Treasures Outside Contexts: Improving Event Detection via Global Statistics

Rui Li, Wenlin Zhao, Cheng Yang*, Sen Su
Beijing University of Posts and Telecommunications, Beijing, China
{lirui,zhaowenlin,yangcheng,susen}@bupt.edu.cn

Abstract

Event detection (ED) aims at identifying event instances of specified types in given texts, which has been formalized as a sequence labeling task. As far as we know, existing neural-based ED models make decisions relying on the contextual semantic features of each word in the input text, which we find is easy to get confused by varied contexts in the test stage. To this end, we come up with the idea of introducing a set of statistical features from word-event co-occurrence frequencies in the entire training set to cooperate with the contextual features. Specifically, we propose a **S**emantic and **S**tatistic-**J**oint **D**iscriminative **N**etwork **(S^2-JDN)** consisting of a semantic feature extractor, a statistical feature extractor, and a joint event discriminator. In experiments, S^2-JDN effectively exceeds ten recent state-of-the-art (SOTA) baseline methods on ACE2005 and KBP2015 benchmark datasets. Further, we perform extensive experiments to investigate the effectiveness of S^2-JDN.

1 Introduction

Event detection (ED) is an important information extraction task in the NLP field, which aims to identify event instances of specified types in given texts. Associated with each event mention is a phrase, i.e., the event trigger[1] evoking that event. More precisely, the task involves identifying event triggers and classifying them into the specific types. For instance, according to ACE2005 annotation guideline, in the sentence "A police officer was killed in New Jersey today", an ED model should be able to recognize the word "killed" as a trigger for the event type "Die".

With the development of deep learning, ED has been formalized as a sequence labeling task and implemented by a variety of neural network models (Chen et al., 2015; Zhao et al., 2018). As claimed by the famous distributional hypothesis (Harris, 1954; Firth, 1957), the words in a text are characterized by their contexts, i.e., themselves and their surrounding words. Thus, to fully consider the information of each word in the sequential prediction process, existing neural-based ED models make decisions based on elaborately extracted contextual features of the words (Hong et al., 2018; Tong et al., 2020b,a), which has become a "standard" decision-making pattern.

Two texts in the training set:	Trigger	Event
T1: There have been too many civilians killed ···	Killed	Die
T2: ··· a sister were wounded late Wednesday ···	Wounded	Injure
Four variants of T1 and T2 in testing:	DMBERT	Ours
V1: There have been too many civilians killed.	Die √	Die √
V2: ··· a sister were killed late Wednesday···	Injure ×	Die √
V3: There have been too many civilians wounded . . .	Die ×	Injure √
V4: EU foreign policy supremo Javier Solana likewise slammed the attack, saying, "There have been too many civilians wounded . . ."	Other ×	Attack ×

Figure 1: A case study of SOTA model DMBERT (Wang et al., 2019) and our proposed model. We train DMBERT and our proposed model on ACE2005, and select two texts (T1 and T2) with "killed" and "wounded" as triggers in the training set. For testing, we change the contexts of the two triggers in T1 and T2 to obtain V1 to V4, and let the two ED models predict their event types in the new contexts. We can see that DMBERT is easy to get confused in the new contexts, while our model is obviously more stable. The detailed explanations are in Section 4.3.5.

Theoretically, the contextual information in a given text is enough to determine the event type of each word. But the fact is an event can be described by various contexts, and the contexts in the test stage are usually not covered by the training set. Therefore, a common shortcoming of existing neural-based ED models is that they are prone to get confused by the changeable contexts during testing. As shown in Figure 1, the variation of any

* Corresponding author.

[1] The event trigger is usually a single verb or nominalization, and some papers also refer to the multi-word trigger as event nugget. In this paper, we uniformly use "trigger".

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 2625–2635
November 7–11, 2021.

surrounding words will lead to a different set of extracted contextual features and may misguide their final judgments.

To alleviate this shortcoming, a natural idea is to seek an additional "stable" decision-making basis unchanged across varied contexts to cooperate with the contextual features. Therefore, we envisage extracting a set of event features of words from the entire training set through global statistics—word-event co-occurrence frequencies.

Global statistics were widely used by many early pattern-based and feature-based ED models (Liao and Grishman, 2010; Li et al., 2013). However, with the introduction of deep learning technologies (Chen et al., 2015), various powerful neural-based ED models directly extract contextual semantic features from each text via end-to-end architectures. The use of global statistics has gradually faded out of researchers' attention. In our work, we find that the word-event co-occurrence frequencies in ED datasets have potential to help neural-based ED models eliminate a large number of interference options in testing (Section 3.1). Thus, as the title suggests, the valuable features for neural-based ED models can be collected outside the contexts.

Specifically, we design a simple but novel **S**emantic and **S**tatistic-**J**oint **D**iscriminative **N**etwork (**S^2-JDN**) to implement the new decision-making pattern, which consists of three modules. (i) *Semantic feature extractor* takes Dynamic Multi-pooling BERT (Wang et al., 2019) as the prototype and is used to obtain the token-level contextual features in each text. (ii) *Statistical feature extractor* mines event features in both direct and indirect ways from word-event co-occurrence frequencies, and then fuses and rescales them as the final statistical event features. (iii) *Joint event discriminator* adopts a layer normalization to unify the two types of event features and combines them for decision-making.

In experiments, we compare with ten strong baselines on ACE2005 and KBP2015 datasets, where S^2-JDN exceeds the state-of-the-art (SOTA) models by 1.9% and 1.9% in trigger classification F1, respectively. Further, we perform extensive experiments and draw multiple useful conclusions about S^2-JDN.

Our contributions can be summarized as follows:

(1) As far as we know, all existing neural-based ED models make decisions entirely based on contextual features (i.e., the standard pattern) in the NLP field, and we are the first to explore to introduce the statistical features as an additional stable decision-making basis.

(2) We propose S^2-JDN for ED task, which takes into account both the (contextual) semantic features and statistical features of each word to make a decision. Specifically, the statistical feature extractor of S^2-JDN mines features in both direct and indirect ways from word-event co-occurrence frequencies, thereby acquiring abundant statistical event features.

(3) We demonstrate that S^2-JDN effectively exceeds ten strong baselines on ACE2005 and KBP2015 datasets, and conduct extensive exploration experiments to comprehensively probe S^2-JDN.

2 Related Work

As the key component of the event extraction system (Yang et al., 2019; Li et al., 2020; Ferguson et al., 2018), the research of ED has experienced the periods of traditional methods and deep learning methods.

During the period of traditional methods, global statistics collected from the training set were widely used as the knowledge sources or decision-making basis of different ED models (Saurí et al., 2005; Ahn, 2006; Ma and Cisar, 2009; Liao and Grishman, 2010; Li et al., 2013). Grishman et al. (2005) and Shinyama and Sekine (2006) used statistical information to train a MaxEnt event classifier. Wan et al. (2009) constructed a frequency pattern-based framework for ED. Ji and Grishman (2008) obtained document-wide and cluster-wide statistics to correct the results of ED. Qin et al. (2013) proposed a classifier-based method to process statistical features for event filtering. Cao et al. (2015) recorded the frequency that each pattern is associated with an event type, and treated the frequencies as core features.

With the introduction of deep learning technologies, many neural-based ED models extracted contextual semantic features from each text via end-to-end architectures (Orr et al., 2018; Liu et al., 2019; Lai et al., 2020). Chen et al. (2015) acquired the contextual features via convolution and dynamic multi-pooling techniques. Liu et al. (2018) and Zhao et al. (2018) employed the attention mechanisms during contextual feature extraction. Nguyen and Grishman (2015), Ding et al. (2019), and Yan et al. (2019) introduced external knowledge to as-

sist neural networks to better understand each given text. Recently, many studies applied powerful pre-trained language models to better comprehend contexts (Du and Cardie, 2020; Liu et al., 2020a,b; Huang and Ji, 2020).

Neural-based ED models significantly outperform the traditional ones, and become the new research hotspot. Accordingly, the use of global statistics has faded out of researchers' attention. As far as we know, there is no neural-based ED model explicitly using global statistics in the training sets. Although neural networks possess powerful learning ability in theory, their training is based on individual sentences, and thus it's not easy for them to capture the global statistics. More importantly, we observe that the word-event co-occurrence frequencies for most words concentrate on only a few events, which are potential to help neural-based ED models eliminate a large number of interference options in testing (Section 3.1). To this end, we propose S^2-JDN, which takes into account both the (contextual) semantic features and statistical features of each word to make a decision. Besides employing the collected global statistics as features directly like traditional ED models, S^2-JDN also leverages the property of neural networks to indirectly extract more abundant features from word-event co-occurrence frequencies (Section 3.2.2).

3 Methodology

In this section, we first elaborate the motivation for introducing word-event co-occurrence frequencies (Section 3.1), then propose a concrete instantiation called Semantic and Statistic-Joint Discriminative Network (S^2-JDN) (Section 3.2), and finally describe the training details of S^2-JDN (Section 3.3).

3.1 Motivation for Introducing Word-Event Co-occurrence Frequencies

The benefits of word-event co-occurrence frequencies for neural-based ED models are as follows:

Stability and Accessibility. First of all, word-event co-occurrence frequencies are collected from the entire training set and will not be disturbed by various contexts, which satisfy our requirement of stability. With their assistance, the neural-based ED models are easier to make correct predictions in the changeable contexts during testing. Also, the collection of these statistics does not rely on any external tools or data.

Clear Directivity. The event set of an ED dataset consists of many event types and a special "Other" type for non-triggers. As plotted in Figure 2, an ED dataset often contains dozens of events, but most words can only evoke at most three of them[2]. This phenomenon can be explained by the characteristic of ED task, i.e., although the total event number of a realistic ED dataset can be large, most triggers are single words[3] with limited meaning and can only evoke very few events. This characteristic can be reflected by the word-event co-occurrence frequencies in the ED dataset. We refer to the trait that word-event co-occurrence frequencies concentrate on only a few events as clear directivity. Therefore, such statistics can provide clear indications for neural-based ED models and have great potential to help them eliminate lots of interference options in testing.

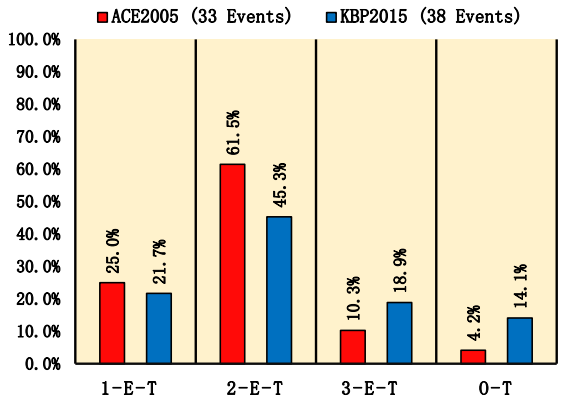

Figure 2: Statistics about the numbers of events that can be evoked by each trigger word in ACE2005 and KBP2015 datasets. In our statistics, a word will be regarded as a trigger word if it evokes a specified event in a text in the dataset. When counting the number of events evoked by a trigger word, "Other" will also be considered. k-E-T ($k = 1, 2, 3$) denotes the proportion of the trigger words evoking k events over all trigger words, and O-T corresponds to the rest trigger words.

3.2 Semantic and Statistical-Joint Discriminative Network

To take advantage of both contextual semantic features and the statistical features extracted from word-event co-occurrence frequencies, we propose a simple but novel **S**emantic and **S**tatistic-**J**oint **D**iscriminative **N**etwork (**S^2-JDN**) as shown in Fig-

[2]Please see Section 4.1 for more details of ACE2005 and KBP2015 datasets.

[3]According to the statistics, the numbers of multi-word/single-word triggers are 234/4994 and 362/12350 on ACE2005 and KBP2015 datasets.

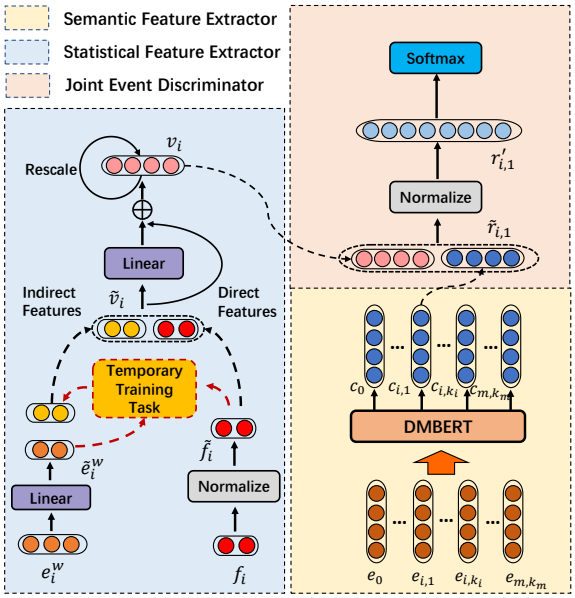

Figure 3: The architecture of proposed S^2-JDN.

ure 3, which consists of a semantic feature extractor, a statistical feature extractor, and a joint event discriminator.

3.2.1 Semantic Feature Extractor

The semantic feature extractor collects contextual features from given texts and is acted by a Dynamic Multi-pooling BERT (Wang et al., 2019) in this work. It is also possible to explore other choices in future work.

Consider an m-word input sequence $s_w = (w_1, \ldots, w_m)$. After wordpiece tokenization (Wu et al., 2016), s_w is further decomposed into $([\text{CLS}], \text{t}_{1,1}, \ldots, \text{t}_{1,k_1}, \ldots, \text{t}_{m,1}, \ldots, \text{t}_{m,k_m})$, where [CLS] is a specific token of BERT (Devlin et al., 2018), $t_{i,j}$ is the j_{th} token of w_i. We use $e_{i,j}$ to denote the BERT's input corresponding to $t_{i,j}$, which is the sum of the token and position embeddings[4]. The BERT's output for $e_{i,j}$ is $h_{i,j}$. Next, $(h_{1,1}, \ldots, h_{m,k_m})$ are further processed by a dynamic multi-pooling operation, and the output of $h_{i,j}$ is denoted as $c_{i,j}$ and calculated as:

$$c^z_{i,j} = \max\{h^z_{1,1}, ..., h^z_{i,j}\} + \max\{h^z_{i,j+1}, ..., h^z_{m,k_m}\} \quad (1)$$

where $c^z_{i,j}$ and $h^z_{i,j}$ is the z^{th} features of $c_{i,j}$ and $h_{i,j}$. Dynamic multi-pooling extracts important features on both sides of $t_{i,j}$ to form its contextual vector. Since the size of $h_{i,j}$ is large enough, we combine the pooling results by addition operation, rather than concatenation.

3.2.2 Statistical Feature Extractor

Parallel to the semantic feature extractor, the statistical feature extractor mines a set of features from word-event co-occurrence frequencies as additional decision-making basis. To take full advantage of these statistics, we extract features in both direct and indirect ways.

Extracting Direct Statistical Features: For word w_i in text s_w, we denote its word-event co-occurrence frequency vector as $f_i = (n^1_i, \ldots, n^K_i) \in \mathbb{R}^K$, where n^z_i is the number of times w_i evokes the z^{th} event type in the training set, and K is the total event number (including "Other"). f_i is normalized to a vector with direct statistical features $\tilde{f}_i = f_i / n_i$, where $n_i = \sum_{z=1}^{K} n^z_i$ is the total number of w_i's occurrences in the training set. In testing, $\tilde{f}_i = \mathbf{0}$ for words unseen in the training set.

Although most $\tilde{f}_i$ have clear directivity and can reflect global event information of w_i (Section 3.1), its feature dimensions are much smaller than the contextual vectors $c_{i,j}$. Therefore, the information provided by $\tilde{f}_i$ might be insufficient to guide the final decision. In light of this, we propose a temporary training task named Frequency Supervised Multi-Label Classification (FSMLC), which finetunes the generic word embedding of each word w_i to acquire more statistical features indirectly.

Extracting Indirect Statistical Features: As mentioned above, FSMLC needs to encode statistical information into the generic word embedding e^w_i of w_i by finetuning. In our work, e^w_i is 300-dimension Glove embedding (Pennington et al., 2014). Concretely, FSMLC first maps e^w_i into a new vector $\tilde{e}^w_i$ by a linear transformation, then feeds $\tilde{e}^w_i$ into a temporary event classifier and adopts the normalized frequencies in $\tilde{f}_i$ as the targets for training. The loss function of FSMLC for word sequence s_w is:

$$\mathcal{L}^{s_w}_{MLC} = -\sum_{i=1}^{m} \frac{1}{n_i} \tilde{f}_i \cdot \log(\tilde{p}_i) \quad (2)$$

$$\tilde{p}^z_i = \frac{\exp(\tilde{e}^w_i \cdot \tilde{u}_z)}{\sum_{l=1}^{K} \exp(\tilde{e}^w_i \cdot \tilde{u}_l)} \quad (3)$$

$$\tilde{e}^w_i = M_1 \cdot e^w_i \quad (4)$$

where $\tilde{p}^z_i$, and $\tilde{u}_z$ are the predicted probability and trainable projection vector of the z^{th} event type, M_1 is the trainable linear transformation matrix.

[4]Since the input sequence contains only one segment, the segment embedding and the special token [SEP] are removed.

Now, we explain the settings of FSMLC. According to previous studies (Bespalov et al., 2012; Tang et al., 2014), word embeddings can gain the ability to represent some specific information via targeted training. Therefore, the temporary training task FSMLC finetunes the generic semantic features within e_i^w into a set of new ones in $\tilde{e}_i^w$ that can better represent the events evoked by w_i in the training set, thereby indirectly using $\tilde{f}_i$ to obtain more features. Note that the trained $\tilde{e}_i^w$ possesses both semantic information and word-event co-occurrence information, but we refer to it as an indirect statistical feature vector. Finally, the loss of FSMLC on the entire training set is:

$$\mathcal{L}_{MLC} = \sum_{s_w} \mathcal{L}_{MLC}^{s_w} = -\sum_{w \in V} \tilde{f}_w \cdot \log(\tilde{p}_w) \quad (5)$$

where V is the vocabulary of the training set. Therefore, all words can get the same level of training. In testing, the temporary classifier will be discarded.

Feature Fusing and Rescaling: The statistical event vector of w_i is the fusion of $\tilde{f}_i$ and $\tilde{e}_i^w$:

$$v_i = M_2 \cdot \tilde{v}_i + \tilde{v}_i \quad (6)$$

where $\tilde{v}_i = \left[\tilde{f}_i; \tilde{e}_i^w\right]$, M_2 is a trainable fusion matrix. Although Eq.(6) is formally equivalent to a linear transformation of $\tilde{v}_i$, we find that the approach of separating $\tilde{v}_i$ alone and adding it to $M_2 \cdot \tilde{v}_i$ sometimes performs better in practice.

As is well-known, the credibility of statistics is closely related to the number of each word's occurrences. The fewer times a word appears in the training set, the less reliable its word-event occurrence frequencies are. To weaken the influence of low-frequency words' statistical features, we rescale v_i as:

$$v_i^{'} = \alpha_i \cdot v_i, \quad \alpha_i = \min\{1, \frac{n_i}{c}\} \quad (7)$$

where c is an integer hyperparameter denoting the threshold of occurrence number that a word's statistics can be trusted. For $n_i < c$, v_i is scaled down, and $v_i^{'} = \mathbf{0}$ for words unseen in the training set.

3.2.3 Joint Event Discriminator

The joint event discriminator combines the token-level semantic vectors $c_{i,j}$ and the word-level statistical vector $v_i^{'}$ for decision-making.

Firstly, each $c_{i,j}$ $(j \in 1, \ldots, k_i)$ will be concatenated with $v_i^{'}$ to form the token-level event vector $\tilde{r}_{i,j} = [c_{i,j}; v_i^{'}]$. Since features in $c_{i,j}$ and $v_i^{'}$ come from two different sources, we apply a simple layer normalization to unify them:

$$r_{i,j}^z = \frac{\tilde{r}_{i,j}^z - \mu_{i,j}}{\sigma_{i,j}} \quad (8)$$

$$\mu_{i,j} = \frac{1}{d_r} \sum_z \tilde{r}_{i,j}^z \quad (9)$$

$$\sigma_{i,j} = \sqrt{\frac{1}{d_r} \sum_z (\tilde{r}_{i,j}^z - \mu_{i,j})^2} \quad (10)$$

where $r_{i,j}^z$ and $\tilde{r}_{i,j}^z$ are the z^{th} features of $r_{i,j}$ and $\tilde{r}_{i,j}$, and d_r is the dimension of vector $\tilde{r}_{i,j}$.

In testing, if $n_i = 0$, we have $v_i^{'} = 0$, but Eq.(8-10) will change the statistical part in $r_{i,j}$ to non-zero values. So we multiply $r_{i,j}$ by a vector to change the statistical part to zeros again after normalizing, i.e., $r_{i,j}^{'} = r_{i,j} \odot [\mathbf{1}; I_i^{sta}]$, where I_i^{sta} corresponds to the statistical part in $r_{i,j}$, and $I_i^{sta} = \mathbf{0}$ if $n_i = 0$ else $I_i^{sta} = \mathbf{1}$; $\odot$ is element-wise multiplication.

Then, $r_{i,j}^{'} (j = 1, ..., k_i)$ are inputted into a Softmax layer to calculate the token-level prediction distributions $p_{i,j}$:

$$p_{i,j}^z = \frac{\exp\left(r_{i,j}^{'} \cdot u_z\right)}{\sum_{l=1}^{K} \exp\left(r_{i,j}^{'} \cdot u_l\right)} \quad (11)$$

where $p_{i,j}^z$ and u_z are respectively the predicted probability and trainable projection vector of the z^{th} event type. The prediction distribution of word w_i is the average of $p_{i,j}$, i.e., $p_i = \frac{1}{k_i} \sum_{j=1}^{k_i} p_{i,j}$.

3.3 Training of S^2-JDN

In the training of S^2-JDN, besides the BERT, the trainable parameters also contain $\tilde{u}_z$ and M_1 in Eq.(3-4), M_2 in Eq.(6), and u_z in Eq.(11). We jointly train the main task ED and the temporary task FSMLC, and the total loss for sequence s_w is:

$$\mathcal{L}^{s_w} = -\sum_{i=1}^{m} q_i \cdot \log(p_i) + \beta \mathcal{L}_{MLC}^{s_w} \quad (12)$$

where q_i is the one-hot ground-truth event label of w_i, β is the coefficient of FSMLC's loss. In experiments, the loss of Eq.(12) is optimized by Adam optimizer (Kingma and Ba, 2014).

4 Experiments

4.1 Experimental Setups

Datasets: We take two benchmark datasets ACE2005[5] and KBP2015[6] for evaluation. ACE2005/KBP2015 contains 599/360 documents and 33/38 specified event types. For ACE2005, we follow the previous studies (Peng et al., 2016; Du and Cardie, 2020; Liu et al., 2020b; Lai et al., 2020) and use 40 documents from newswire domain for testing, 30 documents for validation, and the rest for training. For KBP2015, we also use the official test set, and split about 20% of sentences from the training set for validation. Their statistics are presented in Table 1.

Implementation Details of S^2-JDN[7]: On ACE2005/KBP2015, the learning rate is 1e-5/2.5e-5 (from [1e-6, 1e-4]), batch size is 48/16 (is selected from {16, 32, 48, 64, 128}). For the semantic feature extractor, the BERT has 24 16-head attention layers and 1024 hidden embedding dimension. For the statistical feature extractor, the dimension of $\tilde{e}_i^w$ in Eq.(4) is 300/200 (from {50, 100, 200, 300, 400, 500}), and we adopt dropout with dropout rate 0.5/0.35 (from [0,0.5]) on $\tilde{e}_i^w$. The threshold c in Eq.(7) is 4/4 (from {1,...,5}), and the coefficient β in Eq.(12) is 15/10 (from {1, 2, 3, 5, 10, 15, 20}). The reason for large values of β is that the weights $1/n_i$ in Eq.(2) have reduced the loss of FSMIC. We experiment with Pytorch 0.4.1 on Nvidia Tesla P40 GPU. The best-performing S^2-JDN appears after 107/132 epochs, and each epoch takes about 4 minutes, so the total training time is in 7/8.8 hours.

4.2 Main Experiments

Baselines: We take SOTA models *without external ED-related knowledge* on ACE2005 and KBP2015 as baselines for fair comparisons. The baselines include: *MSEP* (Peng et al., 2016) develops an event detection and co-reference system with minimal supervision; *FBRNN* (Ghaeini et al., 2016) employs forward-backward recurrent neural networks for ED; *GCN-ED* (Nguyen and Grishman, 2018) exploits a new pooling method to aggregate the outputs of GCN; *PLMEE* (Yang et al., 2019) directly uses a basic BERT classification model to identify and classify triggers[8]; *DMBERT* (Wang et al., 2019) connects a dynamic multi-pooling behind a BERT to extract contextual features[9]; *ED-QA* (Du and Cardie, 2020) formulates ED as a question answering task; *RCEE* (Liu et al., 2020a) casts ED task as a machine reading comprehension problem; *M-FULL* (Liu et al., 2020b) proposes a training mechanism called context-selective mask generalization for ED; EEGCN (Cui et al., 2020) simultaneously exploits syntactic structure and typed dependency label information to perform ED; *G-GCN* (Lai et al., 2020) combines BERT, GCN, and gating mechanism for ED. Besides, we also present the top TAC KBP2015 Event Nugget Detection result: *TAC-TOP* (Mitamura et al., 2015).

Dataset	Train		Valid		Test	
	#Sen	#Tri	#Sen	#Tri	#Sen	#Tri
ACE2005	14477	4088	831	386	952	764
KBP2015	5175	5450	1167	931	3809	6331

Table 1: Statistics of training/valid/test set of ACE2005 and KBP2015 datasets. #Sen and #Tri respectively denote the numbers of the sentences and triggers.

	ACE2005 (%)			KBP2015 (%)		
	P	R	F1	P	R	F1
TACTop	N/A	N/A	N/A	75.2	47.7	58.4
ED-QA	71.1	73.7	72.4	N/A	N/A	N/A
RCEE	75.6	74.2	74.9	N/A	N/A	N/A
M-full	75.2	74.4	74.8	N/A	N/A	N/A
EEGCN	76.7	78.6	77.6	N/A	N/A	N/A
G-GCN	78.8	76.3	77.6	N/A	N/A	N/A
MSEP	70.4	65.0	67.6	69.2	47.8	56.6
FBRNN	66.8	68.0	67.4	71.6	48.2	57.6
GCNED	77.9	68.8	73.1	70.3	50.6	58.8
PLMEE*	76.0	73.9	74.9	75.5	53.4	62.6
DMBERT*	77.5	76.2	76.8	76.1	54.9	63.8
S^2-JDN	**80.3**	**78.8**	**79.5**	**78.4**	**56.5**	**65.7**

Table 2: Precision (P), Recall (R), and micro-F1 (F1) of trigger classification. The baselines with "*" are implemented by ourselves.

Results and Analysis By convention, we use precision, recall, and micro-F1 as metrics, which are presented in Table 2.

As shown in Table 2, on ACE2005, S^2-JDN exceeds the SOTA models EEGCN/G-GCN by 3.6%/1.5%, 0.2%/2.5%, and 1.9%/1.9% in terms of

[5]https://catalog.ldc.upenn.edu/LDC2006T06

[6]https://tac.nist.gov/2015/KBP/data.html

[7]The code is available at https://github.com/Buted/SSJDN.

[8]Since we cannot fully reproduce their results in our settings, we report the results implemented by ourselves instead.

[9]Since DMBERT is the base model of our S^2-JDN, we implement it by ourselves to ensure a fair comparison.

precision, recall, and F1. On KBP2015, DMBERT and S^2-JDN are significantly better than MSEP, FBRNN, GCNED, and TACTop with the help of the pre-trained BERT, and S^2-JDN further outperforms DMBERT by 2.3%, 1.6%, and 1.9% in precision, recall, and F1. All the neural-based baselines employ the standard decision-making pattern, and RCEE, ED-QA, M-FULL, G-GCN, PLMEE, and DMBERT are engined by BERT. These results show the effectiveness of S^2-JDN introducing the statistical features from word-event co-occurrence frequencies. Therefore, we can infer that the new decision-making pattern used by S^2-JDN is more suitable for ED task than the standard one.

4.3 Further Exploration

4.3.1 Ablation Study

We evaluate the importance of each part in S^2-JDN via ablation studies on five variants. V1 removes the indirect statistical features (ISF) by substituting v_i with $\tilde{f}_i$. V2 omits the direct statistical features (DSF) by replacing v_i with $\tilde{e}_i^w$. V3 removes the temporary training task FSMLC by setting $\beta = 0$; V4 omits layer normalization (LN) in joint event discriminator by directly inputting $\tilde{r}_{i,j}$ into the final Softmax layer. V5 eliminates the introduction of global statistics (GS) via replacing $v_i^{'}$ with the Glove embedding e_i^w. The results are reported in Table 3.

We analyze the results in Table 3 from five aspects. (i) V0 outperforms V1-V5 on all datasets, which shows that each part of S^2-JDN has a positive effect on the overall performance. (ii) V1 is worse than V2. This is because the information in direct statistical features are relatively deficient and sparse, and thus is harder to change the final decisions. Therefore, the indirect statistical features are more powerful than the direct ones. (iii) V3 is better than V1, which means simply using the generic word embeddings as the indirect statistical features can also bring some improvement. We speculate that this is because the word embeddings can indicate the "identities" of the words and make it easier to recognize the words that fixedly evoke some events. (iv) V4 is worse than V2, so layer normalization is necessary, whose importance even exceeds the direct statistical features. (v) Theoretically, V5 should be the worst one among variants V1-V5, but it performs slightly better than V1. We can combine the above analyses (ii) and (iii) to explain. On the one hand, the deficient and sparse direct statistical features of V1 are difficult to effectively change the overall model's decisions. On the other hand, the model of V5 can learn to recognize the "identities" of words according to Glove embedding during the training of ED task.

		ACE2005	KBP2015
V0	S^2-JDN	79.5	65.7
V1	-without ISF	77.3	64.5
V2	-without DSF	78.8	65.1
V3	-without FDMLC	77.8	64.9
V4	-without LN	78.6	65.0
V5	-without GS	77.7	64.6

Table 3: Trigger classification F1 scores (%) of ablation studies on the three datasets.

To further ensure the credibility of the results, we train each variant for three times with different random seeds, which shows that the standard deviations of S^2-JDN are about 0.3% on both ACE2005 and KBP2015. For the other five variants, the standard deviations are within 0.2%-0.5%. To conclude, our proposed model is stable, and the improvements over the five variants are significant.

4.3.2 Effects of Statistics with Different Degrees of Directivity

In Section 3.1, we pointed out that an advantage of word-event co-occurrence frequencies is clear directivity, which is caused by the characteristic of ED task. The fewer the events evoked by a word, the more clear the directivity is. In this experiment, we study the effect of statistics with different degrees of directivity. Concretely, we divide the words into four categories during testing: $\mathcal{C}1/\mathcal{C}2/\mathcal{C}3/\mathcal{C}4$ respectively corresponds to the words evoking 1/2/3/$\geq$4 events in the training set. For comparisons, we construct a base model with only semantic event features (Section 3.2.1) as the decision-making basis and denote it as **Base**. The results of the base model and S^2-JDN on the 4 categories of words are reported in Table 4.

From Table 4, we can find that S^2-JDN has a certain improvement on each category, and its advantages decrease across $\mathcal{C}1$, $\mathcal{C}2$, $\mathcal{C}3$, and $\mathcal{C}4$, which means the more clear the directivity of word-event co-occurrence frequencies possess, the greater the effect of the statistical features have. Thus, the effectiveness of S^2-JDN is closely related to the characteristic of ED task.

		$\mathcal{C}1$	$\mathcal{C}2$	$\mathcal{C}3$	$\mathcal{C}4$
ACE2005	Base	76.9	77.8	71.3	65.2
	S^2-JDN	80.8	80.2	72.7	66.0
KBP2015	Base	66.9	65.2	58.8	57.0
	S^2-JDN	70.1	67.3	60.1	57.6

Table 4: Trigger classification F1 scores (%) of each model on the word instances of the 4 categories.

		ACE2005	KBP2015
Proportion of $\lvert\mathcal{A}_{\mathcal{G}}\rvert$	Base	7.5%	7.7%
	S^2-JDN	5.5%	6.1%
Proportion of $\lvert\mathcal{A}_{\mathcal{NG}}\rvert$	Base	15.0%	16.2%
	S^2-JDN	14.2%	15.5%

Table 5: Proportions of $\lvert\mathcal{A}_{\mathcal{G}}\rvert$ and $\lvert\mathcal{A}_{\mathcal{NG}}\rvert$ in the test of ACE2005 and KBP2015 for Base and S^2-JDN.

4.3.3 Performance on Predicting Global and Non-Global Events

In the training set, we refer to the most frequent event (including "Other") evoked by a word as its **global event**, which corresponds to the maximum word-event co-occurrence frequency of the word. Are the statistical features introduced by S^2-JDN only helpful for predicting global events, but not for words with non-global event as the ground-truth labels in testing? To answer this question, we conduct the following experiments.

In each test set, we gather all word instances wrongly predicted by an ED model as collection $\mathcal{A}$. From $\mathcal{A}$, we further select the word instances with their global events as the ground-truth labels to form collection $\mathcal{A}_{\mathcal{G}}$, and pick out the word instances with non-global events as the ground-truth labels to constitute collection $\mathcal{A}_{\mathcal{NG}}$. Obviously, we have $\mathcal{A} = \mathcal{A}_{\mathcal{G}} \cup \mathcal{A}_{\mathcal{NG}}$. The instance proportions of $\mathcal{A}_{\mathcal{G}}$ and $\mathcal{A}_{\mathcal{NG}}$ in the entire test set respectively indicate the proportions of model mistakes caused by inconsistency with global events and non-global events. The results of the base model and S^2-JDN are presented in Table 5.

As shown in Table 5, after introducing statistical features, the proportion of $\lvert\mathcal{A}_{\mathcal{G}}\rvert$ is decreased by 2.0% and 1.6% on ACE2005 and KBP2015 test sets, and the proportion of $\lvert\mathcal{A}_{\mathcal{NG}}\rvert$ is correspondingly reduced by 0.8% and 0.7%. Therefore, the statistical features are not just helpful for predicting the most frequent events evoked by each word. For words with non-global events as ground-truth labels, they also have certain benefits.

	ACE2005			KBP2015		
	$\mathcal{U}$	$\mathcal{L}$	$\mathcal{H}$	$\mathcal{U}$	$\mathcal{L}$	$\mathcal{H}$
Base	51.3	69.4	82.7	42.1	59.5	70.2
S^2-JDN$^-$	44.7	69.7	85.8	39.4	59.6	72.0
S^2-JDN	50.1	70.6	87.1	41.5	60.8	74.3

Table 6: Trigger classification F1 scores (%) of Base, S^2-JDN$^-$, S^2-JDN on $\mathcal{U}$ (Unseen words), $\mathcal{L}$ (Low-frequency words), and $\mathcal{H}$ (High-frequency words).

4.3.4 Effects on Words with Different Occurrence Frequencies

As discussed in Section 3.2.2, the credibility of statistics will decrease with the number of a word's occurrences. To this end, we rescale v_i as Eq.(7) to weaken the impact of low-frequency words' statistical features. Now, we compare the model performance on words with different occurrence frequencies in the training set. Here we define low-frequency words according to hyperparameter c in Eq.(7), which is 4 times on ACE2005 an KBP2015 (Section 4.1). Accordingly, we split the words in each test set into three parts: $\mathcal{U}$={words unseen in the training set}, $\mathcal{L}$={low-frequency words with occurrence numbers in $\{1, 2, 3\}$ in the training set}, $\mathcal{H}$={high-frequency words appearing at least 4 times in the training set}. Besides the base model and S^2-JDN, we also test a variant of S^2-JDN that removes the rescaling operations in Eq.(7) and the element-wise multiplication with $[\mathbf{1}, I_i^{sta}]$ after layer normalization in joint event discriminator (Section 3.2.3), denoted as S^2-JDN$^-$. Their results are shown in Table 6.

As expected, all models perform best on $\mathcal{H}$ in Table 6, and perform worst on $\mathcal{U}$. In predicting the events of words in $\mathcal{U}$, the statistical features are useless and even become interference, so the results of S^2-JDN$^-$ are significantly worse than the other two models, and S^2-JDN avoids the interference by the two rescaling operations. On $\mathcal{L}$ and $\mathcal{H}$, S^2-JDN$^-$ and S^2-JDN are better than Base, which is due to the effect of the statistical features. However, the training and testing of S^2-JDN$^-$ are disturbed by some low-frequency words' false statistical information more seriously, so it can't achieve the same results as S^2-JDN on $\mathcal{L}$ and $\mathcal{H}$.

4.3.5 Case Study

In this subsection, we will analyze the case study results of DMBERT and our S^2-JDN shown in Table 1. V1 is the simplest variant, which just cuts off the

part after "killed" in T1. Although the contextual features of the two models are changed in V1, both models can predict the correct event "Die". V2 and V3 exchange the surrounding words of "killed" and "wounded" in T1 and T2, which successfully confuses DMBERT, while S^2-JDN still makes the correct decisions with the help of statistical features. V4 is the most complex variant. Besides replacing the trigger word in T1 with "wounded", V4 also adds a long piece of content (green part). At this time, DMBERT can't even identify "wounded" as a trigger, and S^2-JDN also wrongly predicts the event as "Attack".

This experiment shows that the statistical features are indeed helpful in varied contexts. But if the contexts change too significantly, S^2-JDN may still be misguided.

5 Conclusion & Future Work

In this paper, we find that existing neural-based ED models are likely to get confused by changeable contexts during testing. To alleviate this problem, we propose S^2-JDN model, which extracts a set of statistical event features from word-event co-occurrence frequencies as an additional decision basis besides contextual information. Experimental results on two benchmark datasets ACE2005 and KBP2015 against ten recent SOTA ED models demonstrate the effectiveness of S^2-JDN and each proposed module.

For future work, there are three intriguing directions: (i) extending the decision-making pattern of S^2-JDN to other neural-based ED models and thus absorbing their advantages; (ii) incorporating the word-event co-occurrence frequencies into neural-based ED models via prior on their output distributions in a Bayesian framework; and (iii) combining S^2-JDN with data augmentation methods, so as to collect more accurate global statistics and further promote the performance.

Acknowledgments

This work is supported by the Innovation Research Group Project of NSFC (61921003), the National Natural Science Foundation of China (No. 62002029), and the Fundamental Research Funds for the Central Universities 2020RC23. We also would like to thank anonymous reviewers for their insightful comments.

References

David Ahn. 2006. The stages of event extraction. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, pages 1–8.

Dmitriy Bespalov, Yanjun Qi, Bing Bai, and Ali Shokoufandeh. 2012. Sentiment classification with supervised sequence embedding. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 159–174. Springer.

Kai Cao, Xiang Li, and Ralph Grishman. 2015. Improving event detection with dependency regularization. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 78–83.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. 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 167–176, Beijing, China. Association for Computational Linguistics.

Shiyao Cui, Bowen Yu, Tingwen Liu, Zhenyu Zhang, Xuebin Wang, and Jinqiao Shi. 2020. Edge-enhanced graph convolution networks for event detection with syntactic relation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 2329–2339.

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*.

Ning Ding, Ziran Li, Zhiyuan Liu, Haitao Zheng, and Zibo Lin. 2019. Event detection with trigger-aware lattice neural network. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 347–356.

Xinya Du and Claire Cardie. 2020. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 671–683.

James Ferguson, Colin Lockard, Daniel S Weld, and Hannaneh Hajishirzi. 2018. Semi-supervised event extraction with paraphrase clusters. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 359–364.

John R Firth. 1957. A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*.

Reza Ghaeini, Xiaoli Fern, Liang Huang, and Prasad Tadepalli. 2016. Event nugget detection with forward-backward recurrent neural networks. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 369–373.

Ralph Grishman, David Westbrook, and Adam Meyers. 2005. Nyu's english ace 2005 system description. *ACE*, 5.

Zellig S Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.

Yu Hong, Wenxuan Zhou, Jingli Zhang, Guodong Zhou, and Qiaoming Zhu. 2018. Self-regulation: Employing a generative adversarial network to improve event detection. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 515–526.

Lifu Huang and Heng Ji. 2020. Semi-supervised new event type induction and event detection. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 718–724.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, ACL-08: HLT*, pages 254–262.

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

Viet Dac Lai, Tuan Ngo Nguyen, and Thien Huu Nguyen. 2020. Event detection: Gate diversity and syntactic importance scores for graph convolution neural networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5405–5411.

Fayuan Li, Weihua Peng, Yuguang Chen, Quan Wang, Lu Pan, Yajuan Lyu, and Yong Zhu. 2020. Event extraction as multi-turn question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 829–838.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 73–82.

Shasha Liao and Ralph Grishman. 2010. Using document level cross-event inference to improve event extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 789–797.

Jian Liu, Yubo Chen, and Kang Liu. 2019. Exploiting the ground-truth: An adversarial imitation based knowledge distillation approach for event detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6754–6761.

Jian Liu, Yubo Chen, Kang Liu, Wei Bi, and Xiaojiang Liu. 2020a. Event extraction as machine reading comprehension. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1641–1651.

Jian Liu, Yubo Chen, Kang Liu, Yantao Jia, and Zhicheng Sheng. 2020b. How does context matter? on the robustness of event detection with context-selective mask generalization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 2523–2532.

Shaobo Liu, Rui Cheng, Xiaoming Yu, and Xueqi Cheng. 2018. Exploiting contextual information via dynamic memory network for event detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1030–1035.

Yunqian Ma and Petr Cisar. 2009. Event detection using local binary pattern based dynamic textures. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 38–44. IEEE.

Teruko Mitamura, Zhengzhong Liu, and Eduard H Hovy. 2015. Overview of tac kbp 2015 event nugget track. In *TAC*.

Thien Huu Nguyen and Ralph Grishman. 2015. Event detection and domain adaptation with convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 365–371.

Thien Huu Nguyen and Ralph Grishman. 2018. Graph convolutional networks with argument-aware pooling for event detection. In *AAAI*, volume 18, pages 5900–5907.

J Walker Orr, Prasad Tadepalli, and Xiaoli Fern. 2018. Event detection with neural networks: A rigorous empirical evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 999–1004.

Haoruo Peng, Yangqiu Song, and Dan Roth. 2016. Event detection and co-reference with minimal supervision. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 392–402.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Yanxia Qin, Yue Zhang, Min Zhang, and Dequan Zheng. 2013. Feature-rich segment-based news event detection on twitter. In *Proceedings of the sixth international joint conference on natural language processing*, pages 302–310.

Roser Saurí, Robert Knippen, Marc Verhagen, and James Pustejovsky. 2005. Evita: A robust event recognizer for QA systems. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 700–707, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Yusuke Shinyama and Satoshi Sekine. 2006. Preemptive information extraction using unrestricted relation discovery. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 304–311.

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1555–1565.

Meihan Tong, Shuai Wang, Yixin Cao, Bin Xu, Juanzi Li, Lei Hou, and Tat-Seng Chua. 2020a. Image enhanced event detection in news articles. In *AAAI*, pages 9040–9047.

Meihan Tong, Bin Xu, Shuai Wang, Yixin Cao, Lei Hou, Juanzi Li, and Jun Xie. 2020b. Improving event detection via open-domain trigger knowledge. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5887–5897.

Li Wan, Jianxin Liao, and Xiaomin Zhu. 2009. A frequent pattern based framework for event detection in sensor network stream data. SensorKDD '09, page 87–96, New York, NY, USA. Association for Computing Machinery.

Xiaozhi Wang, Xu Han, Zhiyuan Liu, Maosong Sun, and Peng Li. 2019. Adversarial training for weakly supervised event detection. 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 998–1008.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Haoran Yan, Xiaolong Jin, Xiangbin Meng, Jiafeng Guo, and Xueqi Cheng. 2019. Event detection with multi-order graph convolution and aggregated attention. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5770–5774.

Sen Yang, Dawei Feng, Linbo Qiao, Zhigang Kan, and Dongsheng Li. 2019. Exploring pre-trained language models for event extraction and generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5284–5294.

Yue Zhao, Xiaolong Jin, Yuanzhuo Wang, and Xueqi Cheng. 2018. Document embedding enhanced event detection with hierarchical and supervised attention. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 414–419.