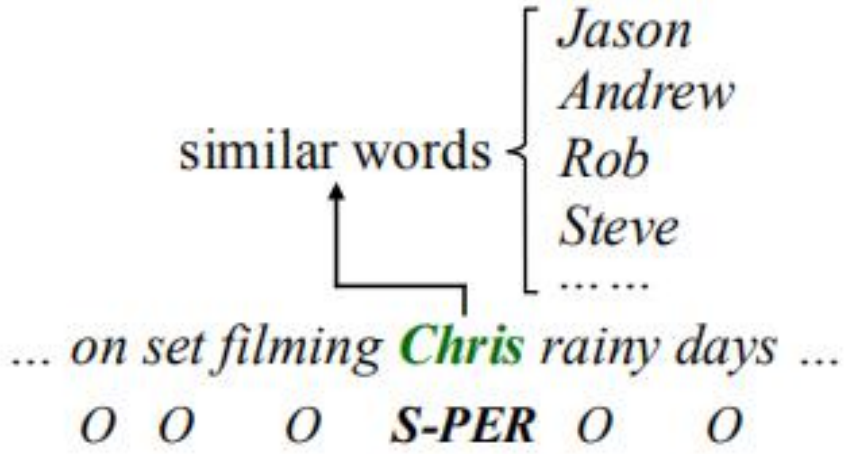
Named Entity Recognition for Social Media Texts with Semantic Augmentation

Yuyang Nie[♢*], **Yuanhe Tian**[♥*], **Xiang Wan**[♡], **Yan Song**[♠♡†], **Bo Dai**[♢]
[♢]University of Electronic Science and Technology of China
[♥]University of Washington [♡]Shenzhen Research Institute of Big Data
[♠]The Chinese University of Hong Kong (Shenzhen)
[♢]nyy207@gmail.com [♥]yhtian@uw.edu [♡]wanxiang@sribd.cn
[♠]songyan@cuhk.edu.cn [♢]daibo@uestc.edu.cn

Problem

- NER： Data sparity; do not follow strict syntactic rules
- Domain information & Semantic augmentation



Summary

- Proposed an effective approach to NER for social media texts with **semantic augmentation**.

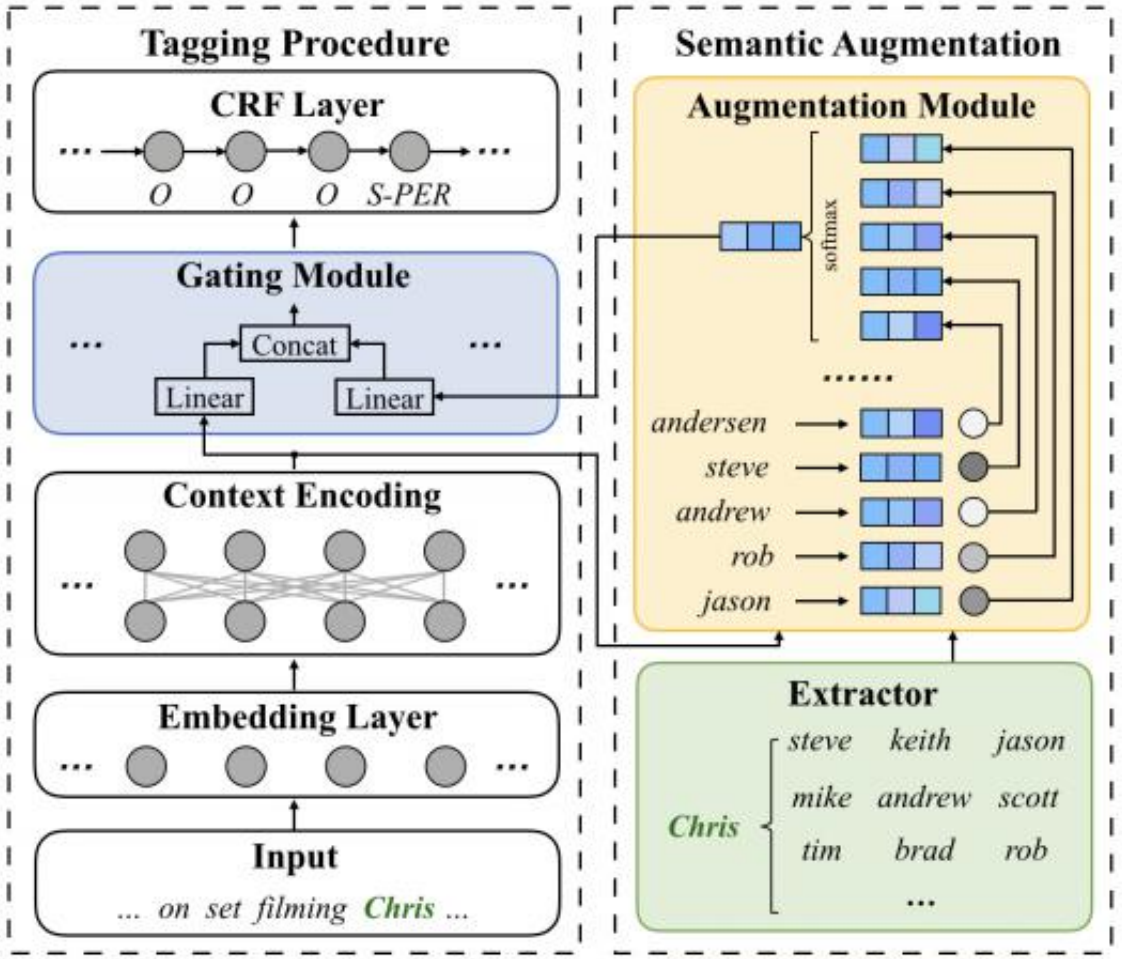


Figure 2: The overall architecture of our proposed model with semantic augmentation. An example sentence and its output NE labels are given, where the augmented semantic information for the word "*Chris*" are also illustrated with the processing through the augmentation module and the gate module.

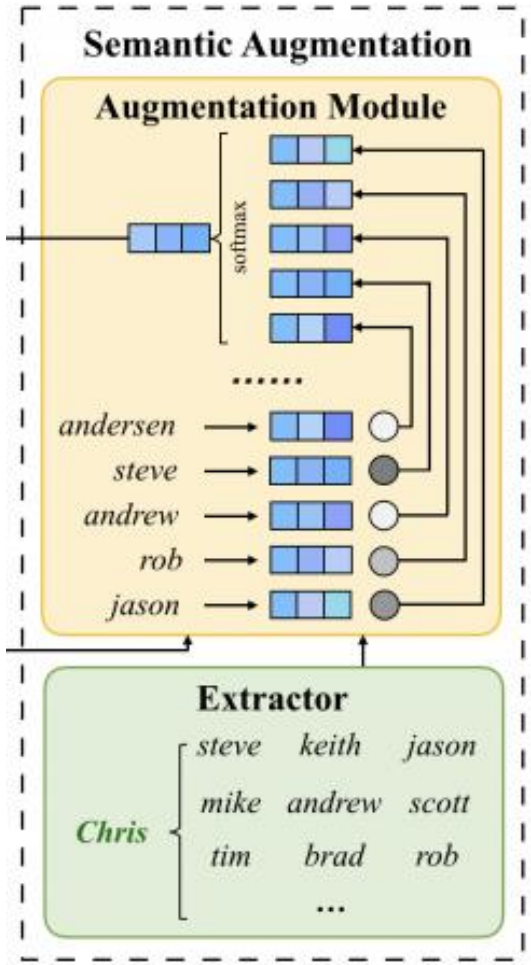
Attentive Semantic Augmentation (AU)

$$\mathcal{X} = x_1, x_2, \cdots, x_n$$

$$C_i = \{c_{i,1}, c_{i,2}, \cdots, c_{i,j}, \cdots, c_{i,m}\} \qquad (1)$$

$$p_{i,j} = \frac{exp(\mathbf{h}_i \cdot \mathbf{e}_{i,j})}{\sum_{j=i}^{m} exp(\mathbf{h}_i \cdot \mathbf{e}_{i,j})},$$

$$\mathbf{v}_i = \sum_{j=1}^{m} p_{i,j} \mathbf{e}_{i,j},$$



The Gate Module

$$\mathbf{g} = \sigma(\mathbf{W}_1 \cdot \mathbf{h}_i + \mathbf{W}_2 \cdot \mathbf{v}_i + \mathbf{b}_g), \quad (4)$$

where $\mathbf{W}_1$ and $\mathbf{W}_2$ are trainable matrices and $\mathbf{b}_g$ the corresponding bias term. Afterwards, we use

$$\mathbf{u}_i = [\mathbf{g} \circ \mathbf{h}_i] \oplus [(\mathbf{1} - \mathbf{g}) \circ \mathbf{v}_i] \quad (5)$$

to balance the information from context encoder and the augmentation module, where $\mathbf{u}_i$ is the derived output of the gate module; $\circ$ represents the element-wise multiplication operation and $\mathbf{1}$ is a 1-vector with its all elements equal to 1.

Tagging Procedure

$$\mathbf{H} = CE(\mathbf{E}), \quad (6)$$

$$\mathbf{e}_i = \mathbf{e}_i^1 \oplus \mathbf{e}_i^2 \oplus \ldots \oplus \mathbf{e}_i^T, \quad (7)$$

$$\mathbf{o}_i = \mathbf{W}_u \cdot \mathbf{u}_i$$

$$\widehat{y}_i = \arg\max_{y_i \in L} \frac{exp(\mathbf{W}_c \cdot \mathbf{o}_i + \mathbf{b}_c)}{\sum_{y_{i-1} y_i} exp(\mathbf{W}_c \cdot \mathbf{o}_i + \mathbf{b}_c)}, \quad (8)$$

Experiments

- WNUT16 (W16)
- WNUT17 (W17)
- Weibo (WB)

Language	Dataset		Train	Dev	Test
English	W16	#Sent.	2,394	1,000	3,850
		#Ent.	1,496	661	3,473
		%Uns.	-	52.1	80.0
	W17	#Sent.	3,394	1,008	1,287
		#Ent.	1,975	835	1,079
		%Uns.	-	34.8	84.5
Chinese	WB	#Sent.	1,350	270	270
		#Ent.	1,885	389	414
		%Uns.	-	51.4	45.2

Table 1: The statistics of all benchmark datasets w.r.t. the number of sentences (# Sent.), named entities (# Ent.) and the percentage of unseen entities (% Uns.).

ID	*SE*	*GA*	W16	W17	WB
1	*N*	*N*	54.79	48.41	65.36
2	*DS*	*N*	55.03	48.36	65.01
3	*DS*	*Y*	56.28	48.98	66.24
4	*AU*	*N*	56.86	49.26	68.21
5	*AU*	*Y*	**57.94**	**50.02**	**69.32**

(a) Development Set

ID	*SE*	*GA*	W16	W17	WB
1	*N*	*N*	52.98	48.82	66.02
2	*DS*	*N*	53.11	48.71	65.78
3	*DS*	*Y*	54.02	49.56	67.52
4	*AU*	*N*	54.29	49.81	68.46
5	*AU*	*Y*	**55.01**	**50.36**	**69.80**

(b) Test Set

Model	W16	W17	WB
Zhang and Yang (2018)	-	-	58.79
Yan et al. (2019)	54.06	48.98	65.03
Zhu and Wang (2019)	-	-	59.31
Gui et al. (2019)	-	-	59.92
Sui et al. (2019)	-	-	63.09
Akbik et al. (2019)	-	49.59	-
Zhou et al. (2019)	53.43	42.83	-
Devlin et al. (2019)	54.36	49.52	67.33
Meng et al. (2019)	-	-	67.60
Xu et al. (2019)	-	-	68.93
Ours	**55.01**	**50.36**	**69.80**

Table 3: Comparison of $F1$ scores of our best performing model (the full model with augmentation module and gate module) with that reported in previous studies on all English and Chinese social media datasets.

Model	W16	W17	WB
# of Unseen NEs	2778	912	189
*Devlin et al. (2019)	49.02	46.73	45.98
*Yan et al. (2019)	48.97	46.89	45.71
Baseline	49.04	46.72	45.79
Ours (+AU +GA)	**51.27**	**49.45**	**48.81**

Table 4: The recall of our models with and without the attentive semantic augmentation (AU) and the gate module (GA) on unseen named entities (whose numbers are also reported at the first row) on all three datasets. The results of our runs of previous models (marked with "*") are also reported for references.

Biomedical Event Extraction as Sequence Labeling

Alan Ramponi[♢♣] **Rob van der Goot**[♠] **Rosario Lombardo**[♣] **Barbara Plank**[♠]
[♢]Department of Information Engineering and Computer Science, University of Trento, Italy
[♠]Department of Computer Science, IT University of Copenhagen, Denmark
[♣]Fondazione the Microsoft Research – University of Trento
Centre for Computational and Systems Biology (COSBI), Italy
ramponi@cosbi.eu, robv@itu.dk, lombardo@cosbi.eu, bapl@itu.dk

Related

- Each event consists of an event mention(trigger) and one or more arguments.
- Proposed a new approach for biomedical event extraction by casting it as a sequence labeling task (BEESL)

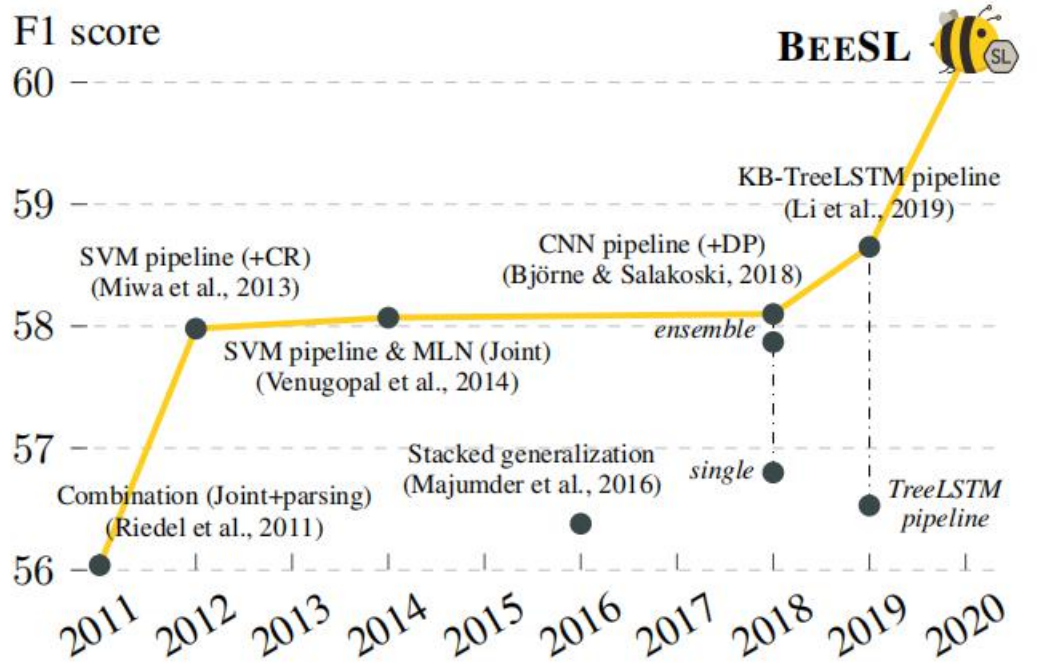
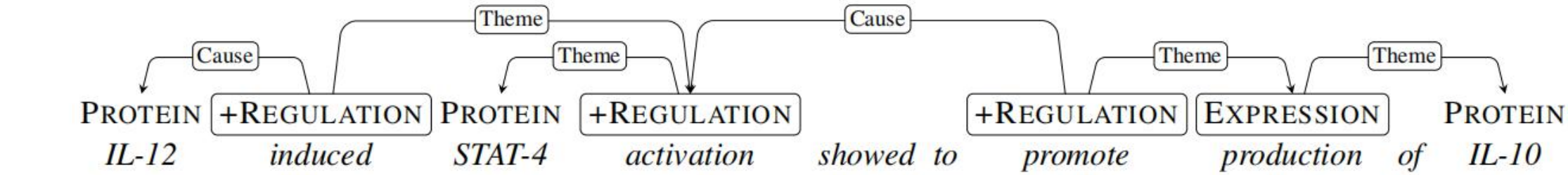


Figure 1: Performance of biomedical event extraction on the BioNLP Genia 2011 test set over time.

Summary

- Convert the event structures into a representation suitable for sequence labeling.
- Leverage a multi-label aware decoder with BERT in a multi-task sequence labeling model.
- SOTA and faster

Event structures



d:	Protein	+Regulation	Protein	+Regulation	+Regulation	Expression	Protein
r:	Cause		Theme	Theme, Cause		Theme	Theme
h:	$+\text{Reg}_{+1}$		$+\text{Reg}_{+1}$	$+\text{Reg}_{-1}$, $+\text{Reg}_{+1}$		$+\text{Reg}_{-1}$	Exp_{-1}

Event type	Arguments
Simple events	
Gene expression	Theme(P)
Transcription	Theme(P)
Protein catabolism	Theme(P)
Phosphorylation	Theme(P)
Localization	Theme(P)
Binding	Theme(P)+
Complex events	
Regulation	Theme(P/E), Cause(P/E)
Positive regulation	Theme(P/E), Cause(P/E)
Negative regulation	Theme(P/E), Cause(P/E)

Sequence labeling encoding

Given $[x_1, ..., x_n]$ a sequence of n tokens

Event structures as token-level labels $[y_1, ..., y_n]$

label y_i for each token x_i-->a tuple <*d, r, h*>

Given $[x_1, ..., x_n]$ a sequence of n tokens, we encode event structures as token-level labels $[y_1, ..., y_n]$, to reduce the task to a sequence labeling problem. Adopting dependency parsing terminology, we encode the label y_i for each token x_i as a tuple $\langle d, r, h \rangle$, where d is the *dependent* and refers to the token and its mention type (either trigger, entity, or nothing), r is the *relation* and used to refer to its role, and *head* (h) denotes the event the token refers to (Figure 2, bottom). In more detail, to discriminate event heads with the same type in text, we encode the heads h as *relative head mention position*.[2] For instance, $h = +\text{REG}_{+1}$ means the head is the first +REGULATION on the right of d in the relative surface order, whereas $h = +\text{REG}_{-2}$ means it is the second +REGULATION on the left. In Figure 2 the label for "production" is $\langle$EXPRESSION, THEME, $+\text{REG}_{-1}\rangle$, denoting the token is an EXPRESSION trigger, THEME of the first +REGULATION event on the left.

Event Extraction as Sequence Labeling

- Bert Encoder (mask entity span)
- Multi-task learning (MTL)
- Multi-label decoder

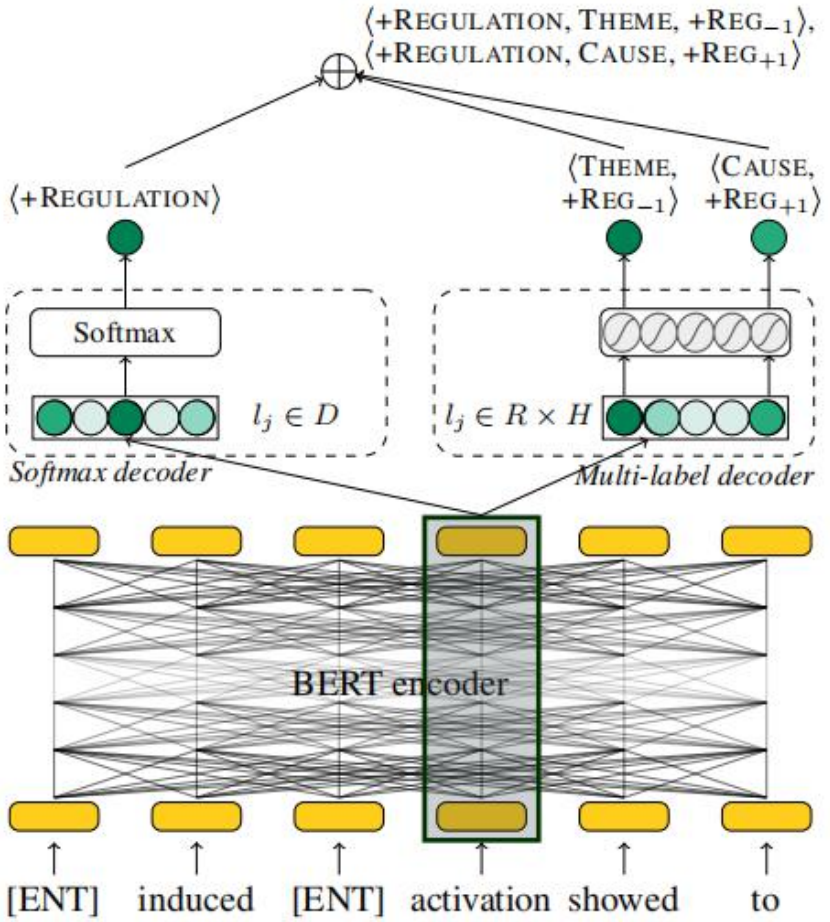


Figure 3: BEESL uses a multi-task multi-label model, using a BERT encoder with layer attention, and dedicated decoders for predicting the labels for each label sub-space, which are trivially merged.

Multi-task strategies

the label spaces for each component of the labels as $d_i \in D$, $r_i \in R$, and $h_i \in H$

Single-task A single-task (ST) setup is used as a baseline. It predicts a single label $y_i = \langle d, r, h \rangle$ for each input token x_i. The label space is up to $\mathcal{L} = |D| \times |R| \times |H|$.

Multi-task The label y_i for each token x_i is decomposed into parts (hereafter, sub-labels), each treated as a prediction task. The decomposition of the label space allows each sub-label space to be framed as a different task with its own private decoder, mitigating the output space sparsity (Vilares et al., 2019). Depending on the decomposition of the label $y_i = \langle d, r, h \rangle$, we have four multi-task learning options (pairs of tasks, or each subpart as a task, respectively) with the following properties:

1. $\langle d \rangle, \langle r, h \rangle$: up to $\mathcal{L} = |D| + |R| \times |H|$;
2. $\langle d, r \rangle, \langle h \rangle$: up to $\mathcal{L} = |D| \times |R| + |H|$;
3. $\langle d, h \rangle, \langle r \rangle$: up to $\mathcal{L} = |D| \times |H| + |R|$;
4. $\langle d \rangle, \langle r \rangle, \langle h \rangle$: up to $\mathcal{L} = |D| + |R| + |H|$.

Multi-label decoder

The multi-label decoder is designed to handle multiple labels per token, thus being suitable for predicting relations and heads. Given a task with $l_j \in L$ labels, it models $P(l_j|e_i)$ for each label l_j. Differently from the single-label decoder, each label is predicted with a sigmoid, where all contribute equally to the loss. Given the probabilities $P(l_j|e_i)$ for the $l_j \in L$ labels and a threshold τ, the token x_i is assigned all the labels l_j with probability $P(l_j|e_i) \geq \tau$. If no $P(l_j|e_i) \geq \tau$ is found, we take the highest scoring label l_j (which may also be empty) as a fallback.[3] We employ a binary cross-entropy loss, averaged across all batches.

Experiments

- Genia 2011 (Kim et al., 2011)
- Comprises both abstracts and full-texts. The corpus consists of annotations for **PROTEIN** entities and **9 fine-grained event types**. The Genia event extraction tasks expect both **texts and entities** as input, and complete **events** need to be predicted.

Item	Train	Dev	Test
Documents	908	259	347
Sentences	8,664	2,888	3,363
Tokens	230,737	74,334	90,091
Entities	11,625	4,690	5,301
Events	10,310	3,250	4,487

Work	Method	P	R	F1
Riedel et al. (2011)	FAUST – Model combination (joint+parsing)	64.75	49.41	56.04
Miwa et al. (2012)	EventMine – SVM pipeline (+coref)	63.48	53.35	57.98
Venugopal et al. (2014)	BioMLN – SVM pipeline & MLN (joint)	63.61	53.42	58.07
Majumder et al. (2016)	Stacked generalization	66.46	48.96	56.38
Björne and Salakoski (2018)	TEES – CNN pipeline (single model)	64.86	50.53	56.80
Björne and Salakoski (2018)	TEES – CNN pipeline (5x ensemble)	68.76	49.97	57.87
Björne and Salakoski (2018)*	TEES – CNN pipeline (mixed 5x ensemble)	69.45	49.94	58.10
Li et al. (2019)	BiLSTM pipeline	62.18	48.44	54.46
Li et al. (2019)	Tree-LSTM pipeline	64.56	50.28	56.53
Li et al. (2019)	KB-driven Tree-LSTM pipeline	67.01	52.14	58.65
BEESL	Multi-task neural sequence labeling	69.72	53.00	**60.22**

Table 2: Performance comparison on the test set of BioNLP Genia 2011. *indicates that the system was trained on training plus part of development data. BEESL uses the official training portion only. Top: traditional ML systems; Middle: state-of-the-art neural systems; Bottom: proposed multi-task sequence labeling system.

Multi-task	P	R	F1
$\langle d \rangle, \langle r, h \rangle$	71.28	55.44	**62.37**
$\langle d, r \rangle, \langle h \rangle$	72.35	51.31	60.04
$\langle d, h \rangle, \langle r \rangle$	73.51	49.49	59.16
$\langle d \rangle, \langle r \rangle, \langle h \rangle$	73.05	51.34	60.30
Multi-label	**P**	**R**	**F1**
BEESL$_{ST}$	73.30	52.42	61.13
with multi-label	71.74	56.71	63.34
BEESL$_{MT}$	71.28	55.44	62.37
with multi-label	71.84	59.42	**65.04**

Table 3: Performance of diverse settings for BEESL (multi-task and multi-label) on the development set.

Event type	BEESL	KBTL
Simple events	**79.31**	78.73
Gene expression	**80.90**	80.28
Transcription	69.46	**75.39**
Protein catabolism	**74.07**	60.87
Phosphorylation	**89.52**	84.36
Localization	**69.51**	68.47
Binding	**50.19**	44.10
Complex events	**48.32**	47.72
Regulation	**45.90**	43.52
Positive regulation	**49.41**	48.26
Negative regulation	47.17	**49.02**
All events	**60.22**	58.65

Table 4: Per-event performance of BEESL and KBTL (KB-driven TreeLSTM) (Li et al., 2019) on the test set.