuous mention (left part) and on discontinuous mentions only (right part).

Experiment

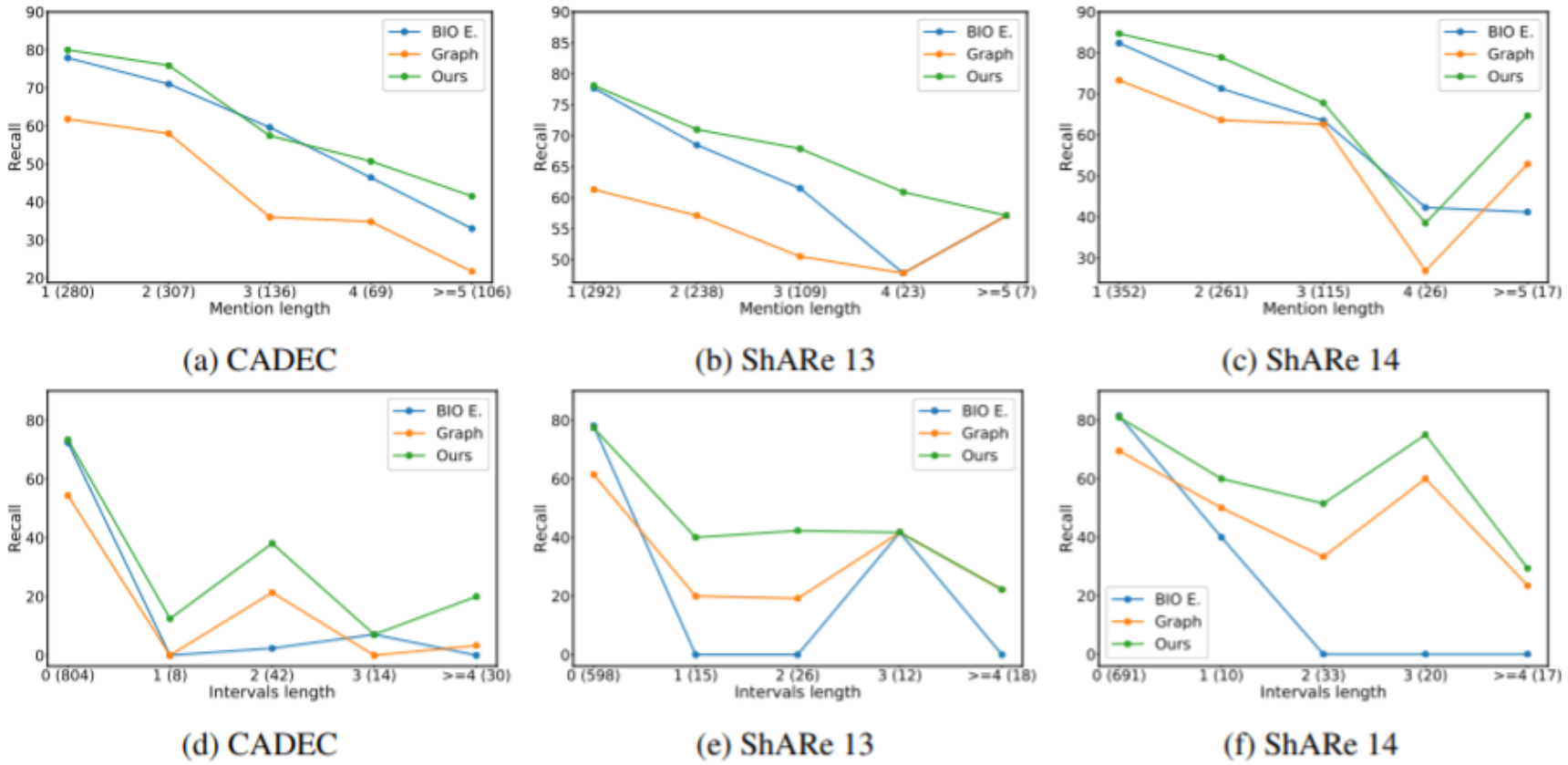

Figure 4: The impact of mention length and interval length on recall. Mentions with interval length of zero are continuous mentions. Numbers in parentheses are the number of gold mentions.

Experiment

		CADEC		ShARe 13		ShARe 14	
	Model	#	F	#	F	#	F
No 𝕆	BIO E.		0.0		7.5		0.0
	Graph	9	0.0	41	32.1	39	45.2
	Ours		0.0		**36.1**		**57.1**
Left 𝕆	BIO E.		6.0		25.0		15.7
	Graph	54	9.2	11	**45.5**	30	37.7
	Ours		**28.6**		33.3		**49.2**
Right 𝕆	BIO E.		0.0		0.0		0.0
	Graph	16	**45.2**	19	**21.4**	5	0.0
	Ours		29.3		13.3		0.0
Multi 𝕆	BIO E.		0.0		–		0.0
	Graph	15	0.0	0	–	6	0.0
	Ours		0.0		–		0.0

Table 4: Evaluation results on different categories of discontinuous mentions. '#' columns show the number of gold discontinuous mentions in development set of each category. 𝕆: overlap.