Low Resource Event Extraction: A Survey

Viet Dac Lai
Department of Computer Science
University of Oregon
Eugene, OR 97403, USA
vietl@cs.uoregon.edu

Abstract

Extracting events from text is one of the key research themes in natural language processing. This process includes several tasks such as event detection, entity extraction, and argument extraction. As one of the most important topics in natural language processing, the applications of event extraction span across a wide range of domains such as newswire, biomedical domain, history and humanity, and cyber security. This report presents a comprehensive survey for event detection from textual documents. In this report, we provide the task definition, the evaluation method, as well as the benchmark datasets and a taxonomy of methodologies for event extraction. We also present our vision of future research directions in event detection.

1 Introduction

Event Extraction (EE) is an essential task in Information Extraction (IE) in Natural Language Processing (NLP). An event is an occurrence of an activity that happens at a particular time and place, or it might be described as a change of state (LDC, 2005). The main task of event extraction is to detect events in the text (i.e., event detection) and then sort them into some classes of interest (i.e., event classification). The second task involves detecting the event participants (i.e., argument extraction) and their attributes (e.g., argument role labeling). In short, event extraction structures the unstructured text by answering the *WH* questions of an event (i.e., what, who, when, where, why, and how).

Event extraction plays an important role in various natural language processing applications. For instance, the extracted event can be used to construct knowledge bases on which people can perform logical queries easily (Ge et al., 2018). Many domains can benefit from the development of event extraction research. In the biomedical domain, event extraction can be used to extract interaction between biomolecules (e.g., protein-protein interactions) that have been described in the biomedical literature (Kim et al., 2009). In the economic domain, events reported on social media and social networks can be used for measuring socioeconomic indicators (Min and Zhao, 2019). Recently, event extraction has been adopted in many other domains such as: literature (Sims et al., 2019), cyber security (Man Duc Trong et al., 2020), history (Sprugnoli and Tonelli, 2019), and humanity (Lai et al., 2021b).

Even though event extraction has been studied for decades, it is still a very challenging task. To perform the event extraction, a system needs to understand the text's semantics and ambiguity and organize the extracted information into structures (LDC, 2005).

It closely connects with other natural language processing tasks such as named entity recognition (NER), entity linking (EL), and dependency parsing. Although these tasks can boost the development of event extraction (McClosky et al., 2011), they might have an inverse impact on the performance of the event extraction systems (Zhang et al., 2018), depending on how the output of these tasks is exploited.

Last but not least, lacking training data is a fundamental problem in expanding event extraction to a new domain because the traditional classification model requires a large amount of training data (Huang et al., 2018b). Therefore, extracting events with a substantially small amount of training data is a new and challenging problem.

2 The event extraction task

2.1 Definition

Event extraction aims to detect the appearance of an event in the text (e.g., sentence, document) and

its related information. A predefined event structure is predefined to formulate the event, such as participants and their relations to the event. This is called close-domain event extraction, as the expected structure is provided according to a certain application. Otherwise, open-domain event extraction does not require any predefined structure. In this report, we focus on close-domain event extraction.

ACE-2005 (LDC, 2005) defines an event schema whose terminologies have been widely used in event extraction:

- Event extent is a sentence within which an event is expressed.

- Event trigger is a word or a phrase that most clearly expresses the event's occurrence. In many cases, the event trigger is the sentence's main verb expressing the event.

- Event arguments are entities that are part of the event. They include participants and attributes.

- Argument role is the relationship between an event and its arguments.

Based on these terminologies, Ahn (2006) proposes to divide the event extraction into sub-tasks: trigger detection, trigger classification, argument detection, and argument classification.

Earlier documents in the case have included embarrassing details about perks **Welch** *received as part of* **his** retirement *package from* **GE** *at a time when corporate scandals were sparking outrage.*

Trigger	retirement
Event type	Personnel:End-Position
Person-Arg	Welch
Entity-Arg	GE
Position-Arg	-
Time-Arg	-
Place-Arg	-

Table 1: Example of a sample in ACE-2005.

2.2 Corpora

The development of event extraction is mainly promoted by the availability of data offered by public evaluation programs such as Message Understanding Conference (MUC), Automatic Content Extraction (ACE), and Knowledge Base Population (TAC-KBP).

Event type	Event subtype
Life	Be-born, Marry, Divorce, Injure, Die
Movement	Transport
Transaction	Transfer-Ownership, Transfer-Money
Business	Start-Org, Merge-Org, Declare-Bankruptcy, End-Org
Conflict	Attack, Demonstrate
Contact	Meet, Phone-Write
Personnel	Start-Position, End-Position, Nominate, Elect
Justice	Arrest-Jail, Release-Parole, Trial-Hearing, Charge-Indict, Sue, Convict, Sentence, Fine, Execute, Extradite, Acquit, Appeal, Pardon

Table 2: List of event types and event subtypes covered in ACE-2005.

Automatic Content Extraction (ACE-2005) is the most widely used corpus in event extraction for English, Arabic, and Chinese. It annotates entities, events, relation, and time (LDC, 2005). There are 7 categories of entities in ACE-2005, i.e., person, organization, location, geopolitical entity, facility, vehicle, and weapon. The ACE-2005 defines 8 event types and 33 event subtypes as presented in table 2. This dataset annotates 599 documents from various sources, e.g., weblogs, broadcast news, newsgroups, and broadcast conversation.

TAC-KBP datasets aim to promote extracting information from unstructured text that fits the knowledge base. The dataset includes the annotation for event detection, event coreference, event linking, argument extraction, and argument linking (Ellis et al., 2015). The event taxonomy in TAC-KBP is mainly derived from ACE-2005, with 9 event types and 38 event subtypes. This dataset contains 360 documents, of which 158 documents are used for training and 202 for testing. The TAC-KBP 2015 contains documents for English only (Ellis et al., 2015), whereas TAC-KBP 2016 includes Chinese and Spanish documents (Ji et al., 2016).

Many corpora for specific domains are available for public use. **MUC** corpus annotates events for domains such as fleet operation, terrorism, and semiconductor production (Grishman and Sundheim, 1996). The **GENIA** is an event detection corpus for the biomedical domain. It is compiled from

scientific documents from PubMed by the BioNLP Shared Task (Kim et al., 2009). **TimeBank** annotates 183 English news articles with event, temporal annotations, and their links (Pustejovsky et al., 2003). Recently, event detection has expands to many other fields such as **CASIE** and **CyberED** for cyber-security (Satyapanich et al., 2020; Trong et al., 2020), **Litbank** for literature (Sims et al., 2019), and music (Ding et al., 2011). However, these corpora are both small in the number of data samples and close in terms of the domain. Consequently, this limits the ability of the pre-trained models to perform tasks in a new domain in real applications.

On the other hand, a general-domain dataset for event detection is a good fit for real applications because it offers a much more comprehensive range of domains and topics. However, manually creating a large-scale general-domain dataset for ED is too costly to anyone ever attempt. Instead, general-domain datasets for event detection have been produced at a large scale by exploiting a knowledge base and unlabeled text. Distant supervision and learning models are the two main methods employed to generate large-scale ED datasets.

Distant supervision (Mintz et al., 2009) is the most widely use with facts derived from existing knowledge base such as WordNet (Miller, 1995), FrameNet (Baker et al., 1998), and Freebase (Bollacker et al., 2008).

Chen et al. (2017) proposes an approach to align key arguments of an event by using Freebase. Then these arguments are used to detect the event and its trigger word automatically. The data is further denoised by using FrameNet (Baker et al., 1998). Similarly, (Wang et al., 2020b) constructs the **MAVEN** dataset from Wikipedia text and FrameNet. This dataset also offers a tree-like event schema structure rooted in the word sense hierarchy in FrameNet.

Similarly, Le and Nguyen (2021) creates **Fed-Semcor** from WordNet and Word Sense Disambiguation dataset. A subset of WordNet synsets that are more likely eventive is collected and grouped into event detection classes with similar meanings. The Semcor is a word sense disambiguation dataset whose tokens are labeled by WordNet synsets. To create the event detection, the text from the Semcor dataset is realigned with the collected ED classes.

Table 3 presents a summary of the existing event extraction dataset for English.

3 Supervised Learning Models

3.1 Feature-based models

In the early stage of event extraction, most methods utilize a large set of features (i.e., feature engineering) for statistical classifiers. The features can be derived from constituent parser (Ahn, 2006), dependency parser (Ahn, 2006), POS taggers, unsupervised topic features (Liao and Grishman, 2011), and contextual features (Patwardhan and Riloff, 2009). These models employ statistical models such as nearest neighbor (Ahn, 2006), maximum-entropy classifier (Liao and Grishman, 2011), and conditional random field (Majumder and Ekbal, 2015).

Ahn (2006) employed a rich feature set of lexical, dependency, and entity features. The lexical features include the word and its lemma, lowercase, and Part-of-Speech (POS) tag. The dependency features include the depth of the word in the dependency tree, the dependency relation of the trigger, and the POS of the connected nodes. The context features include left/right contexts, such as lowercase, POS tag, and entity type. The entity features include the number of dependants, labels, constituent headwords, the number of entities along a dependency path, and the path length to the closest entity.

Ji and Grishman (2008) further introduced cross-sentence and cross-document rules to mandate the consistencies of the classification of triggers and their arguments in a document. In particular, they include (1) the consistency of word sense across sentences in related documents and (2) the consistency of roles and entity types for different mentions of the related events.

Patwardhan and Riloff (2009) suggest using contextual features such as the lexical head of the candidate, the semantic class of the lexical head, lexico-semantic pattern surrounding the candidate. This information provides rich contextual features of the words surrounding the candidate and its lexical-connected words, which provides some signal for the success of convolutional neural networks and graph convolutional neural networks based on the dependency graph in recent studies.

Liao and Grishman (2011) shows that global topic features can help improve EE performance on test data, especially for a balanced corpus. The unsupervised topic model trained on large untagged corpus can provide underlying relations between event and entity types. Therefore, it can reduce

the bias introduced in an imbalanced corpus (e.g., ACE-2005 dataset).

Majumder and Ekbal (2015) extracts various features for biomedical event extraction, such as dependency path and distance to the nearest protein entity. Since the terminologies in the biomedical domain follow some particular rules, the suffix-prefix of words provides substantial semantic information about the terms.

Even though tremendous effort has been poured into feature engineering, feature-based models with statistical classifiers hinder the application of event extraction models in practical situations for two reasons. The first reason is the need for the manual design of the feature set, which requires research expertise in both linguistics and the target-specific domain. Second, since feature extraction tools are imperfect, their incorrect extracted features can harm the statistical models. Hence, a model which can automatically learn would significantly boost the application of event extraction.

3.2   Neural-based models

As mentioned in the previous section, crafting a diverse set of lexical, syntactic, semantic, and topic features require both linguistic and domain expertise. This might hinder the adaptability of the model to real applications where expertise is scarce. Therefore, instead of manually designing linguistic features, automatically extracting features is more practical in virtually every NLP task. Hence, it can revolutionize the common practice of NLP studies. Toward this end, the deep neural network is the perfect match because of its ability to capture features from text automatically.

Deep neural networks employing multiple layers of a large number of artificial neurons have been adapted to various classification and generation tasks. In an artificial neural network, a layer takes input from the output of the lower layer and transforms it into a more abstract representation with two exceptions. The lowest layer takes input as a vector generated from the data sample. The highest layer usually outputs a score for each of the classification classes. These scores are used for the prediction of the label.

3.2.1   Distributed word embedding

Distributed word embedding is one of the most impactful tools for most NLP tasks, including event extraction. Word embedding plays a vital role in transitioning from feature-based to neural-based

modeling. The representation obtained from word embedding captures a rich set of syntactic features, semantic features, and knowledge learned from a large amount of text (Mikolov et al., 2013).

Technically, distributed word embedding is a matrix that can be viewed as a list of low-dimensional continuous float vectors (Bengio et al., 2003). Word embedding maps a word into a single vector within its dictionary. Hence, a sentence can be encoded into a list of vectors. These vectors are fed into the neural network. Among tens of variants, Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) are the most popular word embeddings. These word embeddings were then called context-free embedding to distinguish against contextualized word embedding, which was invented a few years after context-free word embedding.

Contextualized word embedding is one of the greatest inventions in the field of NLP recently. Contrary to context-free word embedding, contextualized embedding encodes the word in a sentence based on the context presented in the text (Peters et al., 2018). In addition, the contextualized embeddings are usually trained on a large text corpus. Hence, its embedding encodes a substantial amount of knowledge from the text. These lead to the improvement of virtually every model in NLP. There have been many variants of contextualized word embedding for general English text e.g. BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019b), multi-lingual text e.g. mBERT (Devlin et al., 2019), XLM-RoBERTa (Ruder et al., 2019), scientific document SciBERT (Beltagy et al., 2019), and text generation e.g. GPT2 (Radford et al., 2019).

3.2.2   Convolutional Neural Networks

Nguyen and Grishman (2015) employed a convolutional neural network, inspired by CNNs in computer vision (LeCun et al., 1998) and NLP (Kalchbrenner et al., 2014), that automatically learns the features from the text, and minimizes the effort spent on feature extraction. Instead of producing a large vector representation for each sample, i.e., tens of thousands of dimensions, this model employs three much smaller word embedding vectors with just a few hundred dimensions. As shown in Figure 1, given a sentence with marked entities, each word in the sentence is represented by a low-dimension vector concatenated from (1)the word embedding, (2) the relative position embedding, and (3) the entity type embedding. The vectors
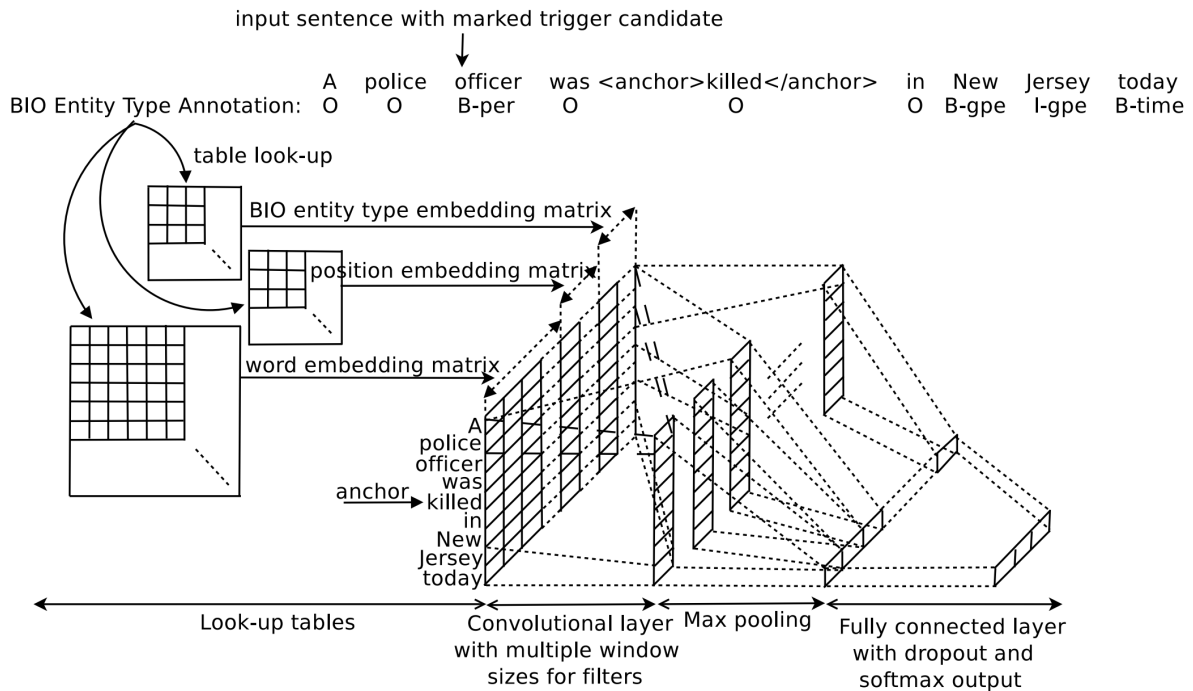
input sentence with marked trigger candidate

A police officer was <anchor>killed</anchor> in New Jersey today
BIO Entity Type Annotation: O O B-per O O O B-gpe I-gpe B-time

table look-up

BIO entity type embedding matrix

position embedding matrix

word embedding matrix

A
police
officer
was
anchor → killed
in
New
Jersey
today

Look-up tables | Convolutional layer with multiple window sizes for filters | Max pooling | Fully connected layer with dropout and softmax output

Figure 1: The convolutional neural network model for event detection in (Nguyen and Grishman, 2015).

of words then form a matrix working as the representation of the sentence. The matrix is then fed to multiple stacks of a convolutional layer, a max-pooling layer, and a fully connected layer. The model is trained using the gradient descent algorithm with cross-entropy loss. Some regularization techniques are applied to improve the model, such as mini-batch training, adaptive learning rate optimizer, and weight normalization.

Many efforts have introduced different pooling techniques to extract meaningful information for event extract from what is provided in the sentence. Chen et al. (2015) improved the CNN model by using multi-pooling (DMCNN) instead of vanilla max-pooling. In this model, the sentence is split into multiple parts by either the examining event trigger or the given entity markers. The pooling layer is applied separately on each part of the sentence. Zhang et al. (2016) proposed skip-window convolution neural networks (S-CNNs) to extract global structured features. The model effectively captures the global dependencies of every token in the sentence. Li et al. (2018) proposed a parallel multi-pooling convolutional neural network (PM-CNN) that applies not only multiple pooling for the examining event trigger and entities but also to every other trigger and argument that appear in the sentence. This helps to capture the compositional

semantic features of the sentence.

Kodelja et al. (2019) integrated the global representation of contexts beyond the sentence level into the convolutional neural network. To generate the global representation in connection with the target event detection task, they label the whole given document using a bootstrapping model. The bootstrapping model is based on the usual CNN model. The predictions for every token are aggregated to generate the global representation.

Even though CNN, together with the distributed word representations, can automatically capture local features, EE models based on CNN are not successful at capturing long-range dependency between words. The reason is that CNN can only model the short-range dependencies within the window of its kernel. Moreover, a large amount of information is lost because of the pooling operations (e.g., max pooling). As such, a more sophisticated neural network design is needed to model the long-range dependency between words in long sentences and documents without sacrificing information.

3.2.3 Recurrent Neural Networks

Nguyen et al. (2016a) employed Gated Recurrent Unit (GRU) (Cho et al., 2014), a RNN-based architecture, to better model relation between words in a sentence. The model produces a rich representa-
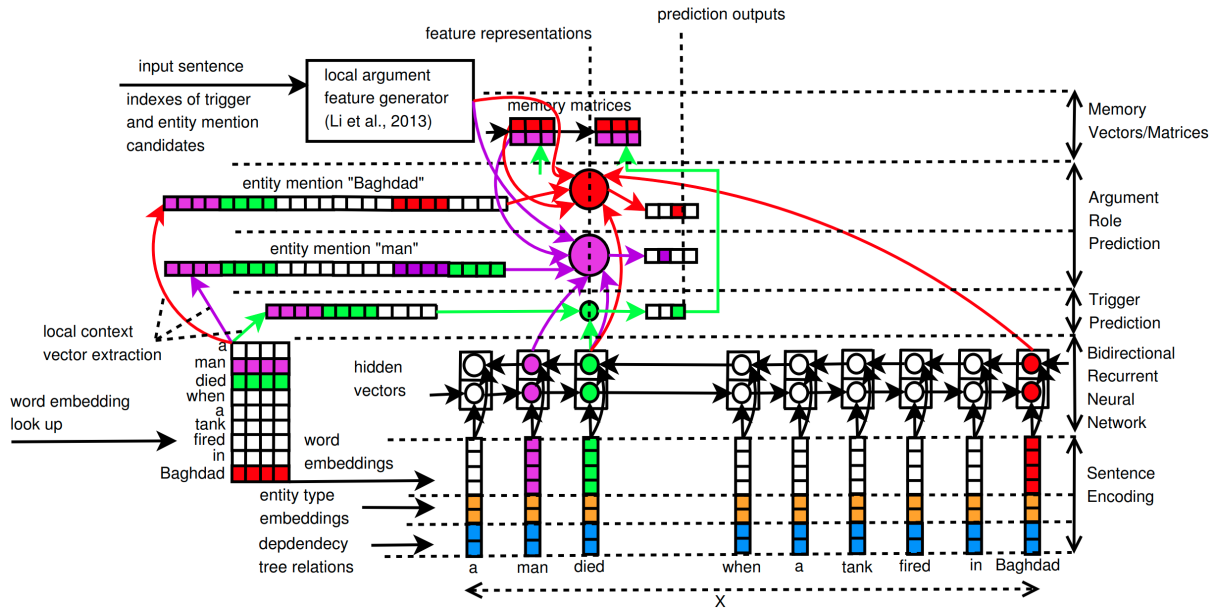
Figure 2: The joint EE model (Nguyen et al., 2016b) for the input sentence "a man died when a tank fired in Baghdad" with local context window d = 1.

tion based on the context captured in the sentence for the prediction of event triggers and event arguments. The model includes two recurrent neural networks, one for the forward direction and one for the backward direction. A diagram of the whole model is presented in Figure 2.

Sentence embedding: Similar to CNN model, each word w_i of the sentence is transformed into a fixed-size real-value vector x_i. The feature vector is a concatenation of the word embedding vector of the current word, the embedding vector for the entity type of the current word, and the one-hot vector whose dimensions correspond to the possible relations between words in the dependency trees.

RNN encoding: The model employs two recurrent networks, forward and backward, denoted as $\overrightarrow{RNN}$ and $\overleftarrow{RNN}$ to encode the sentence word-by-word:

$$(a_1, \cdots, a_N) = \overrightarrow{RNN}(x_1, \cdots, x_N)$$

$$(a'_1, \cdots, a'_N) = \overleftarrow{RNN}(x_1, \cdots, x_N)$$

Finally, the representation h_i for each word is the concatenation of the corresponding forward and backward vectors $h_i = [a_i, a'_i]$.

Prediction: To jointly predict the event triggers and arguments, a binary vector for trigger and two binary matrices are introduced for event arguments. These vectors and matrices are initialized to zero. For each iteration, according to each word w_i, the

prediction is made in a 3-step process: trigger prediction for w_i, argument role prediction for all the entity mentions given in the sentence, and finally, compute the vector and matrices of the current step using the memory and the output of the previous step.

Similarly, Ghaeini et al. (2016) and Chen et al. (2016) employed Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997), another architecture based on RNN. LSTM is much more complex than the original RNN architecture and the GRU architecture. LSTM can capture the semantics of words with consideration of the context given by the context words automatically. Chen et al. (2016) further proposed Dynamic Multi-Pooling similar to the DMCNN (Chen et al., 2015) to extract event and argument separately. Furthermore, the model proposed a tensor layer to model the interaction between candidate arguments.

Even though the vanilla LSTM (or sequential/linear LSTM) can capture a longer dependency than CNN, in many cases, the event trigger and its arguments are distant. As such, the LSTM model can not capture the dependency between them. However, the distance between those words is much shorter in a dependency tree. Using a dependency tree to represent the relationship between words in the sentence can bring the trigger and entities close to each other. Some studies have implemented this structure in various ways. Sha et al. (2018) proposed to enhance the bidi-

rectional RNN with dependency bridges, which channel the syntactic information when modeling words in the sentence. The paper illustrates that simultaneously employing hierarchical tree structure and sequence structure in RNN improves the model's performance against the conventional sequential structure. Li et al. (2019) introduced tree a knowledge base (KB)-driven tree-structured long short-term memory networks (Tree-LSTM) framework. This model incorporates two new features: dependency structures to capture broad contexts and entity properties (types and category descriptions) from external ontologies via entity linking.

3.3 Graph Convolutional Neural Networks

The presented CNN-based and LSTM-based models for event detection have only considered the sequential representation of sentences. However, in these models, graph-based representation such as syntactic dependency tree (Nivre et al., 2016) has not been explored for event extraction, even though they provide an effective mechanism to link words to their informative context in the sentences directly.

For example, Figure 4 presents the dependency tree of the sentence "*This LPA-induced rapid phosphorylation of radixin was significantly suppressed in the presence of C3 toxin, a potent inhibitor of Rho*". In this sentence, there is a event trigger "*suppressed*" with its argument "*C3 toxin*". In the sequential representation, these words are 5-step apart, whereas in the dependency tree, they are 2-step apart. This example demonstrates the potential of the dependency tree in extracting event triggers and their arguments.

Many EE studies have widely used graph convolutional neural networks (GCN) (Kipf and Welling, 2017). It features two main ingredients: a convolutional operation and a graph. The convolutional operation works similarly in both CNNs and GCNs. It learns the features by integrating the features of the neighboring nodes. In GCNs, the neighborhoods are the adjacent nodes on the graph, whereas, in CNNs, the neighborhoods are surrounding words in linear form.

Formally, let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph, and A be its adjacency matrix. The output of the $l + 1$ convolutional layer on a graph $\mathcal{G}$ is computed based on the hidden states $H^l = \{h_i^l\}$ of the l-th layer as
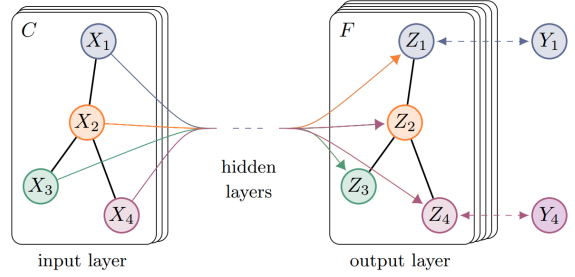

Figure 3: Graph convolutional neural network (Kipf and Welling, 2017).

follows:

$$h_i^{l+1} = \sigma \sum_{(i,j) \in \mathcal{E}} \alpha_{ij}^l W^l h_j^l + b^l \qquad (1)$$

Or in matrix form:

$$H^{l+1} = \sigma(\alpha^l W^l H^l A + b^l) \qquad (2)$$

where W and b are learnable parameters and σ is a non-linear activation function; α_{ij} is the weight for the edge ij, in the simplest way, $\alpha_{ij} = 1$ for all edges.

GCN-ED (Nguyen and Grishman, 2018) and JMEE (Liu et al., 2018) models are the first to use GCN for event detection. The graph used in the model is based on a transformation of the syntactic dependency tree. Let $\mathcal{G}_{\text{dep}} = (\mathcal{V}, \mathcal{E}_{\text{dep}})$ be an acyclic directed graph, representing the syntactic dependency tree of a given sentence. $\mathcal{V} = \{w_i | i \in [1, N]\}$ is the set of nodes; $\mathcal{E}_{\text{dep}} = \{(w_i, w_j) | i, j \in [1, N]\}$ is the set of edges. Each node of the graph represents a token in the given sentence, whereas each directed edge represents a syntactic arc in the dependency tree. The graph G used in GCN-ED and JMEE is derived with two main improvements:

- For each node w_i, a self-loop edge (w_i, w_i) is added to the set of edges so that the representation of the node is computed of the representation of itself.

- For each edge (w_i, w_j), a reverse edge (w_j, w_i) of the same dependency type is added to the set of edges of the graph.

Mathematically, a new set of edge $\mathcal{E}$ is created as follows:

$$\mathcal{E} = \mathcal{E}_{\text{dep}} \cup \{(w_i, w_i) | w_i \in \mathcal{V}\}$$
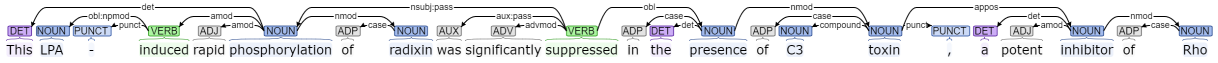$$\cup \{(w_j, w_i) | (w_i, w_j) \in \mathcal{E}_{\text{dep}}\}$$

Figure 4: Dependency tree for sentence "*This LPA-induced rapid phosphorylation of radixin was significantly suppressed in the presence of C3 toxin, a potent inhibitor of Rho*", parsed by Trankit toolkit[1].

Once the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is created, the convolutional operation, as shown in Equation 1 is applied multiple times on the input word embedding. Due to the small scale of the ED dataset, instead of using different sets of weights and biases for each dependency relation type, Nguyen and Grishman (2018) used only three sets of weights and biases for three types of dependency edges based on their origin: (i) The original edges from $\mathcal{E}_{\text{dep}}$, (ii) the self-loop edges, (iii) the inverse edges.

In the dependency graph, some neighbors of a node could be more important for event detection than others. Inspired by this, Nguyen and Grishman (2018) and Liu et al. (2018) also introduced neighbor weighting (Marcheggiani and Titov, 2017), in which neighbors are weighted differently depending on the level of importance. The weight α in Equation 1 is computed as follow:

$$\alpha_{ij}^l = \sigma(h_j^l W_{\text{type(i,j)}}^l) + b^l)$$

where h_j^l is the representation of the j-th words at the l-th layer. $W_{\text{type(i,j)}}^l$ and b^l are weight and bias terms, and σ is a non-linear activation function.

However, the above dependency-tree-based methods explicitly use only first-order syntactic edges, although they may also implicitly capture high-order syntactic relations by stacking more GCN layers. As the number of GCN layers increases, the representations of neighboring words in the dependency tree will get more and more similar since they all are calculated via those of their neighbors in the dependency tree, which damages the diversity of the representations of neighboring words. As such, Yan et al. (2019) introduced Multi-Order Graph Attention Network for Event Detection (MOGANED). In this model, the hidden vectors are computed based on the representations of not only the first-order neighbors but also higher-order neighbors in the syntactic dependency graph. To do that, they used Graph Attention network (GAT) (Veličković et al., 2018) and an attention aggregation mechanism to merge its multi-order representations.

In a multi-layer GCN model, each layer has its scope of neighboring. For example, the rep-

resentation of a node in the first layer is computed from the representations of its first-order neighbors only, whereas one in the second layer is computed from the representations of both first-order and second-order neighbors. As such, Lai et al. (2020b) proposed GatedGCN with an enhancement to the graph convolutional neural network with layer diversity using a gating mechanism. The mechanism helps the model to distinguish the information derived from different sources, e.g., first-order neighbors and second-order neighbors. The authors also introduced importance score consistency between model-predicted importance scores and graph-based importance scores. The graph-based importance scores are computed based on the distances between nodes in the dependency graph.

The above GCN-based models usually ignore dependency label information, which conveys rich and useful linguistic knowledge for ED. Edge-Enhanced Graph Convolution Network (EE-GCN), on the other hand, simultaneously exploited syntactic structure and typed dependency label information (Cui et al., 2020). The model introduces a mechanism to dynamically update the representation of node-embedding and edge-embedding according to the context presented in the neighboring nodes. Similarly, Dutta et al. (2021) presented the GTN-ED model that enhanced prior GCN-based models using dependency edge information. In particular, the model learns a soft selection of edge types and composite relations (e.g., multi-hop connections, called meta-paths) among the words, thus producing heterogeneous adjacency matrices.

3.4 Knowledge Base

As mentioned before, event extraction extract events from the text that involves some named entities such as participants, time, and location. In some domains, such as the biomedical domain, it requires a broader knowledge acquisition and a deeper understanding of the complex context to perform the event extraction task. Fortunately, a large number of those entities and events have been recorded in existing knowledge bases. Hence, these knowledge bases may provide the model with a concrete background of the domain terminologies

as well as their relationship. This section presents some methods to exploit external knowledge to enhance event extraction models.

Li et al. (2019) proposed a model to construct knowledge base concept embedding to enrich the text representation for the biomedical domain. In particular, to better capture domain-specific knowledge, the model leverages the external knowledge bases (KBs) to acquire properties of all the biomedical entities. Gene Ontology is used as their external knowledge base because it provides detailed gene information, such as gene functions and relations between them as well as gene product information, e.g., related attributes, entity names, and types. Two types of information are extracted from the KB to enrich the feature of the model: (1) entity type and (2) gene function description. First, the entity type for each entity is queried, then it is injected into the model similar to (Nguyen and Grishman, 2015). Second, the gene function definition, which is usually a long phrase, is passed through a language model to obtain the embedding. Finally, the embedding is concatenated to the input representation of the LSTM model.

Huang et al. (2020a), on the other hand, argues that the word embedding does not provide adequate clues for event extraction in extreme cases such as non-indicative trigger words and nested structures. For example, in the biomedical domain, many entities have hierarchical relations that might help to provide domain knowledge to the model. In particular, the Unified Medical Language System (UMLS) is the knowledge base that is used in this study. UMLS provides a large set of medical concepts, their pair-wise relations, and relation types. To incorporate the knowledge, words in the sentence are mapped to the set of concepts, if applicable. Then they are connected using the relations provided by the KB to form a semantic graph. This graph is then used in their graph neural network.

3.5 Data generation

As shown in Section 2.2, most of the datasets for Event Extraction were created based on human annotation, which is very laborious. As such, these datasets are limited in size, as shown in Table 3. Moreover, these datasets are usually extremely imbalanced. These issues might hinder the learning process of the deep neural network. Many methods of data generation have been introduced to enlarge the EE datasets, which results in significant improvement in the performance of the EE model.

External knowledge bases such as Freebase, Wikipedia, and FrameNet are commonly used in event generation. Liu et al. (2016) trained an ED model on the ACE dataset to predict the event label on FrameNet text to produce a semi-supervised dataset. The generated data was then further filtered using a set of global constraints based on the original annotated frame from FrameNet. Huang et al. (2016), on the other hand, employs a word-sense disambiguation model to predict the word-sense label for unlabeled text. Words that belong to a subset of verb and noun senses are considered as trigger words. To identify the event arguments for the triggers, the text is parsed into an AMR graph that provides arguments for trigger candidates. The argument role is manually mapped from AMR argument types. Chen et al. (2017); Zeng et al. (2018) proposed to automatically label training data for event extraction based on distant supervision via Freebase, Wikipedia, and FrameNet data. The Freebase provides a set of key arguments for each event type. After that, candidate sentences are searched among Wikipedia text for the appearances of key arguments. Given the sentence, the trigger word is identified by a strong heuristic rule.

Ferguson et al. (2018) proposed to use bootstrapping for event extraction. The core idea is based on the occurrence of multiple mentions of the same event instances across newswire articles from multiple sources. Hence, if an ED model detects some event mentions at high confidence from a cluster, the model can then acquire diverse training examples by adding the other mentions from that cluster. The authors trained an ED model based on limited available training data and then used that model for data labeling on unlabeled newswire text.

Yang et al. (2019) explored the method that uses a generative model to generate more data. They generated data from the golden ACE dataset in three steps. First, the arguments in a sentence are replaced with highly similar arguments found in the golden data to create a noisy sentence. Second, a language model is used to regenerate the sentence from the noisy generated sentence to create a new smoother sentence to avoid overfitting. Finally, the candidate sentences are ranked using a perplexity score to find the best-generated sentence.

Tong et al. (2020) argued that open-domain trigger knowledge could alleviate the lack of data and training data imbalance in the existing EE dataset.

The authors proposed a novel Enrichment Knowledge Distillation (EKD) model that can generate noisy ED data from unlabeled text. Unlike the prior methods that employed rules or constraints to filter noisy data, their model used the teacher-student model to automatically distill the training data.

3.6 Document-level Modeling

The methods for event extraction mentioned so far have not gone beyond the sentence level. Unfortunately, this is a systematic problem as, in reality, events and their associated arguments can be mentioned across multiple sentences in a document (Yang et al., 2018). Hence, such sentence-level event extraction methods struggle to handle documents in which events and their arguments scatter across multiple sentences. The document-level event extraction (DEE) paradigm has been investigated to address the problem of sentence-level event extraction. Many researchers have proposed methods to model document-level relations such as entity interactions, sentence interactions (Huang and Jia, 2021; Xu et al., 2021), reconstruct document-level structure (Huang and Peng, 2021), and model long-range dependencies while encoding a lengthy document (Du and Cardie, 2020).

Initial studies for DEE did not consider modeling the document-level relation properly. Yang et al. (2018) was the first attempt to explore the DEE problem on a Chinese Financial Document corpus (ChiFinAnn) by generating weakly-supervised EE data using distant supervision. Their model performs DEE in two stages. First, a sequence tagging model extracts events at the sentence level in every document sentence. Second, key events are detected among extracted events, and arguments are heuristically collected from all over the document. Zheng et al. (2019), on the other hand, proposed an end-to-end model named Doc2EDAG. The model encodes documents using a transformer-based encoder. Instead of filling the argument table, they created an entity-based directed acyclic graph to find the argument effectively through path expansion. Du and Cardie (2020) transforms the role filler extraction into an end-to-end neural sequence learning task. They proposed a multi-granularity reader to efficiently collect information at different levels of granularity, such as sentence and paragraph levels. Therefore, it mitigates the effect of long dependencies of scattering argument in DEE.

Some studies have attempted to exploit the re-

lationship between entities, event mentions, and sentences of the document. Huang and Jia (2021) modeled the interactions between entities and sentences within long documents. In particular, instead of constructing an isolated graph for each sentence, this paper constructs a unified unweighted graph for the whole document by exploiting the relationship between sentences. Furthermore, they proposed the sentence community consisting of sentences related to the same event's arguments. The model detects multiple event mentions by detecting those sentence communities. To encourage the interaction between entities, Xu et al. (2021) proposed a Heterogeneous Graph-based Interaction Model with a Tracker (GIT) to model the global interaction between entities in a document. The graph leverages multiple document-level relations, including sentence-sentence edges, sentence-mention edges, intra mention-mention edges, and inter mention-mention edges. Huang and Peng (2021) introduced an end-to-end model featuring a structured prediction algorithm, Deep Value Networks, to efficiently model cross-event dependencies for document-level event extraction. The model jointly learns entity recognition, event co-reference, and event extraction tasks, resulting in a richer representation and a more robust model.

3.7 Joint Modeling

The above works have executed the four subtasks of event extraction in a pipeline where the model uses the prediction of other models to perform its task. Consequently, the errors of the upstream subtasks are propagated through the downstream subtasks in the pipeline, ruining their performances. Additionally, the knowledge learned from the downstream subtasks can not influence the prediction decision of the upstream subtasks. Thus, the dependence on the tasks can not be exploited thoroughly. To address the issues of the pipeline model, joint modeling of multiple event extraction subtasks is an alternative to take advantage of the interactions between the EE subtasks. The interactions between subtasks are bidirectional. Therefore, useful information can be carried across the subtasks to alleviate error propagation.

Joint modeling can be used to train a diverse set of subtasks. For example, Lee et al. (2012) trained a joint model for event co-reference resolution and entity co-reference resolution, while Han et al. (2019) proposed a joint model for event

detection and event temporal relation extraction. In the early day, modeling event detection and argument role extraction together are very popular (Li et al., 2013; Venugopal et al., 2014; Nguyen et al., 2016a). Recent joint modeling systems have trained models with up to 4 subtasks (i.e. event detection, entity extraction, event argument extraction, and entity linking) (Lin et al., 2020; Zhang and Ji, 2021; Nguyen et al., 2021, 2022). Table 4 presents a summary of the subtasks that were used for joint modeling for EE.

Early joint models were simultaneously trained to extract the trigger mention and the argument role (Li et al., 2013), Li et al. (2013) formulated a two-task problem as a structural learning problem. They incorporated both global features and local features into a perceptron model. The trigger mention and arguments are decoded simultaneously using a beam search decoder. Later models that are based on a neural network share a sentence encoder for all the subtasks (Nguyen et al., 2016a; Han et al., 2019; Wadden et al., 2019) so that the training signals of different subtasks can impact the representation induced by the sentence encoder.

Besides the shared encoders, recent models use various techniques to encourage interactions between subtasks. Nguyen et al. (2016a) employed a memory matrix to memorize the dependencies between event and argument labels. These memories are then used as a new feature in the trigger and argument prediction. They employed three types of dependencies: (i) trigger subtype dependency, (ii) argument role dependency, and (iii) trigger-argument role dependency. These terminologies were later generalized as intra/inter-subtask dependencies (Lin et al., 2020; Nguyen et al., 2021, 2022).

Luan et al. (2019) proposed the DyGIE model that employed an interactive graph-based propagation between events and entities nodes based on entity co-references and entity relations. In particular, in DyGIE model (Luan et al., 2019), the input sentences are encoded using a BiLSTM model, then, a contextualized representation is computed for each possible text span. They employed a dynamic span graph whose nodes are selectively chosen from the span pool. At each training step, the model updates the set of graph nodes. It also constructs the edge weights for the newly created graph. Then, the representations of spans are updated based on neighboring entities and connected relations. Finally, the

predictions of entities, events, and their relations are based on the latest representations. Figure 5 presents the diagram of the DyGIE model. Wadden et al. (2019) further improved the model with contextualized embeddings BERT while maintaining the core architecture of DyGIE. Even though these models have introduced task knowledge interaction through graph propagation, their top task prediction layers still make predictions independently. In other words, the final prediction decision is still made locally.

To address the DyGIE/DyGIE++ issue, OneIE model (Lin et al., 2020) proposed to enforce global constraints to the final predictions. They employed a beam search decoder at the final prediction layer to globally constrain the predictions of the subtasks. Similar to JREE model (Nguyen et al., 2016a), they considered both cross-subtask interactions and cross-instance interactions. To do that, they designed a set of global feature templates to capture both types of interactions. Given all the templates, the model tries to fill all possible features and learns the weights. To make the final prediction, a trivial solution is an exhaustive search during the inference. However, the search space grows exponentially, leading to an infeasible problem. They proposed a graph-based beam search algorithm to find the optimal graph. In each step, the beam grows with either a new node (i.e., a trigger or an entity) or a new edge (i.e., an argument role or an entity relation).

In the above neural-based models, the predictive representation of the candidates is computed independently using contextualized embedding. Consequently, the predictive representation has not considered the representations of the other related candidates. FourIE model (Nguyen et al., 2021) features a graph structure to encourage interactions between related instances of a multi-task EE problem. Nguyen et al. (2021) further argued that the global feature constraint in OneIE (Lin et al., 2020) is suboptimal because it is manually created. They instead introduced an additional graph-based neural network to score the candidate graphs. To train this scoring network, they employ Gumbel-Softmax distribution (Jang et al., 2017) to allow gradient updates through the discrete selection process. However, due to the heuristical design of the dependency graph, the model may fail to explore other possible interactions between the instances. As such, Nguyen et al. (2022) explicitly model the de-
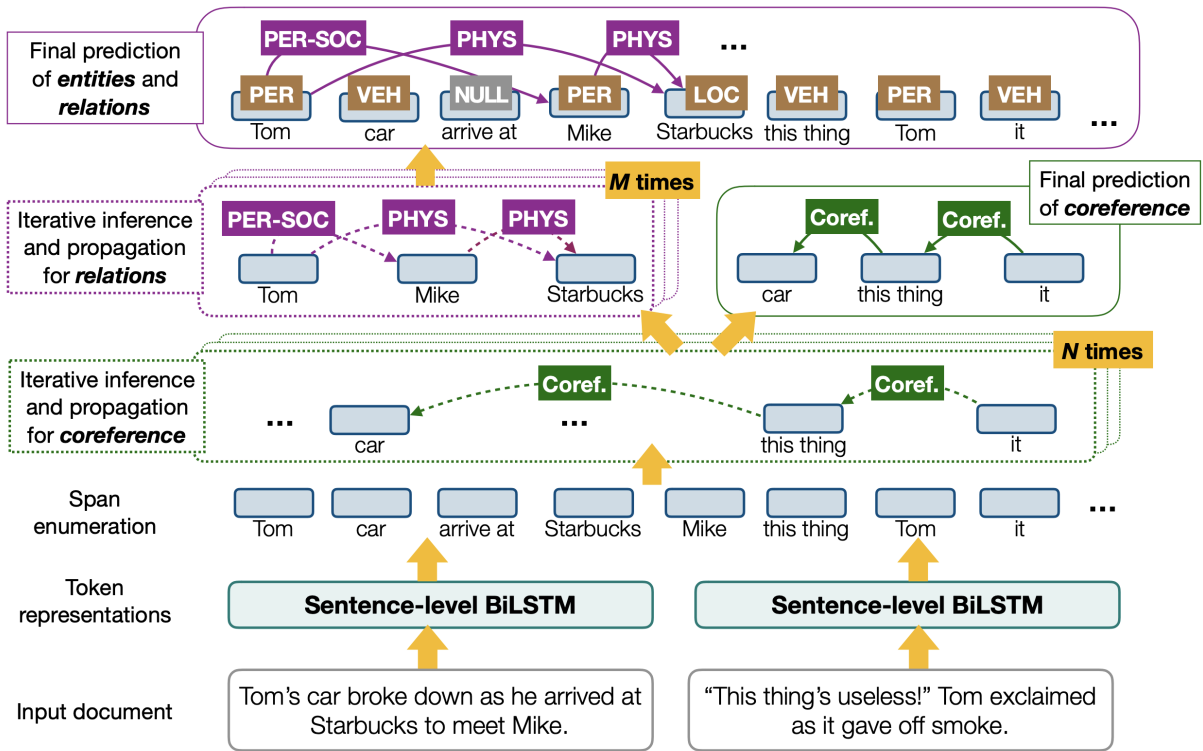
Figure 5: Architecture of the DyGIE model (Luan et al., 2019).

pendencies between tasks by modeling each task instance as a node in the fully connected dependency graph. The weight for each edge is learnable, allowing a soft interaction between instances instead of hard interactions in prior works (Lin et al., 2020; Zhang and Ji, 2021; Nguyen et al., 2021)

Recently, joint modeling for event extraction was formulated as a text generation task using pretrained generative language models such as BART (Lewis et al., 2020), and T5 (Raffel et al., 2020). In these models (Lu et al., 2021; Hsu et al., 2022), the event mentions, entity mentions, as well as their labels and relations are generated by an attention-based autoregressive decoder. The task dependencies are encoded through the attention mechanism of the transformer-based decoder. This allows the model to learn the dependencies between tasks and task instances flexibly. However, to train the model, they have to assume an order of tasks and task instances that are being decoded. As a result, the model suffers from the same problem that arose in pipeline models.

Dataset	Topic	#Class	#Samples	#Language	Tasks
Event Extraction					
ACE (LDC, 2005)	News	33	4,907 events	3	Trig, Arg, Ent, Rel, Coreference
TAC-KBP (Ellis et al., 2015)	News	38	11,975 events	3	Trig, Arg, Ent, Rel, Coreference
TimeBank (Pustejovsky et al., 2003)	Newswire	8	7,935 events	1	Trig Temporal
GENIA (Ohta et al., 2002)	Biomedical	36	36,114 events	1	Trig, Arg, Ent, Rel
CASIE (Satyapanich et al., 2020)	Cyber security	5	8,470 events	1	Trig
CyberED (Man Duc Trong et al., 2020)	Cyber security	30	8,014 events	1	Trig
Litbank (Sims et al., 2019) (Bamman et al., 2019, 2020)	Literature	1	7,849 events	1	Trig, Ent EntCoref
BRAD (Lai et al., 2021b)	Black rebellion	12	4,259 events	1	Trig, Arg, Ent, Rel
SuicideED (Guzman-Nateras et al., 2022)	Mental health	7	36,978 events	1	Trig, Arg, Ent, Rel
MAVEN (Wang et al., 2020b)	General	168	111,611 events	1	Trig
FedSemcor (Le and Nguyen, 2021)	General	449	34,666 events	1	Trig
MINION (Pouran Ben Veyseh et al., 2022)	Economy, Politics, Technology, Nature, Crime, Military	33	50,934	10	Trig
CLIP-Event (Li et al., 2022)	News	33	105,331 events	1	Trig, Arg, Ent
Event Relation					
HiEve (Glavaš et al., 2014)	News stories	2	2,257 pairs	1	Hierarchy, Coreference
TempEval (UzZaman et al., 2013)	News		pairs		Temporal
EventStoryLine (Caselli and Vossen, 2017)	Calamity events	2	8,201 pairs	1	Causal, Temporal
MECI (Lai et al., 2022b)	Wikipedia	2	11,055 pairs	5	Causal
mSubEvent (Lai et al., 2022a)	Wikipedia	2	3,944 pairs	5	Hierarchy

Table 3: Statistics of existing event extraction datasets. Event-related tasks: Trigger Identificaion & Classification (Trig), Event Argument Extraction (Arg), Event Temporal (Temporal), Event Causality (Causal), Event Coreference (Coreference), Event Hierarchy (Hierarchy). Entity-related tasks: Entity Mention (Ent), Entity Linking (Rel), Entity Coreference (EntCoref).

Model Acronym	System	Event	Entity	Argument	Relation	EventCoref	EntityCoref	EventTemp
Lee's Joint	Lee et al. (2012)					✓	✓	
Li's Joint	Li et al. (2013)	✓		✓				
MLN+SVM	Venugopal et al. (2014)	✓		✓				
Araki's Joint	Araki and Mitamura (2015)	✓				✓		
JRNN	Nguyen et al. (2016a)	✓	✓	✓				
Structure Joint	Han et al. (2019)	✓						✓
DyGIE	Luan et al. (2019)	✓	✓		✓			
DyGIE++	Wadden et al. (2019)	✓	✓		✓			
HPNet	Huang et al. (2020b)	✓		✓				
OneIE	Lin et al. (2020)	✓	✓	✓	✓			
NGS	Wang et al. (2020a)	✓		✓				
Text2Event	Lu et al. (2021)	✓		✓				
AMRIE	Zhang and Ji (2021)	✓	✓	✓	✓			
FourIE	Nguyen et al. (2021)	✓	✓	✓	✓			
DEGREE	Hsu et al. (2022)	✓		✓				
GraphIE	Nguyen et al. (2022)	✓	✓	✓	✓			

Table 4: Subtasks for joint modeling in event extraction.

4 Low-resource Event Extraction

State-of-the-art event extraction approaches, which follow the traditional supervised learning paradigm, require great human efforts to create high-quality annotation guidelines and annotate the data for a new event type. For each event type, language experts need to write annotation guidelines that describe the class of event and distinguish it from the other types. Then annotators are trained to label event triggers in the text to produce a large dataset. Finally, a supervised-learning-based classifier is trained on the obtained event triggers to label the target event. This labor-exhaustive process might limit the applications of event extraction in real-life scenarios. As such, approaches that require less data creation are becoming more and more attractive thanks to their fast deployment and low-cost solution. However, this line of research faces a challenging wall due to their limited access to labeled data. This section presents recent studies on low-resource event extraction in various learning paradigms and domains. The rest of the section is organized as follow: Section 4.1 highlights some methods of zero-shot learning; section 4.2 presents a new clusters of recent studies in few-shot learning. Finally, methods for cross-lingual event extraction is presented in section 4.3.

4.1 Zero-shot Learning

Zero-shot learning (ZSL) is a type of transfer learning in which a model performs a task without any training samples. Toward this end, transfer learning uses a pre-existing classifier to build a universal concept space for both seen and unseen samples. Existing methods for event extraction exploits latent-variable space in CRF model (Lu and Roth, 2012), rich structural features such as dependency tree and AMR graph (Huang et al., 2018a), ontology mapping (Zhang et al., 2021), and casting the problem into a question-answering problem (Liu et al., 2020a; Lyu et al., 2021).

The early study by Lu and Roth (2012) showed the first attempt to solve the event extraction problem under zero-shot learning. They proposed to model the problem using latent variable semi-Markov conditional random fields. The model jointly extracts event mentions and event arguments given event templates, coarse event/entity mentions, and their types. They used a framework called structured Preference Modeling (PM). This framework allows arbitrary preferences associated with

specific structures during the training process.

Inspired by the shared structure between events, Huang et al. (2018a) introduced a transfer learning method that matches the structural similarity of the event in the text. They proposed a transferable architecture of structural and compositional neural networks to jointly produce to represent event mentions, their types, and their arguments in a shared latent space. This framework allows for predicting the semantically closest event types for each event mention. Hence, this framework can be applied to unseen event types by exploiting the limited manual annotations. In particular, event and argument candidates are detected by exploiting the AMR graph of the sentence. Figure 6 presents an example of the AMR graph. After this, a CNN is used to encode all the triplets representing AMR edges, e.g. (dispatch-01, :ARG0, China). For each new event type, the same CNN model encodes the relations between event type, argument role, and entity type, e.g. (Transport Person, Destination), resulting in a representation vector for the new event ontology. The model chooses the closest event type based on the similarity score between the trigger's encoded vector and all available event ontology vectors to predict the event type for a candidate event trigger.

Zhang et al. (2021) proposed a zero-shot event extraction method that (1) extracts the event mentions using existing tools, then, and (2) maps these events to the targeted event types with zero-shot learning. Specifically, an event-type representation is induced by a large pre-trained language model using the event definition for each event type. Similarly, event mentions and entity mentions are encoded into vectors using a pre-trained language model. Initial predictions are obtained by computing the cosine similarities between label and event representations. To train the model, an ILP solver is employed to regulate the predictions according to the given ontology of each event type. In detail, they used the following constraints: (1) one event type per event mention, (2) one argument role per argument, (3) different arguments must have different types, (4) predicted triggers and argument types must be in the ontology, and (5) entity type of the argument must match the requirement in the ontology.

Thanks to the rapid development of large generative language models, a language model can embed texts and answer human-language questions in a human-friendly way using its large deep knowl-
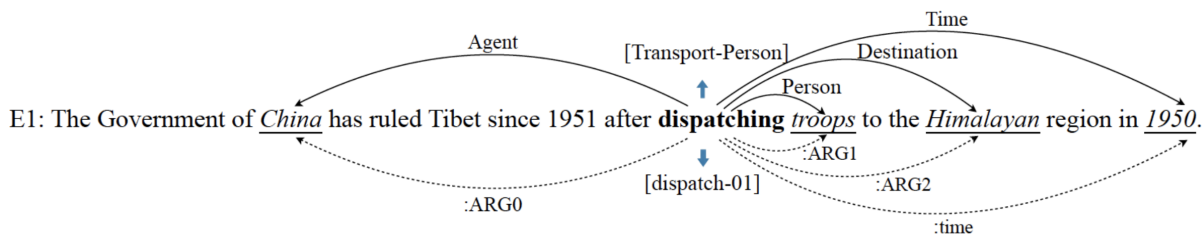
Figure 6: Comparison of AMR graph and event schema(Huang et al., 2018a). The word *dispatching* is an event trigger of the event Transport-Person with 4 arguments. The upper solid lines demonstrate the actual event arguments. The lower dash lines represent the edges in the AMR graph.

edge obtained from massive training data. Liu et al. (2020a) proposed a new learning setting of event extraction. They cast it as a machine reading comprehension problem (MRC). The modeling includes (1) an unsupervised question generation process, which can transfer event schema into a set of natural questions, AND (2) a BERT-based question-answering process to generate the answers as EE results. This learning paradigm exploits the learned knowledge of the language model and strengthens EE's reasoning process by integrating sophisticated MRC models into the EE model. Moreover, it can alleviate the data scarcity issue by transferring the knowledge of MRC datasets to train EE models. Lyu et al. (2021), on the other hand, explore the Textual Entailment (TE) task and/or Question Answering (QA) task for zero-shot event extraction. Specifically, they cast the event trigger detection as a TE task, in which the TE model predicts the level of entailment of a hypothesis (e.g., *This is about a birth event* given a premise, i.e., the original text. Since an event may associate with multiple arguments, they cast the event argument extraction into a QA task. Given an input text and the extracted event trigger, the model is asked a set of questions based on the event type definition in the ontology, and retrieve the QA answers as predicted argument.

4.2 Few-shot Learning

There are several ways of modeling the event detection in a few-shot learning scheme (FSL-ED): (1) token classification FSL-ED and (2) sequence labeling FSL-ED.

Most of the studies following token classification setting (Bronstein et al., 2015; Peng et al., 2016; Lai and Nguyen, 2019; Deng et al., 2020; Lai et al., 2020a) are based on a prototypical network (Snell et al., 2017), which employs a general-purpose event encoder for embed event candidates while the predictions are done using a non-parameterized

metric-based classifier. Figure 7 demonstrates the idea of a prototypical network. Since the classifiers are non-parametric, these studies mainly explore the methods to improve the event encoder.

Bronstein et al. (2015) were among the first working in few-shot event detection. They proposed a different training/evaluation for event detection with minimal supervision. They proposed an alternative method, which uses the trigger terms included in the annotation guidelines as seeds for each event type. The model consists of an encoder and a classifier. The encoder embeds a trigger candidate into a fix-size embedding vector. The classifier is an event-independent similarity-based classifier. In this paper, the authors argue that they can eliminate the costly manual annotation for new event types. At the same time, the non-parametric classifier does not require a large amount to be trained, in fact, just a few example events at the beginning. Peng et al. (2016) addressed the manual annotation by proposing an event detection and coreference system that requires minimal supervision, particularly, a few training examples. Their approach was built on a key assumption: the semantics of two tasks (i) identifying events closely related to some event types and (ii) event coreference are similar. As such, reformulating the task into semantic similarity can help the model to be trained on a large available corpus of event coreference instead of annotating a large dataset for event detection. As a result, the required data for any new event type is as small as the number of samples in the annotation guidelines. To do that, they use a general purpose nominal and verbial semantic role labeling (SRL) representation to represent the structure of an event. The representation involves multiple semantic spaces, including contextual, topical, and syntactic level. Figure 8 demonstrates their method in detail. Similarly, Lai and Nguyen (2019) proposed a novel formulation for event de-

tection, namely learning from keywords (LFK) in which each type is described via a few event triggers. They are pre-selected from a pool of known events. In order to encode the sentence, the model contains a CNN-based encoder and a conditional feature-wise attention mechanism to selectively enhance informative features.

Lai et al. (2020a), Deng et al. (2020) and Lai et al. (2021a) employed the core architecture of the prototypical network while proposed an auxiliary training loss factors during the training process. Lai et al. (2020a) enforce the distances between clusters of samples, namely intra-cluster loss and inter-cluster loss. The intra-cluster loss minimizes the distances between samples of the same class. In contrast, the inter-cluster loss maximizes the distances between the prototype of a class and the examples of the other classes. The model also introduces contextualized embedding, which leads to significant performance improvement over ANN or CNN-based encoders. (Deng et al., 2020), on the other hand, proposed a Dynamic-Memory-Based Prototypical Network (DMB-PN). The model uses a Dynamic Memory Network(DMN) to learn better prototypes and produce better event mention encodings. The prototypes are not computed by averaging the supporting events just once, but they are induced from the supporting events multiple times through DMN's multihop mechanism. Lai et al. (2021a) addressed the outlier and sampling bias in the training process of few-shot event detection. Particularly, in event detection, a null class is introduced to represent samples that are out of the interested classes. These may contain non-interested eventive samples as well as non-eventive samples. As such, this class may inject outlier examples into the support set. As such, they proposed a novel model for the relation between two training tasks in an episodic training setting by allowing interactions between prototypes of two tasks. They also proposed prediction consistency between two tasks so that the trained model would be more resistant to outliers.

Chen et al. (2021) addressed the trigger curse problem in FSL-ED. Particularly, both overfitting and underfitting trigger identification are harmful to the generalization ability or the detection performance of the model, respectively. They argue that the trigger is the confounder of the context and the result of an event. As such, previous models, which are trigger-centric, can easily overfit triggers.

To alleviate the trigger overfitting, they proposed a method to intervene in the context by backdoor adjustment during training.

Recent work by Shen et al. (2021) tackles the low sample diversity in FSL-ED. Their model, Adaptive Knowledge-Enhanced Bayesian Meta-Learning (AKE-BML), introduces external event knowledge as a prior of the event type. First, they heuristically align the event types in the support set and FrameNet to do that. Then they encode the samples and the aligned examples in the same semantic space using a neural-based encoder. After that, they realign the knowledge representation by using a learnable offset, resulting in a prior knowledge distribution for event types. Then they can generate a posterior distribution for event types. Finally, to predict the label for a query instance, they use the posterior distribution for prototype representations to classify query instances into event types.

The second FSL-ED setting is based on sequence labeling. The few-shot sequence labeling setting, in general, has been widely studied in named entities recognition (Fritzler et al., 2019). Similarly, Cong et al. (2021) formulated the FSL-ED as a few-shot sequence labeling problem, which detects the spans of the events and the label of the event at the same time. They argue that previous studies that solve this problem in the **identify-then-classify** manner suffer from error propagation due to ignoring the discrepancy of triggers between event types. They proposed a CRF-based model called Prototypical Amortized Conditional Random Field (PA-CRF). In order to model the CRF-based classifiers, it is important to approximate the transition and emission scores from just a few examples. Their model approximates the transition scores between labels based on the label prototypes. In the meantime, they introduced a Gaussian distribution into the transition scores to alleviate the uncertain estimation of the emission scorer. Figure 9 presents the overview architecture of the model.

4.3 Cross-lingual

Early studies of cross-lingual event extraction (CLEE) relies on training a statistical model on parallel data for event extraction (Chen and Ji, 2009; Piskorski et al., 2011; Hsi et al., 2016). Recent methods focus on transferring universal structures across languages (Subburathinam et al., 2019; Liu et al., 2019a; Lu et al., 2020; Nguyen and Nguyen,
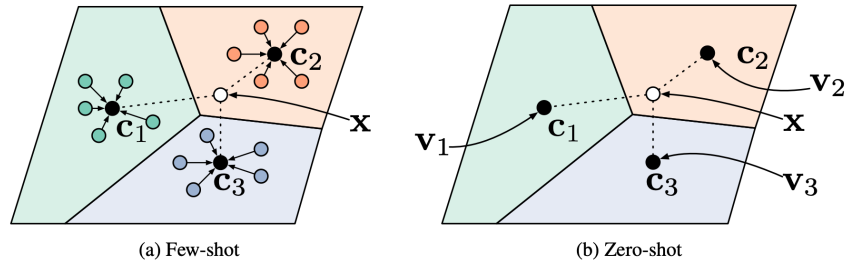
(a) Few-shot   (b) Zero-shot

Figure 7: Prototypical network (Snell et al., 2017). In a prototypical network, an encoder encodes each sample into a vector. Given a few examples per class (figure a, color dots), a prototype (black dots or c_1, c_2, and c_3) of a class is computed as the average of all encoded vectors corresponding to that class. Once there is a query x, a score distribution is computed based on either the similarity scores or the distances between the given query x and the prototypes. The predicted class is selected to maximize the estimated similarity scores or minimize the estimated distances.
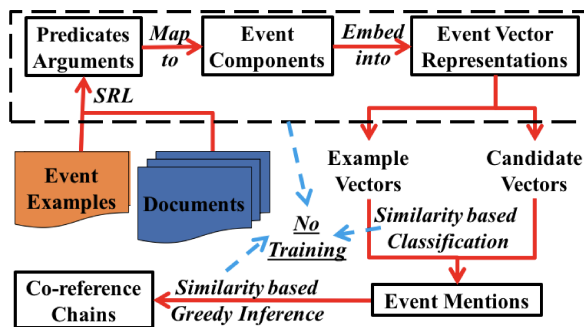

Figure 8: Event Detection framework by Peng et al. (2016)

2021). There are a few other methods were also studied such as topic modeling (Li et al., 2011), multilingual embedding (M'hamdi et al., 2019), and annotation projection (Li et al., 2016; Lou et al., 2022).

Cross-lingual event extraction depends on a parallel corpus for both training and evaluation. However, parallel corpora for this area are scarce. Most of the work in CLEE were done using ACE-2005 (LDC, 2005), TAC-KBP (Mitamura et al., 2015, 2017), and TempEval-2 (Verhagen et al., 2010). These multilingual datasets cover several popular languages, such as English, Chinese, Arabic, and Spanish. Recently, datasets that cover less common languages, e.g., Polish, Danish, Turkish, Hindi, Urdu, Korean and Japanese, were created for event detection (Pouran Ben Veyseh et al., 2022) and event relation extraction (Lai et al., 2022b).

Due to data scarcity in target languages, the model trained on limited data might not be able to predict a wide range of events. Therefore, generating more data from the existing corpus in the source language is a trivial method. Li et al. (2016) proposed a projection algorithm to mine shared hidden

phrases and structures between two languages (i.e., English and Chinese). They project seed phrases back and forth multiple rounds between the two languages using parallel corpora to obtain a diverse set of closely related phrases. The captured phrases are then used to train an ED model. This method was shown to effectively improve the diversity of the recognized events. Lou et al. (2022) addressed the problem of noise appearing in the translated corpus. They proposed an annotation projection approach that combines the translation projection and the event argument extraction task training step to alleviate the additional noise through implicit annotation projection. First, they translate the source language corpus into the target language using a multilingual machine translation model. To reduce the noise of the translated data, instead of training the model directly from them, they use multilingual embedding to embed the source language data and the translated derivatives in the target language into the same vector space. Their representations are then aligned using optimal transport. They proposed two additional training signals that either reduce the alignment scores or the prediction based on the aligned representation. Phung et al. (2021) explored the cross-lingual transfer learning for event coreference resolution task. They introduced the language adversarial neural network to help the model distinguish texts from the source and target languages. This helps the model improve the generalization over languages for the task. Similar to (Lou et al., 2022), the work by Phung et al. (2021) introduced an alignment method based on multiple views of the text from the source and the target languages. They further introduced optimal transport to better select edge examples in the
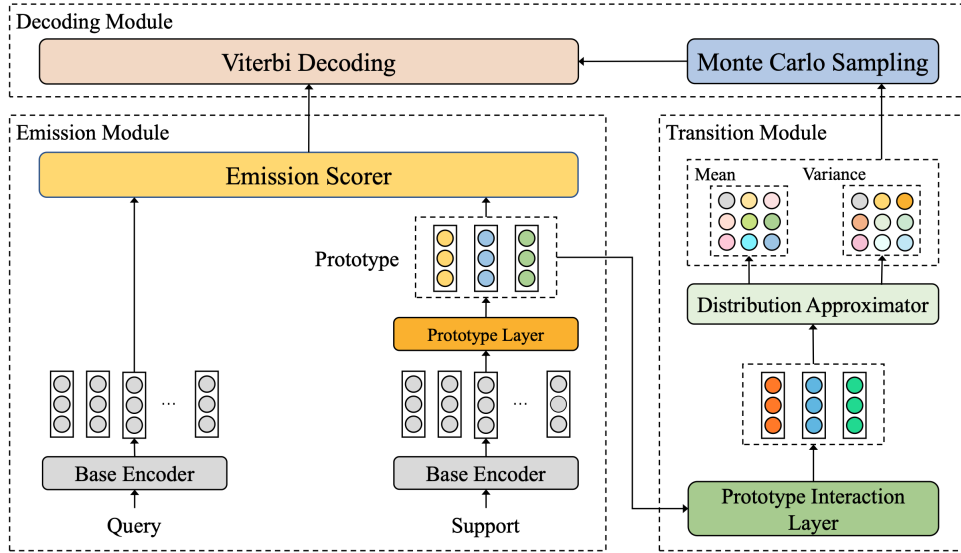
Figure 9: Architecture of the PA-CRF model (Cong et al., 2021). The model consists of 3 modules. The Emission Module computes the emission scores for the query instances based on the prototypes of the support set. The Transition Module estimates the Gaussian distributed transition scores from the prototypes. The Decoding Module consumes the emissions scores and the approximated transition scores to decode the predicted sequences with Monte Carlo sampling.

source and target languages to train the language discriminator.

Multilingual embedding plays an important role in transferring knowledge between languages. There have been many multilingual contextualized embedding built for a large number of languages such as FastText (Joulin et al., 2018), MUSE (Lample et al., 2017), mBERT (Devlin et al., 2019), mBART (Liu et al., 2020b), XLM-RoBERTa (Conneau et al., 2020), and mT5/mT6 (Xue et al., 2021; Chi et al., 2021). (M'hamdi et al., 2019) compared FastText, MUSE and mBERT. The results show that multilingual embeddings help transfer knowledge from English data to other languages, i.e., Chinese and Arabic. The performance boost is significant when all multilingual are added to train the model. Various multilingual embeddings have been employed in cross-lingual event extraction thanks to their robustness and transferability. However, models trained on multilingual embedding still suffer from performance drop in zero-shot cross-lingual settings. It is even worse than monolingual embedding if the monolingual model is trained on a large enough target dataset and a good enough monolingual contextualized embedding (Lai et al., 2022b).

Most of the recent methods for cross-lingual event extraction are done via transferring shared features between languages, such as syntactic struc-

tures (e.g., part-of-speech, dependency tree), semantic features (e.g., contextualized embedding), and relation structures (e.g., entity relation). Subburathinam et al. (2019) addressed the suitability of transferring cross-lingual structures for the event and relation extraction tasks. They exploit relevant language-universal features for relation and events such as symbolic features (i.e., part-of-speech and dependency path) and distributional features (i.e., type representation and contextualized representation) to transfer those structures appearing in the source language corpus to the target language. Figure 10 presents an example of shared structures of two different sentences in two languages (i.e., English and Russian). Thanks to this similarity, they encode all the entity mentions, event triggers, and event context from both languages into a complex shared cross-lingual vector space using a graph convolutional neural network. Hence, once the model is trained in English, this shared structural knowledge will be transferred to the target languages, such as Russian. (Liu et al., 2019a) addressed two issues in cross-lingual transfer learning: (i) how to build a lexical mapping between languages and (ii) how to manage the effect of the word-order differences between different languages. First, they employ a context-dependent translation method to construct the lexical mapping between languages by first retrieving k nearest neighbors in a shared
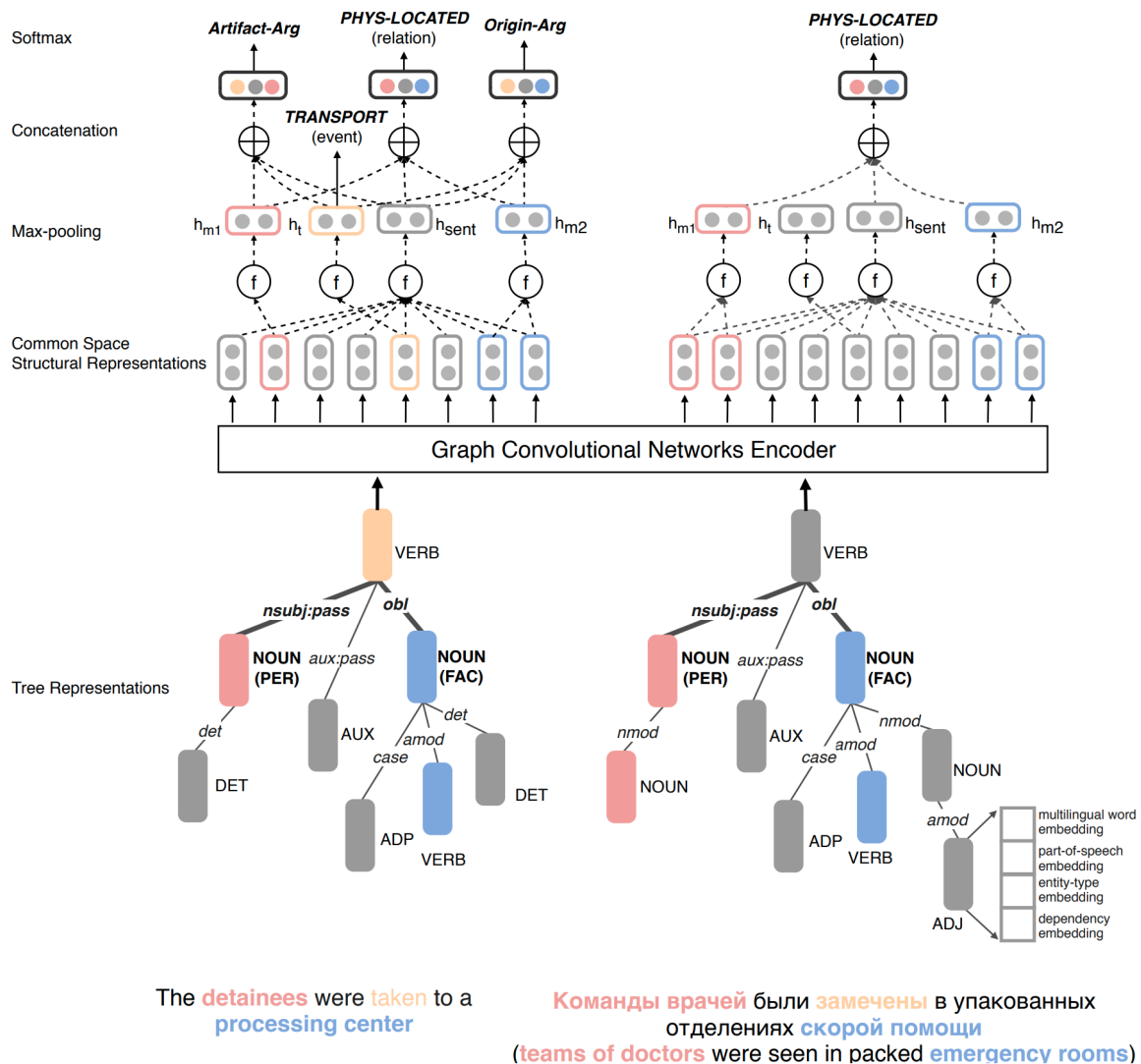
Figure 10: Shared structures of different event mentions, written in English and Russian, of the same event type (Subburathinam et al., 2019).

vector space, then reranking the candidates using a context-aware selective attention mechanism. To encode sentences with language-dependent word order, a GCN model is employed to encode the sentence. To enrich the features for the cross-lingual event argument extraction model, Nguyen and Nguyen (2021) employ three types of connection to build a feature-expanded graph. The core of the graph is derived from the dependency graph used in many other studies to capture syntactic features. They introduced two additional connections to capture semantic similarity and the universal dependency relations of the word pairs. Based on the assumption that most concepts are universal across languages, similarities between words and representing concepts are also universal. They employ a multilingual contextualized embedding to obtain

the word representation, then compute a similarity score between words in a sentence. Secondly, they argue that the relation types play an important role in the connection's strength. Therefore, another connection set of weights is computed based on the dependency relation type between two connected words. Finally, the additional edge weights are added to the graph, scaling to the extent of the similarity score of the relation.

Model Acronym	System	Trigger		Argument	
		Identification	Classification	Identification	Classification
Feature engineering					
	Ahn (2006)	62.6	60.1	82.4	57.3
Cross-document	Ji and Grishman (2008)		67.3	46.2	42.6
Cross-event	Liao and Grishman (2010)	-	68.8	50.3	44.6
Cross-entity	Hong et al. (2011)	-	68.3	53.1	48.3
Structure-prediction	Li et al. (2013)	70.4	67.5	56.8	52.7
CNN					
CNN	Nguyen and Grishman (2015)		69.0	-	-
DMCNN	Chen et al. (2015)	73.5	69.1	59.1	53.5
DMCNN+DS	Chen et al. (2017)	74.3	70.5	63.3	55.7
RNN					
JRNN	Nguyen et al. (2016a)	71.9	69.3	62.8	55.4
FBRNN	Ghaeini et al. (2016)	-	67.4	-	-
BDLSTM-TNNs	Chen et al. (2016)	72.2	68.9	60.0	54.1
DLRNN	Duan et al. (2017)	-	70.5	-	-
dbRNN	Sha et al. (2018)	-	71.9	67.7	58.7
GCN					
GCN-ED	Nguyen and Grishman (2018)	-	73.1	-	-
JMEE	Liu et al. (2018)	75.9	73.7	68.4	60.3
MOGANED	Yan et al. (2019)	-	75.7	-	-
MOGANED+GTN	Dutta et al. (2021)	-	76.8	-	-
GatedGCN	Lai et al. (2020b)	-	77.6	-	-
Data Generation & Augmentation					
ANN-FN	Liu et al. (2016)	-	70.7	-	-
Liberal	Huang et al. (2016)	-	61.8	-	44.8
Chen's Generation	Chen et al. (2017)	74.3	70.5	63.3	55.7
BLSTM-CRF-ILP$_{multi}$	Zeng et al. (2018)	-	82.5	-	37.9
EKD	Tong et al. (2020)	-	78.6	-	-
GPTEDOT	Pouran Ben Veyseh et al. (2021a)	-	79.2	-	-
Document-level Modeling					
HBTNGMA	Chen et al. (2018)	-	73.3	-	-
DEEB-RNN	Zhao et al. (2018)	-	74.9	-	-
ED3C	Pouran Ben Veyseh et al. (2021b)	-	79.1	-	-
Joint Modeling					
DyGIE++	Wadden et al. (2019)	76.5	73.6	55.4	52.5
HPNet	Huang et al. (2020b)	79.2	77.8	60.9	56.8
OneIE	Lin et al. (2020)	-	72.8	-	56.3
NGS	Wang et al. (2020a)	-	74.6		59.5
Text2event	Lu et al. (2021)	-	71.8	-	54.4
AMRIE	Zhang and Ji (2021)	-	72.8	-	57.7
FourIE	Nguyen et al. (2021)	-	73.3	-	58.3
DEGREE	Hsu et al. (2022)	-	71.7	-	58.0
GraphIE	Nguyen et al. (2022)	-	74.8	-	60.2

Table 5: Performance of the presented models on ACE-05 dataset.

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.

Jun Araki and Teruko Mitamura. 2015. Joint event trigger identification and event coreference resolution with structured perceptron. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2074–2080, Lisbon, Portugal. Association for Computational Linguistics.

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 86–90.

David Bamman, Olivia Lewke, and Anya Mansoor. 2020. An annotated dataset of coreference in English literature. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 44–54, Marseille, France. European Language Resources Association.

David Bamman, Sejal Popat, and Sheng Shen. 2019. An annotated dataset of literary entities. 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 2138–2144, Minneapolis, Minnesota. Association for Computational Linguistics.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. 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 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250.

Ofer Bronstein, Ido Dagan, Qi Li, Heng Ji, and Anette Frank. 2015. Seed-based event trigger labeling: How far can event descriptions get us? 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 372–376, Beijing, China. Association for Computational Linguistics.

Tommaso Caselli and Piek Vossen. 2017. The event StoryLine corpus: A new benchmark for causal and temporal relation extraction. In *Proceedings of the Events and Stories in the News Workshop*, pages 77–86, Vancouver, Canada. Association for Computational Linguistics.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2021. Honey or poison? solving the trigger curse in few-shot event detection via causal intervention. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8078–8088, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yubo Chen, Shulin Liu, Shizhu He, Kang Liu, and Jun Zhao. 2016. Event extraction via bidirectional long short-term memory tensor neural networks. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, pages 190–203. Springer.

Yubo Chen, Shulin Liu, Xiang Zhang, Kang Liu, and Jun Zhao. 2017. Automatically labeled data generation for large scale event extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 409–419, Vancouver, Canada. Association for Computational Linguistics.

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.

Yubo Chen, Hang Yang, Kang Liu, Jun Zhao, and Yantao Jia. 2018. Collective event detection via a hierarchical and bias tagging networks with gated multi-level attention mechanisms. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1267–1276, Brussels, Belgium. Association for Computational Linguistics.

Zheng Chen and Heng Ji. 2009. Can one language bootstrap the other: A case study on event extraction. In *Proceedings of the NAACL HLT 2009 Workshop on Semi-supervised Learning for Natural Language Processing*, pages 66–74, Boulder, Colorado. Association for Computational Linguistics.

Zewen Chi, Li Dong, Shuming Ma, Shaohan Huang, Saksham Singhal, Xian-Ling Mao, Heyan Huang, Xia Song, and Furu Wei. 2021. mT6: Multilingual pretrained text-to-text transformer with translation pairs. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1671–1683, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar. Association for Computational Linguistics.

Xin Cong, Shiyao Cui, Bowen Yu, Tingwen Liu, Wang Yubin, and Bin Wang. 2021. Few-Shot Event Detection with Prototypical Amortized Conditional Random Field. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 28–40, Online. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. 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 *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2329–2339, Online. Association for Computational Linguistics.

Shumin Deng, Ningyu Zhang, Jiaojian Kang, Yichi Zhang, Wei Zhang, and Huajun Chen. 2020. Meta-learning with dynamic-memory-based prototypical network for few-shot event detection. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 151–159.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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.

Xiao Ding, Fan Song, Bing Qin, and Ting LIU. 2011. Research on typical event extraction method in the field of music. *Journal of Chinese Information Processing*, 2.

Xinya Du and Claire Cardie. 2020. Document-level event role filler extraction using multi-granularity contextualized encoding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8010–8020, Online. Association for Computational Linguistics.

Shaoyang Duan, Ruifang He, and Wenli Zhao. 2017. Exploiting document level information to improve event detection via recurrent neural networks. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 352–361, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Sanghamitra Dutta, Liang Ma, Tanay Kumar Saha, Di Liu, Joel Tetreault, and Alejandro Jaimes. 2021. GTN-ED: Event detection using graph transformer networks. In *Proceedings of the Fifteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-15)*, pages 132–137, Mexico City, Mexico. Association for Computational Linguistics.

Joe Ellis, Jeremy Getman, Dana Fore, Neil Kuster, Zhiyi Song, Ann Bies, and Stephanie M Strassel. 2015. Overview of linguistic resources for the tac kbp 2015 evaluations: Methodologies and results. In *TAC*.

James Ferguson, Colin Lockard, Daniel 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, New Orleans, Louisiana. Association for Computational Linguistics.

Alexander Fritzler, Varvara Logacheva, and Maksim Kretov. 2019. Few-shot classification in named entity recognition task. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pages 993–1000.

Tao Ge, Lei Cui, Baobao Chang, Zhifang Sui, Furu Wei, and Ming Zhou. 2018. EventWiki: A knowledge base of major events. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

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, Berlin, Germany. Association for Computational Linguistics.

Goran Glavaš, Jan Šnajder, Marie-Francine Moens, and Parisa Kordjamshidi. 2014. HiEve: A corpus for extracting event hierarchies from news stories. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3678–3683, Reykjavik, Iceland. European Language Resources Association (ELRA).

Ralph Grishman and Beth M Sundheim. 1996. Message understanding conference-6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.

Luis Guzman-Nateras, Viet Lai, Amir Pouran Ben Veyseh, Franck Dernoncourt, and Thien Nguyen. 2022. Event detection for suicide understanding. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1952–1961, Seattle, United States. Association for Computational Linguistics.

Rujun Han, Qiang Ning, and Nanyun Peng. 2019. Joint event and temporal relation extraction with shared representations and structured prediction. 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 434–444, Hong Kong, China. Association for Computational Linguistics.

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

Yu Hong, Jianfeng Zhang, Bin Ma, Jianmin Yao, Guodong Zhou, and Qiaoming Zhu. 2011. Using cross-entity inference to improve event extraction. In *ACL*.

Andrew Hsi, Yiming Yang, Jaime Carbonell, and Ruochen Xu. 2016. Leveraging multilingual training for limited resource event extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1201–1210, Osaka, Japan. The COLING 2016 Organizing Committee.

I-Hung Hsu, Kuan-Hao Huang, Elizabeth Boschee, Scott Miller, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. 2022. DEGREE: A data-efficient generation-based event extraction model. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1890–1908, Seattle, United States. Association for Computational Linguistics.

Kung-Hsiang Huang and Nanyun Peng. 2021. Document-level event extraction with efficient end-to-end learning of cross-event dependencies. In *Proceedings of the Third Workshop on Narrative Understanding*, pages 36–47, Virtual. Association for Computational Linguistics.

Kung-Hsiang Huang, Mu Yang, and Nanyun Peng. 2020a. Biomedical event extraction with hierarchical knowledge graphs. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1277–1285, Online. Association for Computational Linguistics.

Lifu Huang, Taylor Cassidy, Xiaocheng Feng, Heng Ji, Clare R. Voss, Jiawei Han, and Avirup Sil. 2016. Liberal event extraction and event schema induction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 258–268, Berlin, Germany. Association for Computational Linguistics.

Lifu Huang, Heng Ji, Kyunghyun Cho, Ido Dagan, Sebastian Riedel, and Clare Voss. 2018a. Zero-shot transfer learning for event extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2160–2170, Melbourne, Australia. Association for Computational Linguistics.

Lifu Huang, Heng Ji, Kyunghyun Cho, and Clare R Voss. 2018b. Zero-shot transfer learning for event extraction. In *ACL*, pages 2160–2170.

Peixin Huang, Xiang Zhao, Ryuichi Takanobu, Zhen Tan, and Weidong Xiao. 2020b. Joint event extraction with hierarchical policy network. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2653–2664, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yusheng Huang and Weijia Jia. 2021. Exploring sentence community for document-level event extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 340–351, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. *ICLR*.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *ACL*.

Heng Ji, Joel Nothman, H Trang Dang, and Sydney Informatics Hub. 2016. Overview of tac-kbp2016 trilingual edl and its impact on end-to-end cold-start kbp. *Proceedings of TAC*.

Armand Joulin, Piotr Bojanowski, Tomas Mikolov, Hervé Jégou, and Edouard Grave. 2018. Loss in translation: Learning bilingual word mapping with a retrieval criterion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2979–2984, Brussels, Belgium. Association for Computational Linguistics.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665, Baltimore, Maryland. Association for Computational Linguistics.

Jin-Dong Kim, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano, and Jun'ichi Tsujii. 2009. Overview of BioNLP'09 shared task on event extraction. In *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*, pages 1–9, Boulder, Colorado. Association for Computational Linguistics.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

Dorian Kodelja, Romaric Besançon, and Olivier Ferret. 2019. Exploiting a more global context for event detection through bootstrapping. In *European Conference on Information Retrieval*, pages 763–770. Springer.

Huiyuan Lai, Antonio Toral, and Malvina Nissim. 2022a. Multilingual pre-training with language and task adaptation for multilingual text style transfer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 262–271, Dublin, Ireland. Association for Computational Linguistics.

Viet Lai, Franck Dernoncourt, and Thien Huu Nguyen. 2021a. Learning prototype representations across few-shot tasks for event detection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5270–5277, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Viet Lai, Amir Pouran Ben Veyseh, Minh Nguyen, Franck Dernoncourt, and Thien Nguyen. 2022b. Meci: A multilingual dataset for event causality identification. In *COLING*. Association for Computational Linguistics.

Viet Dac Lai, Minh Van Nguyen, Heidi Kaufman, and Thien Huu Nguyen. 2021b. Event extraction from historical texts: A new dataset for black rebellions. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2390–2400, Online. Association for Computational Linguistics.

Viet Dac Lai and Thien Nguyen. 2019. Extending event detection to new types with learning from keywords. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 243–248, Hong Kong, China. Association for Computational Linguistics.

Viet Dac Lai, Thien Huu Nguyen, and Franck Dernoncourt. 2020a. Extensively matching for few-shot learning event detection. In *Proceedings of the First Joint Workshop on Narrative Understanding, Storylines, and Events*, pages 38–45, Online. Association for Computational Linguistics.

Viet Dac Lai, Tuan Ngo Nguyen, and Thien Huu Nguyen. 2020b. 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, Online. Association for Computational Linguistics.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. *ICLR*.

LDC. 2005. *ACE (Automatic Content Extraction) English Annotation Guidelines for Events*. Linguistic Data Consortium.

Duong Le and Thien Huu Nguyen. 2021. Fine-grained event trigger detection. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2745–2752, Online. Association for Computational Linguistics.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Heeyoung Lee, Marta Recasens, Angel Chang, Mihai Surdeanu, and Dan Jurafsky. 2012. Joint entity and event coreference resolution across documents. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 489–500, Jeju Island, Korea. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Diya Li, Lifu Huang, Heng Ji, and Jiawei Han. 2019. Biomedical event extraction based on knowledge-driven tree-LSTM. 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 1421–1430, Minneapolis, Minnesota. Association for Computational Linguistics.

Fangyuan Li, Ruihong Huang, Deyi Xiong, and Min Zhang. 2016. Learning event expressions via bilingual structure projection. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1441–1450, Osaka, Japan. The COLING 2016 Organizing Committee.

Hao Li, Heng Ji, Hongbo Deng, and Jiawei Han. 2011. Exploiting background information networks to enhance bilingual event extraction through topic modeling. In *Proc. of International Conference on Advances in Information Mining and Management*.

Lishuang Li, Yang Liu, and Meiyue Qin. 2018. Extracting biomedical events with parallel multi-pooling convolutional neural networks. *IEEE/ACM transactions on computational biology and bioinformatics*, 17(2):599–607.

Manling Li, Ruochen Xu, Shuohang Wang, Luowei Zhou, Xudong Lin, Chenguang Zhu, Michael Zeng, Heng Ji, and Shih-Fu Chang. 2022. Clip-event: Connecting text and images with event structures. In

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 16420–16429.

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, Sofia, Bulgaria. Association for Computational Linguistics.

Shasha Liao and Ralph Grishman. 2010. Using document level cross-event inference to improve event extraction. In *ACL*.

Shasha Liao and Ralph Grishman. 2011. Acquiring topic features to improve event extraction: in preselected and balanced collections. In *RANLP*.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009, Online. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

Jian Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2019a. Neural cross-lingual event detection with minimal parallel resources. 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 738–748, Hong Kong, China. Association for Computational Linguistics.

Shulin Liu, Yubo Chen, Shizhu He, Kang Liu, and Jun Zhao. 2016. Leveraging FrameNet to improve automatic event detection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2134–2143, Berlin, Germany. Association for Computational Linguistics.

Xiao Liu, Zhunchen Luo, and Heyan Huang. 2018. Jointly multiple events extraction via attention-based graph information aggregation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1247–1256, Brussels, Belgium. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020b. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

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

Chenwei Lou, Jun Gao, Changlong Yu, Wei Wang, Huan Zhao, Weiwei Tu, and Ruifeng Xu. 2022. Translation-based implicit annotation projection for zero-shot cross-lingual event argument extraction. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2076–2081.

Di Lu, Ananya Subburathinam, Heng Ji, Jonathan May, Shih-Fu Chang, Avi Sil, and Clare Voss. 2020. Cross-lingual structure transfer for zero-resource event extraction. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1976–1981, Marseille, France. European Language Resources Association.

Wei Lu and Dan Roth. 2012. Automatic event extraction with structured preference modeling. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 835–844, Jeju Island, Korea. Association for Computational Linguistics.

Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. 2021. Text2Event: Controllable sequence-to-structure generation for end-to-end event extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2795–2806, Online. Association for Computational Linguistics.

Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019. A general framework for information extraction using dynamic span graphs. 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 3036–3046, Minneapolis, Minnesota. Association for Computational Linguistics.

Qing Lyu, Hongming Zhang, Elior Sulem, and Dan Roth. 2021. Zero-shot event extraction via transfer learning: Challenges and insights. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 322–332, Online. Association for Computational Linguistics.

Amit Majumder and Asif Ekbal. 2015. Event extraction from biomedical text using crf and genetic algorithm. In *Proceedings of the 2015 Third International Conference on Computer, Communication, Control and Information Technology (C3IT)*, pages 1–7. IEEE.

Hieu Man Duc Trong, Duc Trong Le, Amir Pouran Ben Veyseh, Thuat Nguyen, and Thien Huu Nguyen. 2020. Introducing a new dataset for event detection in cybersecurity texts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5381–5390, Online. Association for Computational Linguistics.

Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1506–1515, Copenhagen, Denmark. Association for Computational Linguistics.

David McClosky, Mihai Surdeanu, and Christopher Manning. 2011. Event extraction as dependency parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1626–1635, Portland, Oregon, USA. Association for Computational Linguistics.

Meryem M'hamdi, Marjorie Freedman, and Jonathan May. 2019. Contextualized cross-lingual event trigger extraction with minimal resources. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 656–665, Hong Kong, China. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Bonan Min and Xiaoxi Zhao. 2019. Measure country-level socio-economic indicators with streaming news: An empirical study. In *EMNLP-IJCNLP*, pages 1249–1254, Hong Kong, China. Association for Computational Linguistics.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, Suntec, Singapore. Association for Computational Linguistics.

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

Teruko Mitamura, Zhengzhong Liu, and Eduard H Hovy. 2017. Events detection, coreference and sequencing: What's next? overview of the tac kbp 2017 event track. In *TAC*.

Minh Van Nguyen, Viet Dac Lai, and Thien Huu Nguyen. 2021. Cross-task instance representation interactions and label dependencies for joint information extraction with graph convolutional networks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 27–38, Online. Association for Computational Linguistics.

Minh Van Nguyen, Bonan Min, Franck Dernoncourt, and Thien Nguyen. 2022. Joint extraction of entities, relations, and events via modeling inter-instance and inter-label dependencies. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4363–4374, Seattle, United States. Association for Computational Linguistics.

Minh Van Nguyen and Thien Huu Nguyen. 2021. Improving cross-lingual transfer for event argument extraction with language-universal sentence structures. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 237–243, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016a. Joint event extraction via recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 300–309, San Diego, California. Association for Computational Linguistics.

Thien Huu Nguyen, Nicolas Fauceglia, Mariano Rodriguez Muro, Oktie Hassanzadeh, Alfio Massimiliano Gliozzo, and Mohammad Sadoghi. 2016b. Joint learning of local and global features for entity linking via neural networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2310–2320, Osaka, Japan. The COLING 2016 Organizing Committee.

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, Beijing, China. Association for Computational Linguistics.

Thien Huu Nguyen and Ralph Grishman. 2018. Graph convolutional networks with argument-aware pooling for event detection. In *Thirty-second AAAI conference on artificial intelligence*.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman.

2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, Portorož, Slovenia. European Language Resources Association (ELRA).

Tomoko Ohta, Yuka Tateisi, Jin-Dong Kim, Hideki Mima, and Junichi Tsujii. 2002. The genia corpus: An annotated research abstract corpus in molecular biology domain. In *Proceedings of the human language technology conference*, pages 73–77. Citeseer.

Siddharth Patwardhan and Ellen Riloff. 2009. A unified model of phrasal and sentential evidence for information extraction. In *EMNLP*.

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, Austin, Texas. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher 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, Doha, Qatar. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Duy Phung, Hieu Minh Tran, Minh Van Nguyen, and Thien Huu Nguyen. 2021. Learning cross-lingual representations for event coreference resolution with multi-view alignment and optimal transport. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 62–73, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jakub Piskorski, Jenya Belayeva, and Martin Atkinson. 2011. Exploring the usefulness of cross-lingual information fusion for refining real-time news event extraction: A preliminary study. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, pages 210–217, Hissar, Bulgaria. Association for Computational Linguistics.

Amir Pouran Ben Veyseh, Viet Lai, Franck Dernoncourt, and Thien Huu Nguyen. 2021a. Unleash GPT-2 power for event detection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6271–6282, Online. Association for Computational Linguistics.

Amir Pouran Ben Veyseh, Minh Van Nguyen, Franck Dernoncourt, and Thien Nguyen. 2022. MINION: a large-scale and diverse dataset for multilingual event detection. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2286–2299, Seattle, United States. Association for Computational Linguistics.

Amir Pouran Ben Veyseh, Minh Van Nguyen, Nghia Ngo Trung, Bonan Min, and Thien Huu Nguyen. 2021b. Modeling document-level context for event detection via important context selection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5403–5413, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

James Pustejovsky, Patrick Hanks, Roser Sauri, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa Ferro, et al. 2003. The timebank corpus. In *Corpus linguistics*, volume 2003, page 40. Lancaster, UK.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Sebastian Ruder, Anders Søgaard, and Ivan Vulić. 2019. Unsupervised cross-lingual representation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 31–38, Florence, Italy. Association for Computational Linguistics.

Taneeya Satyapanich, Francis Ferraro, and Tim Finin. 2020. Casie: Extracting cybersecurity event information from text. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8749–8757.

Lei Sha, Feng Qian, Baobao Chang, and Zhifang Sui. 2018. Jointly extracting event triggers and arguments by dependency-bridge rnn and tensor-based argument interaction. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Shirong Shen, Tongtong Wu, Guilin Qi, Yuan-Fang Li, Gholamreza Haffari, and Sheng Bi. 2021. Adaptive knowledge-enhanced Bayesian meta-learning for few-shot event detection. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2417–2429, Online. Association for Computational Linguistics.

Matthew Sims, Jong Ho Park, and David Bamman. 2019. Literary event detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3623–3634, Florence, Italy. Association for Computational Linguistics.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30.

Rachele Sprugnoli and Sara Tonelli. 2019. Novel event detection and classification for historical texts. *Computational Linguistics*, 45(2):229–265.

Ananya Subburathinam, Di Lu, Heng Ji, Jonathan May, Shih-Fu Chang, Avirup Sil, and Clare Voss. 2019. Cross-lingual structure transfer for relation and event extraction. 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 313–325, Hong Kong, China. Association for Computational Linguistics.

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

Hieu Man Duc Trong, Duc Trong Le, Amir Pouran Ben Veyseh, Thuat Nguyen, and Thien Huu Nguyen. 2020. Introducing a new dataset for event detection in cybersecurity texts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5381–5390.

Naushad UzZaman, Hector Llorens, Leon Derczynski, James Allen, Marc Verhagen, and James Pustejovsky. 2013. SemEval-2013 task 1: TempEval-3: Evaluating time expressions, events, and temporal relations. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 1–9, Atlanta, Georgia, USA. Association for Computational Linguistics.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. *ICLR*.

Deepak Venugopal, Chen Chen, Vibhav Gogate, and Vincent Ng. 2014. Relieving the computational bottleneck: Joint inference for event extraction with high-dimensional features. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 831–843, Doha, Qatar. Association for Computational Linguistics.

Marc Verhagen, Roser Saurí, Tommaso Caselli, and James Pustejovsky. 2010. SemEval-2010 task 13: TempEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 57–62, Uppsala, Sweden. Association for Computational Linguistics.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. 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 5784–5789, Hong Kong, China. Association for Computational Linguistics.

Xiaozhi Wang, Shengyu Jia, Xu Han, Zhiyuan Liu, Juanzi Li, Peng Li, and Jie Zhou. 2020a. Neural Gibbs Sampling for Joint Event Argument Extraction. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 169–180, Suzhou, China. Association for Computational Linguistics.

Xiaozhi Wang, Ziqi Wang, Xu Han, Wangyi Jiang, Rong Han, Zhiyuan Liu, Juanzi Li, Peng Li, Yankai Lin, and Jie Zhou. 2020b. MAVEN: A Massive General Domain Event Detection Dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1652–1671, Online. Association for Computational Linguistics.

Runxin Xu, Tianyu Liu, Lei Li, and Baobao Chang. 2021. Document-level event extraction via heterogeneous graph-based interaction model with a tracker. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3533–3546, Online. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

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 5766–5770, Hong Kong, China. Association for Computational Linguistics.

Hang Yang, Yubo Chen, Kang Liu, Yang Xiao, and Jun Zhao. 2018. DCFEE: A document-level Chinese financial event extraction system based on automatically labeled training data. In *Proceedings of ACL 2018, System Demonstrations*, pages 50–55, Melbourne, Australia. Association for Computational Linguistics.

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, Florence, Italy. Association for Computational Linguistics.

Ying Zeng, Yansong Feng, Rong Ma, Zheng Wang, Rui Yan, Chongde Shi, and Dongyan Zhao. 2018. Scale up event extraction learning via automatic training data generation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Hongming Zhang, Haoyu Wang, and Dan Roth. 2021. Zero-shot Label-aware Event Trigger and Argument Classification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1331–1340, Online. Association for Computational Linguistics.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2205–2215, Brussels, Belgium. Association for Computational Linguistics.

Zhengkuan Zhang, Weiran Xu, and Qianqian Chen. 2016. Joint event extraction based on skip-window convolutional neural networks. In *Natural Language Understanding and Intelligent Applications*, pages 324–334. Springer.

Zixuan Zhang and Heng Ji. 2021. Abstract Meaning Representation guided graph encoding and decoding for joint information extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 39–49, Online. Association for Computational Linguistics.

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, Melbourne, Australia. Association for Computational Linguistics.

Shun Zheng, Wei Cao, Wei Xu, and Jiang Bian. 2019. Doc2EDAG: An end-to-end document-level framework for Chinese financial event extraction. 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 337–346, Hong Kong, China. Association for Computational Linguistics.