Utilizing Text Structure for Information Extraction

Amir Pouran Ben Veyseh
Department of Computer and Information Science, University of Oregon,
Eugene, OR 97403, USA
apouranb@cs.uoregon.edu

Abstract

Information Extraction (IE) is one of the important fields of natural language processing (NLP) with the primary goal of creating structured knowledge from unstructured text. In more than two decades, IE has gained a lot of attention and many new tasks and models have been proposed. Moreover, with the proliferation of deep learning and neural nets in recent years, the advanced deep models have brought about a surge in the performance of IE models. Among others, some of the existing deep models resort to structure-based modeling whose goal is to exploit the structure of the text (i.e., interactions of different parts of the text) or external structures (e.g., a knowledge base). In this survey, we will review the structure-based deep models proposed for various IE tasks and also other related NLP tasks. Finally, we will discuss the limitations of the existing models and the potentials for future work.

1 Introduction

Textual materials such as books and websites are still one of the major sources of information in human societies. In the Big Data era and with the expansion of the world wide web and social networks in recent years, the amount of available textual data has also increased substantially. While on the one hand the sheer size of these resources provides valuable information about many topics, on the other hand, it hinders efficient information lookup. To address this limitation, one possible solution is to store the information in pre-defined structures (i.e., knowledge bases) so it can be quickly retrieved. Since converting the information available in textual resources into structured knowledge bases is tedious and the KB could quickly get obsolete, automatic approaches to extract structured information from free text is necessary. These automatic approaches are called Information Extraction (IE) methods and consist of several tasks including: 1) Identifying the real-world entities (e.g., person, company, and dates) that have been mentioned in text (i.e., Named Entity Recognition) (Nadeau and Sekine, 2007; Lample et al., 2016; Mikheev et al., 1999), 2) Assigning unique identity (e.g., entity IDs in a knowledge base) to the entity mentions in text (i.e., Entity Linking) (Lin et al., 2012; Liu et al., 2013b; Hachey et al., 2013), 3) Finding all expressions (e.g., proper nouns and pronouns) that refer to the same entity (i.e., Co-reference Resolution) (Ng and Cardie, 2002; Raghunathan et al., 2010; Lee et al., 2017) 4) Detecting the semantic relationships between entities that are specified in text (e.g., ownership and marriage) (i.e., Relation Extraction) (Zelenko et al., 2003; Mintz et al., 2009; Lin et al., 2016) and 5) Finding information about incidents referred to in text (e.g., divorce and attack); this information might answer questions like "*who did what to whom?*" (i.e., Event Extraction) (Ritter et al., 2012; Ahn, 2006; Nguyen et al., 2016a).

In more than the last two decades, extensive research has been conducted to design effective methods for each of the aforementioned IE tasks. These techniques range from rule-based methods (Eftimov et al., 2017), to feature-based models (Zhou et al., 2005a) and recent advanced deep learning models (Rao et al., 2017). As it has been shown in other NLP tasks including text summarization (Mani et al., 1998), document classification (Zhang et al., 2020a), question answering (Qiu et al., 2019) and machine translation (Ma et al., 2019), incorporating the existing structures into deep models for IE could improve their performance. The employed structure could either refer to syntactic structure, e.g., dependency tree (Bunescu and Mooney, 2005), semantic structure, e.g., entity similarity graph (Min et al., 2012), or external structures (e.g., knowledge base) (Fang et al., 2020). In this survey,

we study techniques that employ structure-based modeling to improve performance on various IE tasks. In addition, we review the application of text structure in other related NLP tasks. Finally, we discuss their limitations and the possible directions for future work.

2 Named Entity Recognition

Named entity recognition (NER) is the first task in the information extraction pipeline and it aims to identify words or phrases that refer to people, organizations, locations, etc. This task has been extensively studied in the more than last two decades. Approaches for this task extend from the unsupervised rule-based methods (Collins and Singer, 1999), to the supervised feature engineering (Zhou and Su, 2002) and the advanced deep learning models (Dernoncourt et al., 2017). Two sub-tasks for this problem should be solved:

- Named entity recognition: The first step for NER is to identify the sub-sequences of the input text that refer to real-world entities. For instance, in the input text *Kabul is controlled by President Abdol Mosharaf's government, which Taleban is fighting to overthrow*, the model should identify the phrases *Kabul*, *Abdol Mosharaf* and *Taleban* as the named entity mentions.

- Named entity classification: The next step for NER is to classify the recognized named entity mentions to one of the pre-defined types. For instance, in the aforementioned example, the model should be able to classify *Kabul* as *Location*, *Abdol Mosharaf* as *Person* and *Taleban* as *Miscellaneous*.

Most of the existing models address the two tasks simultaneously. However, some of the prior work proposes a different model for each task. For instance, (Collins and Singer, 1999) introduced a new rule-based model to predict the named entity type using the spelling of the name and the context in which it appears. The spelling rule might use some look-up tables or predefined patterns (e.g., the existence of *Mr* indicates the type *Person*). On the other hand, the contextual rules could refer to dependencies between a type and some indicative words in the surroundings of the named entity (e.g., *president* in the above example). Similar rules have been used in a subsequent work (Zhou and Su, 2002), however, in (Zhou and Su, 2002) authors employed Hidden Markov Model (HMM) to simultaneously identify the named entity mentions and their types. The HMM model is able to consider the previous tags and also the features of the current word to predict its label.

While the feature-based models have gained some improvements on NER, the state-of-the-art models are now employing deep learning models. NER models can benefit from the pre-trained word embeddings (Nguyen et al., 2016b) and the non-linearity of the deep learning-based models (Li et al., 2020a). More interestingly, these models could also incorporate structural information resulting in better performance on NER (Jie and Lu, 2019; Aguilar and Solorio, 2019; Yu et al., 2020). The structure could refer to the syntactic parse tree. In (Jie and Lu, 2019), authors show that a deep model enhanced with the dependency tree could have two advantages: 1) The syntactic connection between the words is indicative of the entity type, 2) the long-dependencies captured from the dependency tree could improve the representation of the words for the NER task. In this work, the authors proposed an LSTM-CRF model to capture this information. More specifically, the representation of the words to the LSTM model is enhanced with the representation of their parents and the dependency relation between them. Afterward, the interaction between the parent and its child is models via an interaction function (e.g., dot product or a feed-forward neural net) over the corresponding hidden states of the parent and children from the LSTM layer. In another recent work (Aguilar and Solorio, 2019), authors propose to encode the syntactic structure using Tree-LSTM. Furthermore, they introduce local and global attention. The local attention highlights important words with respect to the current word that is being evaluated. On the other hand, global attention emphasizes the important words of the sentence without restricting the attention to the current word.

3 Entity Linking

Entity linking (EL) is the task of identifying the corresponding entities in a knowledge base (KB) for every entity mention in the text. For instance, in the given example *Michael Jordan has recently signed a new contract with his new club. He will be the first goalkeeper of the Rangers for two years.*, there are two entity mentions: 1) Michael Jordan and 2)

Figure 1: Graph of entities for entity linking (Cao et al., 2018)

Rangers. Both of these entity mentions could refer to multiple entities (e.g., Michael Jordan could refer to the English goalkeeper, American football offensive lineman or American former professional basketball player and the Rangers could refer to the entities Rangers football club in South Africa, an association football club from Glasgow or Rangers football club in New Zealand). The correct mapping between the entity mentions and the entities in the KB depends on the context and the relation between entities in the KB.

As KBs are structured, this task inherently benefits from encoding the structure (here, structure mainly refers to the relationships between entities in the KB). While several feature-based models have been proposed for EL (Ji and Grishman, 2011; Veyseh, 2016; Khalife and Vazirgiannis, 2018), the state-of-the-art performances are achieved using deep learning models (Wu et al., 2019; Yamada and Shindo, 2019; Fang et al., 2019). Most of the existing work breaks down EL into two sub-tasks (Sevgili et al., 2020):

- Candidate Generation: Using string similarity or descriptions available in KB, a list of candidate entities is generated (Sevgili et al., 2020). For instance, for the given example, the model compares the similarity between the entity mention *Rangers* and the entities in the KB to extract the list of Rangers football club in South Africa, an association football club from Glasgow or Rangers football club in New Zealand. Authors in (Zwicklbauer et al., 2016; Le and Titov, 2019) use a simple string-match to find the list of entities. Others might use the aliases for the KB entities computed from the knowledge base metadata (e.g., redirect pages in Wikipedia) (Fang et al., 2019) or pre-calculated prior probabilities (e.g., computed from mention-link count statistics) (Ganea and Hofmann, 2017)
- Entity Ranking: Based on the consistency between the context of the entity mention and the representations of the entities, the candidate entities are ranked to choose the entity with the highest score (Sevgili et al., 2020). For instance, in the given example, the model will choose the entity *an association football club from Glasgow* as the most likely entity among the extracted list of possible entities for the entity mention *Rangers*

For the second sub-task, to represent the entities and encode their similarities with the entity mentions in the text, KB structure and the relationship between different entity mentions in the text could be helpful and the recent work has shown that deep graph architectures are able to efficiently encode this information (Fang et al., 2020; Wu et al., 2020a). One of the early works that applied graph convolution network (GCN) for entity linking is (Cao et al., 2018). They employed GCN to model the coherency between the candidate entities. In order to handle the large number of entities in the KB, they proposed to apply GCN only on the subset of entities extracted in the first phase (i.e., candidate generation). An example of the graph is shown in Figure 1. In another work, authors in (Fang et al., 2020) proposed a graph attention network to attend to the previous and next entity mentions in the text to encode the sequential inter-dependencies between the entity mentions in the text. Authors in (Wu et al., 2020a) propose to dynamically compute and refine the graph structure to model the dependencies between entities. The dynamic graph computation has been shown to be effective for other related tasks too (Nan et al., 2020). In this method, the representation of the nodes is used to compute the structure of the graph for the next iteration of the graph convolution network.

4 Coreference Resolution

Coreference resolution (CR) is a fundamental task of IE whose goal is to identify different entity mentions in the document that refer to the same entity. For instance, in the example *"I voted for Nader because he was most aligned with my values," Sara said*, there are three entity mentions for the person *Sara* (i.e., *I*, *my* and *Sara*) and two entity mentions for the person *Nader* (i.e., *Nader* and *he*). A CR model should be able to find the chain of entity mentions for the entities *Sara* and *Nader*.This task is crucial for many downstream applications including Relating Extraction and Question Answering.

According to (Stylianou and Vlahavas, 2019), traditional methods for CR can be categorized into four categories:

- Mention-pair: This method determines if a pair of mentions refer to the same thing. This method employs the features of the two mentions and performs a binary classification (Soon et al., 2001).
- Mention-ranking: In this category, models collectively consider all mentions to resolve a specific mention. More specifically, for each mention, all candidate antecedents are ranked and the one with the highest score is selected to be chained to the current mention (Rahman and Ng, 2009).
- Entity-based methods: Models of this category employ clustering techniques to decide if two clusters of mentions should be merged or not (Ratinov and Roth, 2012).
- Latent structure models: These models create a hierarchy of the mentions to collectively cluster them (Björkelund and Kuhn, 2014). The major difference between entity-based and latent structure models is that, contrary to the former which employs agglomerative clustering, in the latter, the clusters are created in a tree-like structure.

Similar techniques have been also employed in deep learning models (Wiseman et al., 2015; Clark and Manning, 2016; Lee et al., 2018). In addition, some deep learning models formulate this task as question answering (Wu et al., 2020b) or they use reinforcement learning to perform this task (Fei et al., 2019). While the traditional methods have proven the importance of text structure (i.e., dependency tree) for this task (Lappin and Leass, 1994; Björkelund and Kuhn, 2014), only recently syntactical structure has been used in deep models (Fang and Jian, 2019). Authors in (Fang and Jian, 2019) proposed to use the syntactic structure of the sentence for Chinese coreference resolution. The syntactic structure has three purposes in this work: (1) To filter out unlikely entity mentions. More specifically, they keep only those candidate entity mentions (i.e., spans of words) that are represented by a node in the syntactic tree; (2) To represent the context. In particular, the syntactic tree traverse is employed to gather the syntax-based context for each entity mention (i.e. node in the syntactic tree); (3) Encode structural features (e.g., degree of the node or its siblings).

5 Relation Extraction

Relation extraction (RE) is the task of identifying the semantic relation between entity mentions in the text. For instance, in the given example *Some Arab countries also want to play a role in the stable operation in Iraq but are reluctant to send troops because of political, religious and ethnic considerations, the official said*, a relationship of *Organization-Affiliation* is mentioned between entities *Arab countries* and *troops*. An RE model should be able to extract the relationship between different entity mentions or decide whether or not the entities of interest are in a relation.

This task has been extensively studied and several settings for that have been proposed including single-sentence, document-level, distantly supervised, end-to-end, and cross-domain. In this survey, we first review the most important existing works and datasets for each of these settings of RE. Afterward, we provide details on the existing structure-aware deep RE models.

5.1 Single-sentence

In this setting, the input to the model will be only one sentence consisting of at least two entity mentions. The goal is to predict the relation type between every pair of entity mentions in the input sentence. For this setting, the major existing datasets include ACE (Doddington et al., 2004), TACRED (Zhang et al., 2017b) and SemEval 2010 Task 8 (Hendrickx et al., 2009). The ACE dataset is a series of datasets, i.e. ACE 2003, ACE 2004, ACE 2005, ACE 2007, and ACE 2008, released

ACE 2003			ACE 2004		
Type	**Subtype**	**Count**	**Type**	**Subtype**	**Count**
AT	based-in	496	PHYS	LOCATED	745
	located	2879		NEAR	87
	residence	395		PART-WHOLE	384
NEAR	relative-location	288	PER-SOC	BUSINESS	179
PART	other	6		FAMILY	130
	part-of	1178		OTHER	56
	subsidiary	366	EMP-ORG	EMPLOY-EXEC	503
ROLE	affiliate-partner	219		EMPLOY-STAFF	554
	citizen-of	450		EMPLOY-undetermined	79
	client	159		MEMBER-OF-GROUP	192
	founder	37		SUBSIDIARY	209
	general-staff	1507		PARTNER	12
	management	1559		OTHER	82
	member	1404	ART	USER/OWNER	200
	other	174		INVENTOR/MANUFACTURER	9
	owner	274		OTHER	3
SOCIAL	associate	119	OTHER-AFF	ETHNIC	39
	grandparent	10		IDEOLOGY	49
	other-personal	108		OTHER	54
	other-professional	415	GPE-AFF	CITIZEN/RESIDENT	273
	other-relative	86		BASED-IN	216
	parent	149		OTHER	40
	sibling	23	DISC	DISC	279
	spouse	89			

Figure 2: Statistics of relation types and sub-types in ACE 2003 and ACE 2004 (Pawar et al., 2017)

by NIST for the entity, relation, and event extraction. The statistics of the relation types for ACE 2003 and ACE 2004 are provided in Figure 2. The SemEval 2010 Task 8 dataset provides 8,853 instances for 9 relation types. The relations in the SemEval dataset is directed meaning that the total number of relations will be 18 plus one special relation (i.e., *Other*) for entities that are not in a relation. The corpus to be annotated for the SemEval dataset is obtained via a pattern-based search for each relation type from the Web. The statistics for each relation type is provided in Figure 3. Despite the vast application of these two datasets for sentence-level relation extraction, there are at least two limitations in them. First, these datasets cover a limited number of relation types (i.e., 19 relations in SemEval and 24 relations in ACE 2003 and 2004 datasets). This small number of relations will not represent the challenges in a real-world application of RE. Second, the common issue in both ACE and SemEval datasets is that these datasets are relatively small for data-hungry deep learning models. In other words, this small size prevents the models from more effective feature extractions from the data. To address this limitation, authors in (Zhang et al., 2017b) proposed a new large-scaled sentence-level relation extraction dataset, i.e., TACRED. This dataset contains 106,264 examples (both positive (i.e., examples in which the two entities are in a relation) and negative (i.e., examples in which the entity are not in a relation)) in 42 relation types. The annotation is conducted over the TAC

Relation	Freq	Pos	IAA
Cause-Effect	1331 (12.4%)	91.2%	79.0%
Component-Whole	1253 (11.7%)	84.3%	70.0%
Entity-Destination	1137 (10.6%)	80.1%	75.2%
Entity-Origin	974 (9.1%)	69.2%	58.2%
Product-Producer	948 (8.8%)	66.3%	84.8%
Member-Collection	923 (8.6%)	74.7%	68.2%
Message-Topic	895 (8.4%)	74.4%	72.4%
Content-Container	732 (6.8%)	59.3%	95.8%
Instrument-Agency	660 (6.2%)	60.8%	65.0%
Other	1864 (17.4%)	N/A[4]	N/A[4]
Total	10717 (100%)		

Figure 3: Annotation Statistics for SemEval 2010 dataset. Freq: Absolute and relative frequency in the dataset; Pos: percentage of "positive" relation instances in the candidate set; IAA: inter-annotator agreement (Hendrickx et al., 2009)

KBP evaluations from 2009 to 2015. The annotation refers to the relations between organizations, people, and locations.

The traditional methods for sentence-level relation extraction use feature-based and statistical models (Zhou et al., 2005b; Bunescu and Mooney, 2005; Sun et al., 2011; Chan and Roth, 2010). The major limitation of the feature-based models is that it requires extensive feature engineering efforts and domain knowledge to find the effective patterns for the relation mentions. Moreover, these models cannot generalize well to unseen data. To address these limitations, deep learning models are employed for RE and they have gained considerable attention from the community (Zeng et al., 2014; Nguyen and Grishman, 2015a,b; Zhou et al., 2016; Wang et al., 2016a; Nguyen and Grishman, 2015a; Zhang et al., 2017b; Nguyen and Nguyen, 2018b). Using

deep architectures, e.g., Convolutional Neural Net (CNN) and Long Short-Term Memory (LSTM), along with the background knowledge provided via word embeddings, deep models reached the state-of-the-art performance on different datasets. In addition, some of the deep models embrace the findings of the feature-based models to improve RE performance. For instance, using dependency trees in deep learning models has been shown to be effective for deep learning-based RE models. (Xu et al., 2015; Liu et al., 2015b; Miwa and Bansal, 2016; Zhang et al., 2018). For this purpose, graph neural networks (GNN) could be employed to model the dependency structure. Zhang et al. (2018) proposed one of the early GNN-based models for RE. One of the major challenges to employ the dependency tree in a deep model is that neural models operating directly on parse trees are usually difficult to parallelize and thus computationally inefficient (Zhang et al., 2018). To address this issue, the prior work pruned the dependency tree to keep only the words on the shortest dependency path (SDP) between the two entity mentions in the dependency tree. However, such simplification will result in loss of information as some of the words off the path could be also important. To address this issue, Zhang et al. (2018) proposed to use graph convolution networks (GCN) (Kipf and Welling, 2016). GCNs are able to efficiently encode the graph structures with the parameter sharing. In order to improve the performance, Zhang et al. (2018) also proposed to prune the dependency tree along with the SDP up to a pre-defined distance between the off-the-path and on-the-path words. Their evaluations of TACRED dataset prove the effectiveness of this method.

5.2 Document-level

In this category of RE models, the input to the system is a document consisting of multiple entities. Entity mentions might appear in one sentence or across multiple sentences in the given document. In general, the relation mentions in documents could be categorized into two groups: 1) Intra-sentence relations: If both entity mentions that are in relation are mentioned in the same sentence, the relation between them is an intra-sentence relation; 2) Inter-sentence relations: In this category, the two entity mentions appear in different sentences across the document. For instantce, in the given document *Elias Brown (May 9, 1793– July 7, 1857) was a U.S. Representative from Maryland. Born near Baltimore, Maryland, Brown attended the common schools. He died near Baltimore, Maryland, and is interred in a private cemetery near Eldersburg, Maryland.*, the relation between the entity *U.S.* and *Maryland* is *COUNTRY* and the relation between the entity *Maryland* and *Baltimore* is *LOCATED_IN*. As both relations can be inferred from the immediate sentence in which the entities appear, the two mentioned relations are intra-sentence relations. On the other hand, the relation between the entity *Baltimore* and *U.S.* is *COUNTRY* that should be inferred from the different sentences in which the entity mentions appear. Thus, this relation is of type inter-sentence relations.

While there are some domain-specific (Li et al., 2016a) or distantly supervised (Quirk and Poon, 2016; Peng et al., 2017) document-level relation extraction datasets, the only large scale manually labeled document-level relation extraction dataset available is provided by (Yao et al., 2019). This dataset, called DocRED, contains 56,354 relation facts and 132,357 entity annotations across 5,053 Wikipedia documents. Among all relation facts, 40.7% of them are inter-sentence relations which require inference in document level.

The major challenge for document level relation extraction is to infer the long range dependencies between the entities across sentences. To deal with this issue, most of the existing work propose to employ structure-based modeling. More specifically, a structure that could represent the dependencies between different parts of the document is constructed, either using some heuristics (Zeng et al., 2020) or it is learned by a trainable component (Nan et al., 2020). In order to infer a task specific structure for document-level RE, authors in (Christopoulou et al., 2019) proposed to infer the document structure from the representations of its edges. More specifically, they first create a dense graph whose vertices are the entity mentions, sentences and the entities (i.e., the people, organizations, etc that have been mentioned in the document). The entity mentions are represented using their corresponding hidden states of a bi-directional LSTM (BiLSTM) network. The sentence and the entity representations are computed by pooling the representations of all words or mentions of them, respectively. Afterwards, the representations of edges of the graph are obtained using the representation of their heads and tails. Finally, to compute the representations for longer paths (e.g., paths

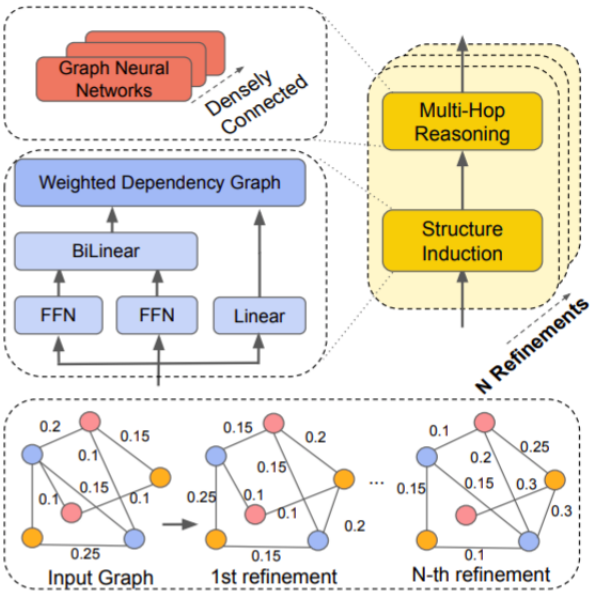

Figure 4: Overview of the dynamic reasoner model in (Nan et al., 2020)

consisting of two edges), a feed forward neural net is employed to combine the representations of all edges in that path. The path representation between the two entity mentions of interest is used to predict the relation. While this work proposed a method to infer the structure-based entity mentions relations, it fails to dynamically update the representations of the nodes, including the entity mentions themselves. To address this issue, authors in (Nan et al., 2020) proposed a structure inference mechanism to dynamically and consecutively update the node representation and the graph structure, in turn. More specifically, after obtaining the representations of the nodes[1], the weights of all edges in the dense graph are computed from the head and tail representation of the edge. Afterwards, a GCN layer is employed to update the node representations. Using the updated representation of the nodes, a new set of weights for edges of the graph is computed. This process is repeated for N times. Finally, the representation of the two entity mentions are used for relation prediction. Figure 4 shows a diagram of this model.

Most recently, authors in (Wang et al., 2020a) proposed another saturate-based document-level relation extraction model. In the proposed approach, authors first construct a set of nodes based on the sentences, entities, and mentions. Afterwards, similar to prior work, they connect the nodes based on some heuristics (e..g, if a mention is hosted by a sentence there would be connection between the corresponding sentence node and the mention node). Using the obtained global graph and a GCN model, authors update the initial representations of the nodes which are obtained from a sequence-based encoder. In the next step, the representations of the nodes are updated using multi-head self-attention component. This component could capture the semantic dependencies between the extracted nodes, i.e., sentences, mentions and entities. Finally, by concatenating the representations obtained from the GCN layer and the self-attention layer for the two entities of interest, the final representation vector is constructed and it is consumed by a logistic regression classifier to predict the semantic relations between the two entity mentions in the document. A diagram of this model is shown in Figure 5.

5.3 Distantly Supervised

One of the major challenges for RE is that collecting training data is expensive. Thus, the exiting datasets are quite small, specifically for data-hungry deep models. One remedy to this issue could be to use distantly supervised (DS) datasets. In this setting, some heuristics are employed to collect examples for pre-defined relation sets. In the seminal work (Mintz et al., 2009), authors employed the relations between entities in Freebase knowledge base and an unlabeled corpus to extract examples for each relation. For instance, consider the two entities *Steve Jobs* and *Apple*. Suppose that the relation between this two entity mention in the KB is *Works_At*. Based on the method proposed by (Mintz et al., 2009), one could extract examples for relation *Works_At* by extracting all sentences in a large corpus (e.g., Wikipedia) that contains mentions for both entities *Steve Jobs* and *Apple*

While the distantly supervised RE dataset could extend the size of training sets, they also introduce noisy examples. More specifically, sentences that contain both entities of interest but do not mention the supposed relation between entities are incorrectly labeled. This examples are indeed the false positives. Due to this problem, a distantly supervised RE model should be able to deal with the noisy example which might be extracted in this process. To this end, several techniques are proposed to exclude or rectify the incorrectly labeled

[1] in this work, entity mentions, the words on the SDP between every pair of entity mentions and the entities themselves serve as the nodes of the graph

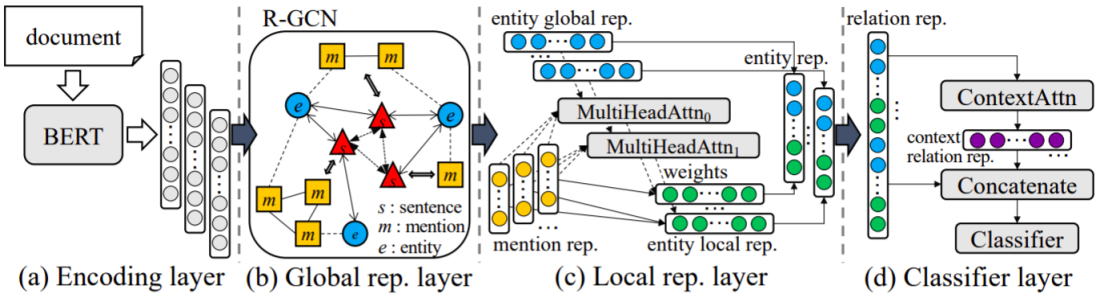

Figure 5: Document-level relation extraction by global and local graph encoding. The global-level graph encoding is fulfilled by a graph convolution layer. The local-level graph encoding is obtained by using a self-attention component (Wang et al., 2020a).

samples in the training data. Some of the prior works exploit reinforcement learning (RL) to identify the incorrectly labeled samples. Feng et al. (2018) introduced a two-module RE model. The first module is an instance selector which identifies the instances with incorrect labels and filter them out. The second module is a relation classification model which use the input training data to learn the RE task. The reward for the instance selector is computed using the performance of the second component on the evaluation set. In a similar approach, Qin, Xu, and Wang (Qin et al., 2018) proposed to employ RL to denoise the training data. However, in their method, instead of excluding the noisy samples, they suggested to change the label of the false positives to *None*, indicating there is no relation between the two entity mentions in the sentence.

One issue with the RL-based approaches is that they make a hard decision to either exclude or change the label of noisy samples. In other words, during the training of the relation classifier, the hard labels of the noisy samples might be detrimental for the training process. In order to alleviate the effect of the incorrect hard labels, Liu et al. (2017) introduced a soft-label multi-instance learning method for relation extraction with noisy training samples. In this method, all samples of a pair of entity mentions h_i and t_j are grouped into the set $< h_i, t_j >$ consisting of c sentences $S = \{x_1, x_2, \ldots, x_c\}$. The set $< h_i, t_j >$ could be represented either by only one of the sentences in S or an attention-based pooling of the sentences. Afterwards, to obtain the label for the set $< h_i, h_t >$, instead of using the one-hot $L_{i,j}$ vector label obtained from the distantly supervised dataset, they proposed to learn a dense vector $\bar{L}_{i,j}$ from the bag representation and

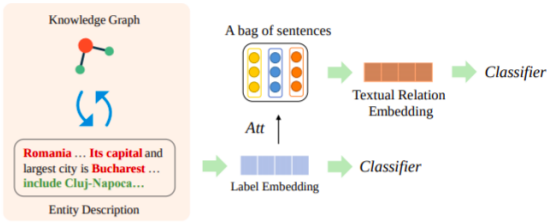

Figure 6: Knowledge graph structure employed to compute attention scores for every sentence in the noisy training set (Hu et al., 2019)

the one-hot vector $L_{i,j}$. The soft label $\bar{L}_{i,j}$ will be used in the next epoch by the relation classifier as the gold label for the set $< h_i, t_j >$.

For the distantly supervised relation extraction setting, the structure-based modeling has been also shown to be effective. The graph-based models employed for DS relation extraction encode the structure of the knowledge graph. More specifically, the structure of the knowledge base is employed to model the interaction between the entities, thereby, denoise the samples for every pair of entity mentions. For instance, authors in (Hu et al., 2019) proposed to employ the knowledge graph structure to learn an embedding vector for each relation type. More specifically, using the graph encoding method proposed by (Bordes et al., 2013), they learn the representation of the head (h), tail (t) and relation (r) of the triples $< h, r, t >$ in the knowledge graph. Afterwards, using the representation of the relations in the training set, an attention score is computed for each sentence in the training set. The attention-based representation of the sentences are employed by the relation classifier. Figure 6 shows the diagram of this model.

In addition to the application of the graph structure for denoising the samples in DS datasets,

some researchers have employed graph structure to learn the dependencies between relations predicted for an entity pair (e_1, e_2) from a set of sentences $S = \{s_1, s_2, \ldots, s_n\}$. Two relation are dependent on each other, if the existence of one infer the existence of another. Note that it would be a directed dependency. For instance, *President_of* between a person and a country could also induce the relation *Lives_in* between the person and the country. To encode this dependency, authors in (Shang et al., 2020) proposed to build a graph structure where the nodes are the relation types and the edges could represent the dependencies between them. During training the model is optimized to learn a dependency relation graph for every pair of entities that could represent the gold relations between the two entity.

5.4 End-to-end

Relation extraction is the task of identifying the relations between entity mentions in text. To this end, the entity mentions should be first identified. While a pipeline approach identifies the entities and relations in separate stages, the major limitation is that the errors in the entity recognition stage could be propagated to the relation extraction stage. In order to prevent this error propagation, an end-to-end (E2E) RE model jointly recognizes the entity mentions and the the relation between them in a given text snippet.

While the sentence-level relation extraction datasets (e.g., ACE or SemEval 2010 Task 8) could be used to train and evaluate an E2E RE model (Miwa and Bansal, 2016), for this setting, most of the recent work report the performance of the models on NYT (Miwa and Bansal, 2016) and WebNLG (Gardent et al., 2017) datasets. NYT was originally proposed to address the high level of noise in the datasets prepared by the distant supervision technique (Mintz et al., 2009). To this end, they proposed a semi-supervised method to extract relation tipples (i.e., (*entity*$_1$,*relation,entity*$_2$)) from New York Times using the Freebase as the knowledge base. WebNLG is a corpus created using a natural language generation (NLG) framework operated on the DBpedia knowledge base.

Although prior work for E2E RE employed feature-engineering methods (Nguyen and Moschitti, 2011), recent deep models are proved to achieve the state-of-the-art results for this task (Miwa and Bansal, 2016). Moreover, in the recent work (Fu et al., 2019), authors have shown that the structure-based modeling could improve the performance of an E2E RE system. In particular, two graph structures are employed in this work: (1) The syntactic tree of the sentence is employed by a graph convolution network (GCN) to enrich word representations The syntax-enriched word representations are employed to predict the entities and also the relation types between words; (2) A full-graph consisting of the words as the nodes and the pair-wise relations between words as the edges is created. In this graph, the edges (i.e., relations) that are predicted in the first stage (i.e., using the dependency based GCN) are emphasized by giving more attention weights to them. The main purpose of this graph is to encode the relation dependencies between words. Specifically, for those relations that share an entity (e.g., the head entity), the dependency between them could be encoded by the GCN layer to infer the direct relation between the other entities (e.g. the tail entities). For instance, if the triples *(BarackObama,LiveIn, WhiteHouse)* and *(WhiteHouse, PresidentialPalace, UnitedStates)* are predicted in the first stage, the second stage employ GCN to infer the third triple *(BarackObama, PresidentOf, UnitedStates)*. In addition, in order to predict multiple relations between every pair of entities, authors proposed to use a threshold in which every relation type r between the pair of words w_1 and w_2 (i.e., two predicted entity mentions), predicted in the second phase, will be included in the final model's prediction to create the triple (w_1, r, w_2).

5.5 Cross-domain

While the aforementioned settings suppose that the training and the evaluation data come from the same domain, it could not be guaranteed in all scenarios. For those cases that the RE model is trained and evaluated in different domains, a cross-domain RE system is required. The main challenge of such a setting is that the features that are useful during training might not be relevant or helpful in the evaluation phase. To train and evaluate models in this setting, the ACE 2005 datasets is widely used. In this dataset, there are 6 different domains, i.e., (*bc, bn, cts, nw, un,* and *wl*), covering text from news, conversations and web blogs. Cross-domain models are trained on one of these domains (e.g., news) and are evaluated on the other domains (e..g, conversations and web blogs). Similar to the other

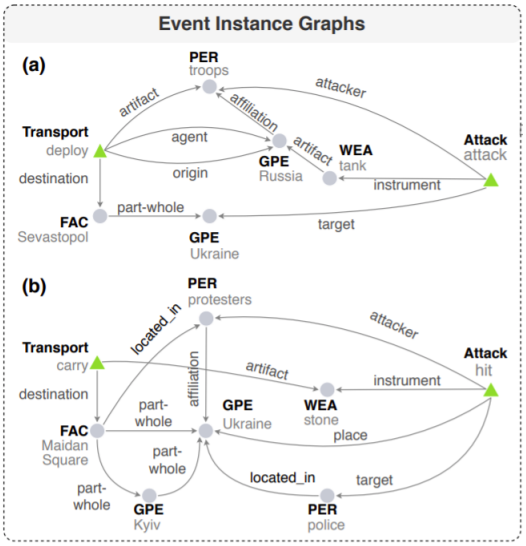

Figure 7: Event Graph from a news article. The triangle nodes represent the events and the circle nodes represent the entities. The edges between the event node and the entity node show the role of the entity (i.e., argument) in the corresponding event. The edges between two entities are the relation between them. (Li et al., 2020c)

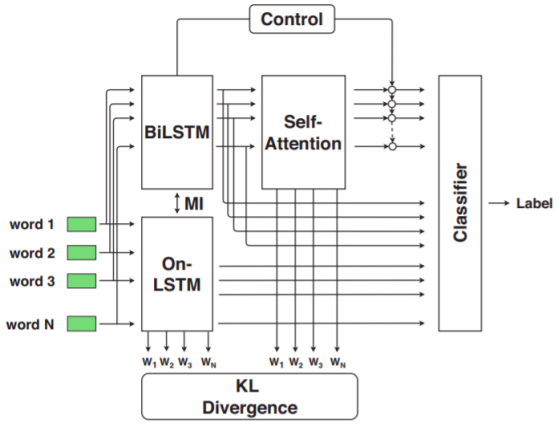

Figure 8: Relation extraction in cross-domain setting using structure inference (Veyseh et al., 2020b)

settings, for cross-domain RE, prior work started to employ feature-based models (Yu et al., 2015). However, deep models are proved to be more effective for this setting (Nguyen and Grishman, 2015a). Until recently, the graph-based deep model have not been explored for this task. Recently, Veyseh et al. (2019a) have shown that the structure of the text (e.g., dependency tree) could be used to improve the performance for cross-domain RE. Also, in the recent work (Veyseh et al., 2020b), they have employed deep learning to infer the structure of the text without using off-the-shelf parsers. More specifically, they propose to employ two deep architectures, i.e., ordered-neuron LSTM (Shen et al., 2018) and self-attention mechanism (Vaswani et al., 2017), to infer two views of structure of the input sentence. Afterwards, by exploiting a neural-based mutual information estimator (Belghazi et al., 2018), they increased the consistency between two structural views. Their evaluation on ACE 2005 dataset show that this techniques achieves the state-of-the-art results for cross-domain relation extraction.

6 Event Extraction

Event extraction is the task of identifying real word incidents mentioned in text such as *attack*, *divorce*, or *birth*. According to the ACE annotation guidelines, an event is described as something that happens and change the state of an entity. For instance, the sentence *Ames recruited her as an informant in 1983, then married her two years later*, implies that the marriage status of *Ames* is changed so it refers to an event of *marriage*. According to ACE annotation guidelines, every event mention consist of two components:

- Trigger: This is the word or phrase which most clearly express the occurrence of the event. It could be a verb, noun or adjective. For instance, in the sentence *John robert bond was born in England*, the verb *born* is the event trigger which indicates the occurance of the event *BE-BORN*. Note that each event trigger evokes a specific incident known as event type.

- Argument: Those entities that are participants of the event and their states are changed due

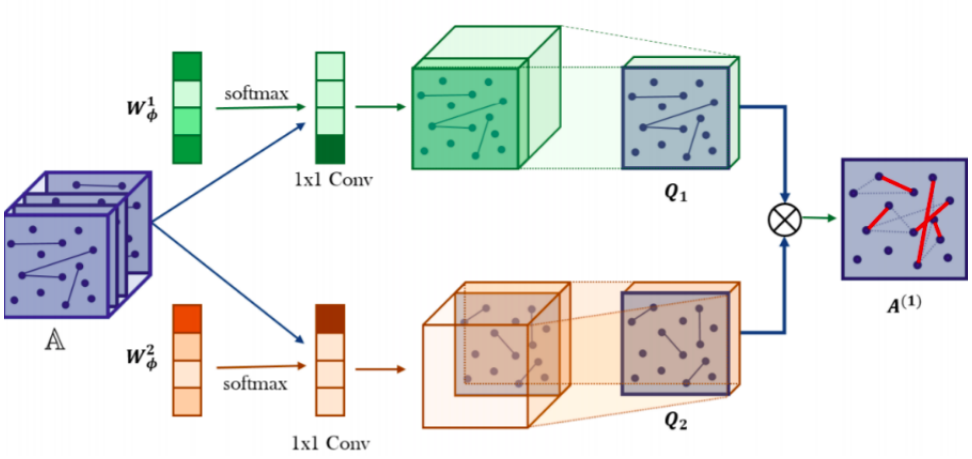

Figure 9: Graph Transformer Network (Yun et al., 2019)

to the occurrence of the event are considered as the event argument. In addition to the event participants, the other attributes of the event, e.g., time or location of the event, are also considered as the event arguments. For instance, in the sentence *The man accused of killing seven people near Boston on Tuesday got his guns in Massachusetts*, there is an event mention of *Kill*. The trigger word for this event is *killing* and the arguments of this event are *man*, *seven people*, *Boston* and *Tuesday*. It is worth noting that each argument takes a specific role in the event. For instance, in the given example, the role of the argument *seven people* is *victim* and the role of the argument *Boston* is *place*.

The task of identifying the trigger and its type is known as Event Detection (ED) and the task of identifying the event arguments and their roles is known as Event Argument Extraction (EAE). For each of this tasks there is a wealth of prior work extending from feature-based models (Ahn, 2006; Ji and Grishman, 2008; Patwardhan and Riloff, 2009; Liao and Grishman, 2010a,b; Riedel and McCallum, 2011; Hong et al., 2011; McClosky et al., 2011; Li et al., 2013; Miwa et al., 2014; Yang and Mitchell, 2016) to advanced deep learning systems (Chen et al., 2015; Sha et al., 2018; Zhang et al., 2019b; Yang et al., 2019; Nguyen and Nguyen, 2019; Zhang et al., 2020b). While most of the prior work consider sentence-level event extraction, some recent work has also introduced event extraction in document level (Ebner et al., 2019). Moreover, in addition to the text-based event extraction, there are some recent work that attempt to extract event mentions from multiple modalities (e.g., text and image) (Zhang et al., 2017a; Li et al., 2020b). Furthermore, some prior work consider the open event extraction which aims to extract the event triggers without the assumption of a pre-defined domain (i.e., ontology of event types) (Wang et al., 2019a; Sims et al., 2019; Naik and Rosé, 2020). Event extraction systems could be employed in knowledge base construction, question answering, and text summarization. In this section, we will review the important existing work and their major advantages and limitations. In the reviews of the models, we emphasize the application of text structure for event detection and event argument extraction.

6.1 Datasets

The most popular dataset among event extraction researches is ACE 2005 dataset. It has annotations for 599 documents with 6,000 labels for events (Xiang and Wang, 2019). The events are annotated with 8 types and 33 sub-types. Table 1 shows the event types and sub-types in ACE 2005 dataset. Docuemnts annotated for ACE 2005 are in English, Arabic and Chinese from six different domains, i.e., Newswire, Broadcast News, Broadcast Conversations, Weblog, Usenet News Group, and Conversational Telephone Speech. Table 2 shows the statistics of each of these domains in English section of ACE 2005 dataset.

In addition to ACE 2005 dataset, there are other datasets that are exploited by event extraction works:

- TAC-KBP: introduced by Linguistic Data Consortium (LDC) (tac, 2016), provides anno-

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

Table 1: Event Types and Sub-Types in ACE 2005 (Xiang and Wang, 2019)

Domain	Proportion
Newswire	20%
Broadcast News	20%
Broadcast Conversations	15%
Weblog	15%
Usenet News Group	15%
Conversational Telephone Speech	15%

Table 2: Domain Statistics of the English portion of the ACE 2005 (Xiang and Wang, 2019)

	CASIE	**CySecED**
# event types	5	30
# positive examples	8,470	8,014
# negative examples	240,682	282,220
# sentences per document (average)	16.69	24.94

Table 3: Statistics for CASIE and CySecED. Negative examples refer to non-trigger words while positive examples are the annotated trigger words for the event types of interest (Hieu Man Duc Trong, 2020).

tations for 360 documents with 9 event types and 38 event sub-types.

- LitBank: This dataset annotates 100 English literary texts. It includes annotations for both entities and event triggers. Unlike ACE, LitBanck does not provide event types for the triggers.

- TimeBank: This dataset, provided by LDC (Pustejovsky et al., 2003), includes annotations for events, times, and temporal relation between event mentions. Similar to LitBank, this dataset also does not provide types of the event triggers.

- Domain-Specific Datasets: In addition to the general-domain event annotation, some datasets focus on domain-specific datasets. BioNLP-ST is a collection of event mention annotations from various corpora including GENIA event corpus, BioInfer corpus, Gene regulation event corpus, GeneReg corpus and PPI corpora (Xiang and Wang, 2019; Vanegas et al., 2015; Nédellec et al., 2013). Another domain that has gained attention for event extraction is cyber-security domain. In this domain, event are categorized into four general topics: (1) Discover: Events referring to identification of a vulnerability in a system, (2) Patch: Events mentioning the fixes of a known vulnerability, (3) Attack: Exploitation of a vulnerability to impact the system and (4) Impact: consequences of an attack on a system (Hieu Man Duc Trong, 2020). For cyber-security domain, CySecED (Hieu Man Duc Trong, 2020) and CASIE (Satyapanich et al., 2020) are the largest datasets available. The statistics of these datasets are provided in Table 3.

- Multi-modal Event Extraction: In additon to text-based event extraction, some recent work proposed a new dataset for extracting events from both textual and visual data (Li et al., 2020b).

6.2 Feature-based Models

Early work on event extraction has employed feature engineering for event extraction from text. In the early stages of event extraction research, Riloff and Shoen (Riloff and Shoen, 1995), proposed a pattern-based EE system. In their system, the syntactic parse of the sentence is employed to extract general patterns for event mentions. For instance, in the sentence *World trade center was bombed by*

terrorists, identifying the subject (i.e., *Wordl trade center*), verb phrase (i.e., *was bombed*) and prepositional phrase (i.e., *by terrorists*) could lead to the event patterns *[x] was bombed* and *bombed by [y]* to identify the attack event and its arguments in text (Xiang and Wang, 2019). Based on the statistics of the patterns in the corpus, the high confident patterns are selected to be used in evaluation phase. Later in the following years, feature-based models employed statistical models such as nearest neighbors (Ahn, 2006), maximum-entropy learner (Chen and Ji, 2009), support vector machine (Saha et al., 2011), and conditional random field (Majumder and Ekbal, 2015).These models employ the lexical forms of the words, the syntactic parse (e.g., the POS tag, the parent or children of the word in the dependency tree, or the label of the dependency edges), synonyms of the words, and the event or entity type (Xiang and Wang, 2019). For a complete review of these methods, refer to the survey provided by (Xiang and Wang, 2019).

6.3 Deep Models

Despite all progress obtained from more effective features employed in statistical models, the major limitations of feature-based systems is that these models are not able to incorporate background knowledge and also to infer new useful patterns from the training data. Deep learning addresses these limitations by utilizing the word embeddings pre-trained on large corpus and also by exploiting deep architectures to induce effective patterns from the training data. Due to these advantages, the recent event extraction systems employ deep learning (Chen et al., 2015; Sha et al., 2018; Zhang et al., 2019b; Yang et al., 2019; Nguyen and Nguyen, 2019; Zhang et al., 2020b). Some of the deep models exploit sequence-based architectures (Sha et al., 2018), convolutional neural networks (CNN) (Björne and Salakoski, 2018), or recent transformer-based models (Ahmad et al., 2020).

In addition to the deep architectures and background knowledge, some recent models attempted to incorporate the interaction between event types (Li et al., 2019b) or argument roles (Wang et al., 2019b) using hierarchy-based modeling. For instance, authors in (Wang et al., 2019b), proposed to encode the hierarchy of event argument types using neural module network (NMN) (Andreas et al., 2016). In particular, the hierarchy of the event argument types are employed to capture the dependency between related argument roles. For instance, in the sentence *Steve Jobs sold Pixar to Disney in 2006*, identifying the role of the entity *Steve Jobs* as *Seller* and its hierarchical dependency with role *Buyer* (i.e., considering the fact that both *Seller* and *Buyer* are entities of type *Person* or *Organization*) could help the model to predict the role of the entity *disney* as *Buyer* (see figure 11). To encode this hierarchical information, authors proposed to train separate attention functions for each type which could be applied to the input sentence to obtain type-dependent repression of the input text. The aggregation of type-dependent representations of all possible types of an entity is used in the final classifier to predict the role of the entity. Figure 10 shows the diagram of this model.

6.4 Graph-based Models

The structure-based modeling has two applications for event extraction: 1) Text Representation and 2) Event Graph. In this section, we study each of them in details

6.4.1 Text Representation

The syntax or semantic based structures of the sentence might be employed by deep models to encode the interactions between the words, thereby improving the performance of event detection or event argument extraction. For instance, authors in (Amir Pouran Ben Veyseh, 2020) proposed to infer the task-specific syntactic and semantic structure of the input sentence using deep architectures. Specifically, the syntactic structure is induced by feeding the pair of dependency-based distance of the words to the trigger/argument into a feed forward neural net. The output of the feed forward neural net are employed as the entries of the syntax-based adjacency matrix. To infer the semantic structure of the input text, authors propose to employ self-attention mechanism (Vaswani et al., 2017). Finally, for efficient combination of the syntactic and semantic structures, graph transformer network (GTN) (Yun et al., 2019) is employed. GTN uses convolution operation to combine the structures and also encode the heterogeneous paths by multiplying the adjacency matrix of all structures. An overview of this network is shown in figure 9. One limitation of the GTN architecture is that it could result in overfitting to the training data due to the increased number of parameters for combining the structures. In order to alleviate this issue, authors in (Amir Pouran Ben Veyseh, 2020) proposed to employ in-

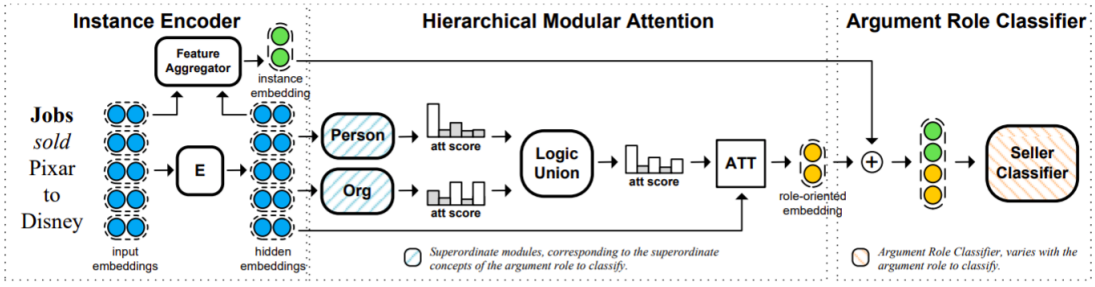

Figure 10: Hierarchical Modular Event Argument Extraction (Wang et al., 2019b)

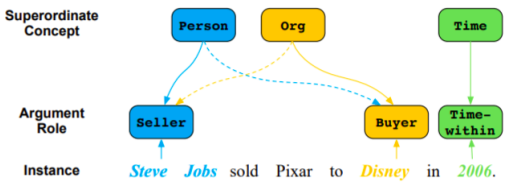

Figure 11: An example of the concept hierarchy (Wang et al., 2019b).

formation bottleneck technique. Specifically, they decrease the mutual information between the input and output of the GTN, treating this network as information bottleneck. This technique could prevents the model from memorizing patterns specific to the training data.

In another work, authors in (Li et al., 2019a) proposed to employ Tree-LSTM to encode the dependency tree of the input text. Tree-LSTM is a version of LSTM with the key difference that at each time step, the hidden states of the Tree-LSTM neurons are updated using the representation of the current word and the hidden states of all of its children in the input tree structure. In addition to the dependency tree encoded by Tree-LSTM, authors also proposed to encode the external knowledge encoded in a domain-specific knowledge base (KB) using gating mechanism added to the Tree-LSTM update rules. More specifically, firstly, for each entity in the input text, their types and descriptions are obtained from the knowledge base. Next, the entity type and description are represented using randomly-initialized embedding of their words. Note that these embeddings will be fine-tuned during training. Afterwards, using the pooled representation of the entity type and description, a new gate vector is computed. The gate vector will be employed in the Tree-LSTM to control how much information should be transferred from the children to the parent node at each time step. Diagram 12 shows the overall architecture of the proposed model.

Although the Tree-LSTM or GCN architectures employed in the above mentioned works are effective to capture the structure of the input text, the performance of these models will degrade by increasing the number of layers. This limitation prevent the model from encoding longer dependencies in the graph structure. To overcome this issue, authors in (Yan et al., 2019) proposed to encode multi-order graph structure. More specifically, they compute the graph-based representation of the input text by employing the dependency tree adjacency matrix A, the second order of the adjacency matrix A^2 and the third-order of the adjacency matrix A^3. The aforementioned adjacency matrices will be encoded using graph attention network (GAT) which is a variant of GCN. The representation obtained for each order will be aggregated using attention function atop the proposed GATs.

Another issue with prior work is that they utilize dependency tree for event extraction while ignoring the dependency relation type between words. More specifically, the dependency tree is encoded using a binary adjacency matrix in which an entry is set to 1 if there is a dependency edge between the corresponding words, otherwise the entry is set to zero. To solve this limitation, in the proposed mode by (Cui et al., 2020), authors suggest to model the structure of the sentence by encoding the dependency relations between words. More specifically, instead of using a binary adjacency matrix to encode the dependency tree, authors employ the tensor E of the dimension $n \times n \times p$ whose entry $E_{i,j}$ is a vector of size p, i.e., the total number of dependency relations in the dependency tree. Moreover, each edge in the dependency tree is represented with a randomly initialized vector. The words of the sentence are also encoded by the high-dimensional vectors obtained from a BiLSTM network. Next, to update the word representations,

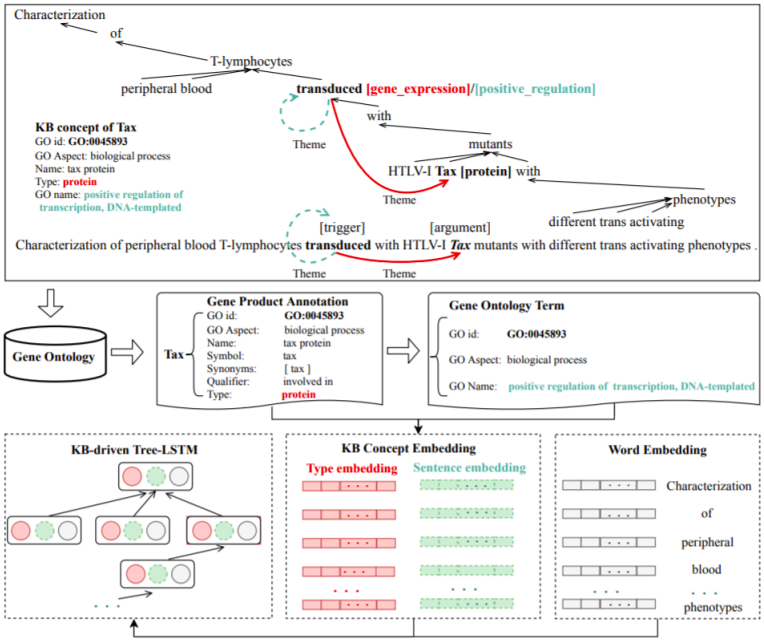

Figure 12: Knowledge-based event extraction. The upper component represent the depednecy tree and the knowledge-base concept representations for the argument *Tax*. The middel component shows the KB information for the concept *Tax*. The lower component shows the KB-drivent Tree-LSTM model (Li et al., 2019a).

each channel of the adjacency tensor E is employed by a GCN layer to aggregate the representations of the neighbor nodes and connecting edges with the respect to the relation type corresponding to the selected channel. Finally, using the updated representations of the nodes and the previous state of the edge vectors, the representation of each edge is updated using a feed forward neural net. By stacking of L layers of GCN and feed forward net to update the word representations and the edge representation, respectively, the final representation of the words is obtained. Finally, a feed forward classifier consumes the representations of the words to predict the event triggers. The diagram of this model is shown in Figure 16.

6.4.2 Event Graph

A document might includes several event mentions. These events could have temporal, hierarchical or causal relations with each other. For instance, Figure 15 shows an event graph constructed from a document based on temporal and causal relations between events. More specifically, the event *storm* causes three other events *killed*, *died* and *canceled*. In addition to the causal relation, this figure also shows the temporal relations between events, e.g. the event *die* has happened before the event *canceled*. To construct the event graphs, prior works takes two major steps: (1) Event mention detection which identifies the events in the document, (2) Event-Event relation extraction which aims to predict the causal or temporal relations between events.

Recently, event-event relation extraction has gained more attention. For instance, authors in (Wang et al., 2020b) proposed joint model for simultaneously predicting the temporal and causal relations between event pairs using contextualized word embeddings and common sense knowledge injection. In particular, to pre-train a model for common sense knowledge injection, they propose to construct a set of positive and negative samples for event-event relations from two knowledge base ConceptNet and TemProb. Specifically, they extract 30,000 triples from these knowledge bases and annotate them using the relation specified between them in the knowledge base. They also construct another set of 30,000 triples in which there is no relation between the head and the tail based on the knowledge base facts. Afterwards, in a contrast learning framework, they train a multi-layer perception to distinguish the triples in which there is a relation between them from the ones that are irrelevant to each other (i.e., with no relation between the head and the tail). Finally, during training of the event-event relation extraction model, the activations obtained from the pre-trained common sense knowledge network is employed as extra features

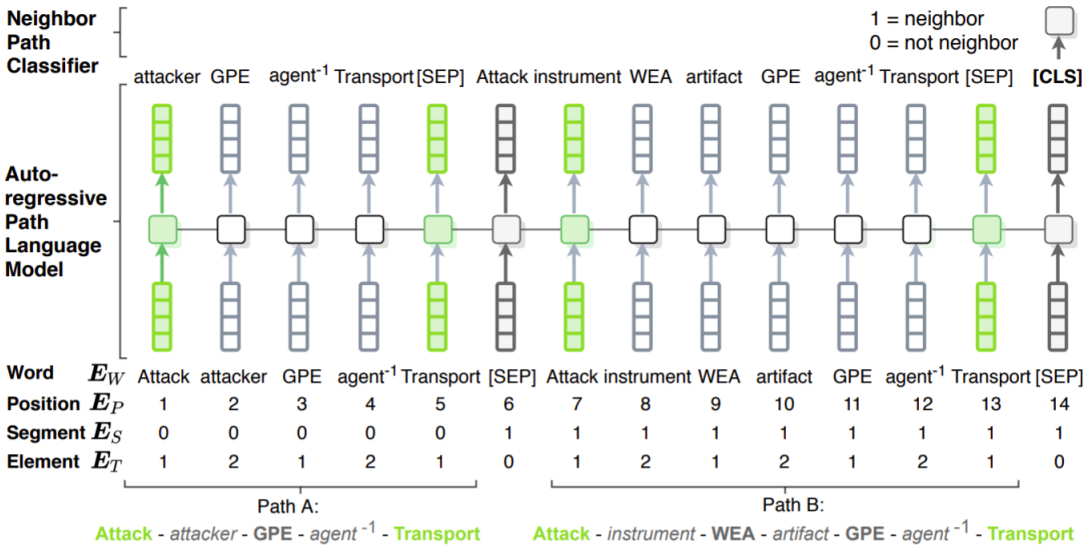

Figure 13: Autoregressive path language model with neighbor path classification (Li et al., 2020c)

to be concatenated with the features extracted for the input event-event paris. To obtain the event-event pair representations, the input document is first encoded by a pre-trained contextualized language model. Afterwards, the representations of the words in the document are concatenated with their POS tag embedding and are fed into a bi-directional LSTM (Bi-LSTM) model. Next, using the representation of the event triggers obtained from the Bi-LSTM layer and the features obtained by the pre-trained common sense knowledge network, the temporal and causal relations between every pair of events is predicted. Figure 14 shows the diagram of this model. The main advantage for joint temporal and causal relation extraction is that it could learn the features from one task that are indicative for the other task too.

In addition to temporal and causal relations between events, they might share their arguments and the arguments could have relations with each other too. These relations between events and their arguments could be encoded using graph structure. For instance, Figure 7 shows an event graph consisting of two events, their arguments and the relation between them. Identifying this graph could be helpful to recognize the co-occurring events and arguments for event extraction. Due to the importance of this task, recently it has gained attention (Wang et al., 2020c; Li et al., 2020c) Specifically, authors in (Li et al., 2020c) proposed a language-model-based approach to construct the event graph between every pair of events from all documents in a corpus. In particular, they propose to find all possible connections between a pair of events using their mentions in multiple documents. Note that connection between two events refer to any path between the events in the event graph that includes one or multiple arguments. For instance, in the Figure 7 the path *Transport,artifact,PER,attacker,Attack* is the connection between the event *Transport* and the event *Attack* via their common argument *PER* (i.e., entity of type *Person* whose role in *Transport* event is *artifact* and in *Attack* event is *Attacker*). After finding all possible connections, a language model (i.e., BERT model), pre-trained on the paths in the training set, predicts the importance of all found connections in test set. Finally, those connections that are above a threshold are selected to be used in the final event graph constructed for the two events of interest. Note that to predict the importance of a connection using the pre-trained language-model, authors propose to compute two types scores:

- Coherence and salience: This score evaluates the degree to which the candidate path is consistent with the two event types. For instance, the path *Attack, attacker, GPE, agent, Transport* should have high score with respect to coherence as it appears in the training data (See Figure 7). To train the pre-train language-model to give high score to coherent paths, authors propose to train the BERT model in an autoregressive fashion (i.e., given the previous elements of a path, the model predicts the next element)

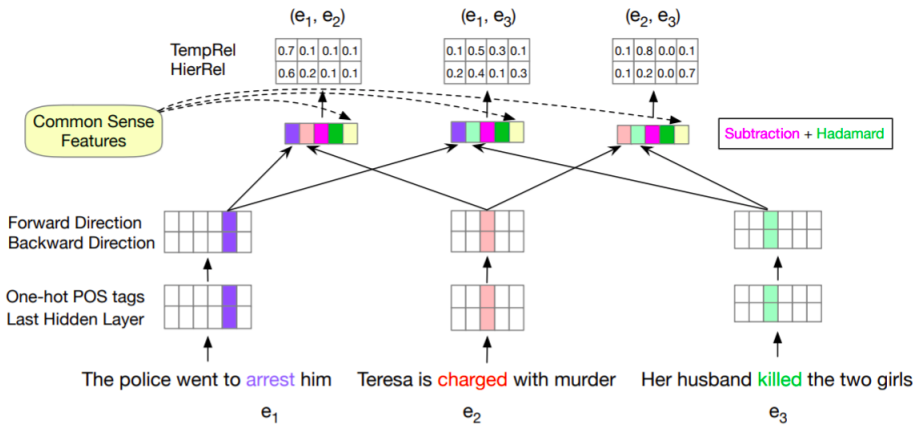

Figure 14: Joint model for causal and temporal event-event relation extraction (Wang et al., 2020b)

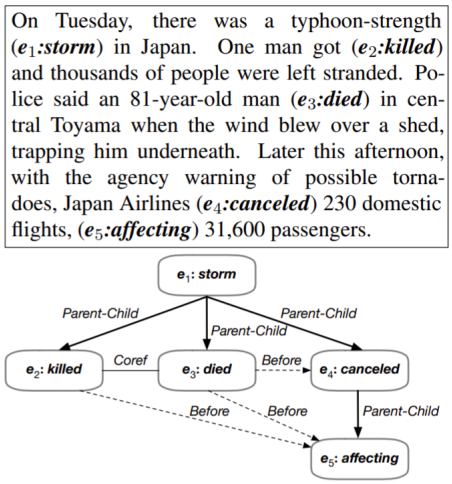

Figure 15: Event Graph constructed based on causal and temporal relations between events (Wang et al., 2020b)

- Path co-occurrence: For a pair of events, some paths are more common and they frequently co-occur with each other. For instance, the path *Attack, attacker, GPE, agent, Transport* and *Attack, instrument, WEA, artifact, GPE, agent, Transport* co-occur with each other as they both appear in the event graph shown in Figure 7. In order to train the language-model to give higher scores to the co-occurring paths, authors employ a contrasting learning objective. Specifically, they propose to construct an input sequence consisting of two paths for the BERT language model. If two paths belong to the same event graph, the input is labeled as positive, otherwise it is labeled as negative sample. Figure 13 shows the diagram of this model.

While the approach proposed by authors in (Li et al., 2020c) achieves promising results on construing the event graph, due to the breaking down of the event graph into paths, this model fail to capture any graph-level interactions between edges and nodes in the event graph. As such, a potential direction for future work is to apply deep graph models to encode the event graph.

7 Veracity

In addition to the application of structure-based modeling for traditional information extraction tasks, these models could be also useful for other sub-fields of information extraction. One of these sub-fields is information veracity. For information veracity, the goal is to evaluate how much valid or factual is a stated claim or event. This general topic can be formulated as event factuality (Rudinger et al., 2018), rumor stance classification (Veyseh et al., 2017) or rumor resolutions in social networks (Veyseh et al., 2019b). In this section, we study these tasks and the application of graph-based models for them.

7.1 Event Factuality

For event factuality, the goal is to determine the degree to which an event mentioned in text has happended. For instance, in the example *I will, after seeing the treatment of others, go back when I need medical care*, the event *go back* has not happened (it can be inferred by considering the verb *will*). This task can be formulated as either a classification or a regression task. In the classification task, the system could make a binary prediction (i.e., for *happened* or *not-happened* classes) or it could predict the levels of factuality. More specifically, the system could predict one of the classes

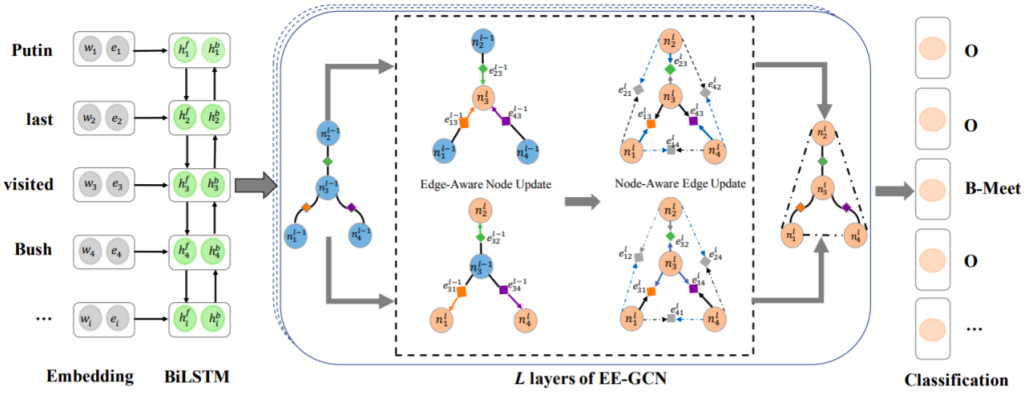

Figure 16: Event extraction by utilizing dependency relations. The dependency tree is encoded in a trainable tensor which is utilized in the edge-aware node update module to update the word representations. Afterwards, the dependency tensor is updated by the node-aware edge update module using the new word representations (Cui et al., 2020).

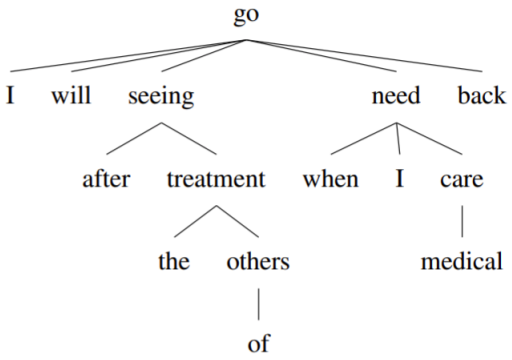

Figure 17: Dependency tree for the sentence *I will, after seeing the treatment of others, go back when I need medical care*(Pouran Ben Veyseh et al., 2019)

of $\{-3, -2, -1, +1, +2, +3\}$, where the class *-3* represent the predictions for cases that the model is confident that the event is not factual (i.e., it has not happened) and the class *+3* represent the predictions for cases in which the model is confident that the event is factual (i.e., it has happened). On the other hand, a regression model predicts any number from the range of *-3* to *+3*. The predictions that are closer to *+3* indicates that the model is more confident about the factuality of the event mention.

For event factuality, the existing dataset provide annotations for either the classification task (i.e., discrete scores from the set $\{-3, -2, -1, +1, +2, +3\}$) or regression task (i.e., continuous numbers from the ragne $[-3, +3]$). The most important existing datasets are:

- FactBank (Saurí and Pustejovsky, 2009): A classification dataset provided otop the TimeBank dataset (Pustejovsky, 2006). Factbank provide the factuality assessment with respect to different sources (e.g., author).
- MEANTIME (Mititelu et al., 2018): This dataset re-annotate a portion of the FactBank dataset to capture the factuality of the event with respect to the pragmatic context of the event.
- UW (Lee et al., 2015): This a regression-based dataset provides annotations for event factuality in the range $[-3, +3]$. It also maps the discrete labels from FactBank and MEANTIME to the same range $[-3, +3]$.
- UDS-IH1 (White et al., 2016) and UDS-IH2 (Rudinger et al., 2018): These regression-based datasets provide annotations for 6,920 and 27,282 event predicates. The datasets are annotated using crowd-sourcing (e.g., annotators from Amazon Mechanical Turk (MTurk)).

For event factuality prediction, the exisiting state-of-the-art models employ the syntactic or semantic structure of the sentence to capture the important words regarding the given event trigger. For instance, in the sentence *I will, after seeing the treatment of others, go back when I need medical care*, the factuality of the event trigger *go back* could be induced from the verb *will*. However, this cue (i.e., the word *will*) is sequentially far from the trigger word *go back*. Due to this long distance, a sequence based model will fail to effectively capture the dependency between these two words. In contrast, a structure-based model could benefit the

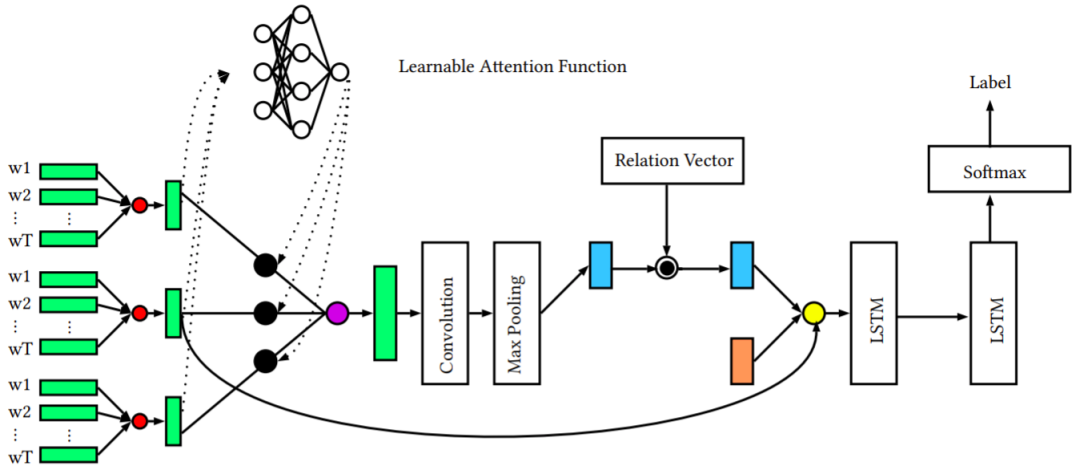

Figure 18: Deep learning for rumor stance classification (Veyseh et al., 2017)

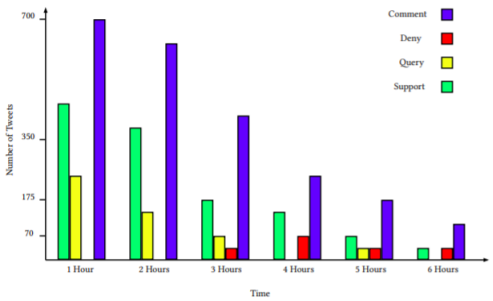

Figure 19: Distribution of Labels as time passes. The horizontal axis displays how much time has passed from the originating rumorous tweet. The vertical axis displays the number of tweets in each class (Veyseh et al., 2017).

short distance between these two words in the dependency tree to infer the importance of the word *will* for the trigger word *go back* (See Figure 17 for the dependency tree of the this sentence). To encode this information, authors in (Rudinger et al., 2018) proposed to employ the Tree-LSTM architecture. More specifically, the dependency tree of the sentence is employed as the syntax-based structure of the sentence to identify the children of each word in the sentence. Afterwards, the representation of each word is updated based on its embedding and the representation of its children obtained from the hidden states of the Tree-LSTM neurons.

In addition to the syntax-based structure, the semantic connection between words could be crucial too. For instance, in the sentence *Knight lied when he said I went to the ranch*, the semantics of the word *lied* indicates that the event mentioned after this is not factual. Capturing this semantic connection between words requires semantics-based structure modeling. To obtain the semantics-based structure, authors in (Pouran Ben Veyseh et al., 2019) proposed to employ a sequential encoder (i.e., LSTM) to represent the words. Next, a two-layer feed forward neural network predicts the weight of the pair-wise connections between every pair of words using the concatenation of their representations obtained from the sequential encoder (i.e., LSTM). Moreover, authors suggested to combine the learnt semantic structure with the syntactic tree to enhance the word representations for event factuality prediction task. To achieve this goal, they linearly combine the learnt semantics-based adjacency matrix with the dependency tree adjacency matrix. Finally, a graph convolution layer (GCN) updates the word representations using the combined syntactic and semantic structures.

7.2 Information Veracity in Social Network

With the proliferation of social media, these networks are now one of the well-known source of information for people. The high reachability of the contents in these networks makes them a potential platform for people to quickly spread news to a large audience. This characteristics combined with lack of editorial board to filter out non-factual content, make the social networks (SN) a potential platform for evil-doers to spread false information (e.g. rumors). In order to prevent this side-effect of SNs, automatic information veracity assessment tools are required. Hence, in recent years, there have been some efforts to design new models to predict the veracity of the information in social networks. These models mainly focus on the rumor resolution task. In particular, rumor resolution aims

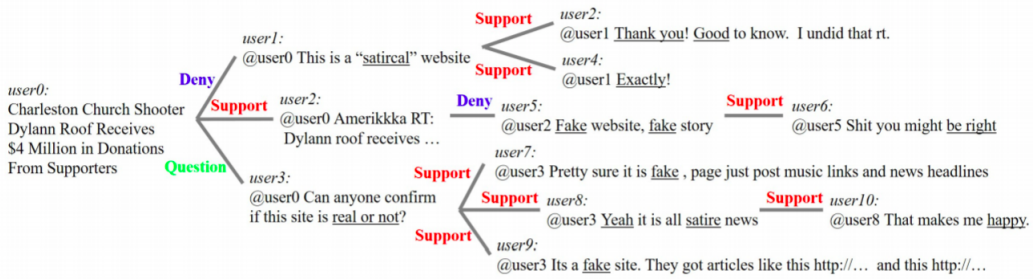

Figure 20: An example of tree structure in Twitter for rumor stance classification (Ma et al., 2018)

to categorize a claim in social network (e.g., a tweet in Twitter) into one of the following categories: (1) True Rumor: Claims that turns to be factual, (2) False Rumor: Claims that could be proved as non-factual, (3) Without Rumor: Contents that do not convey any questionable claim, and (4) Unrecognizable: Claims that could not be identified either as true rumor or false rumor.

Rumor resolution has been the subject of a series of evaluation forums named as RumorEval (Gorrell et al., 2018). This shared-task provides datasets for rumor resolution from Twitter network. Also, in addition to the main task (i.e., rumor resolution), they provide dataset for another related sub-task called rumor stance classification. The purpose of the rumor stance classification is to identify the attitude of the people toward a claim which contains a rumor. More specifically, the attitudes are categorized into four groups: (1) Support (i.e., confirming that the claim is true) (2) Deny (i.e., rejecting the claim) (3) Comment (i.e., the user does not confirm or reject the claim, instead, he/she expresses his/her view about the claim) (4) Question (i.e., inquires about the main claim). These attitudes could be expressed in the comments/replies to a post in social network. Together with the main post itself, these replies form a tree structure. An example of this tree is shown in Figure 20. The tree structure of replies could be employed by graph-based models to encode each reply, thereby, to predict the attitude of its author. This section first reviews the existing work on rumor stance classification. Afterwards, it discusses the existing work for rumor resolutions on social media.

7.3 Rumor Stance Classification

The first work that has employed machine learning models for rumor stance classification is (Qazvinian et al., 2011). They employ a feature-based model for this task. Later, deep learning models have been exploited for rumor stance classification (Kochkina et al., 2017; Ma et al., 2016). More specifically, authors in (Kochkina et al., 2017) employ a sequential encoder (i.e., LSTM) to encode each branch in the tree structure of replies (see Figure 20). The representation obtained for the branch will be further employed by a logistic regression classifier to separately predict the stance of each reply in the corresponding branch. In another work (Zubiaga et al., 2016) authors use conditional random field (CRF) to encode the sequential dependencies between the replies in a branch. It has been also shown that the temporal relation between replies could be helpful for this task (Lukasik et al., 2019, 2016). More specifically, authors in (Lukasik et al., 2019) propose to employ Gussian Process to encode the process of different reply attitudes (i.e., frequency in which the different attitudes are posted for a give main post containing the rumor). In another work, authors in (Lukasik et al., 2016) employ Hawkess Process for the same task. Despite the improvements made available by these works, they cannot infer the dynamics of the changes in people's attitude toward a topic. In other words, people's attitude tends to shift from comment and query to deny or support after a period of time. This fact is expressed in the recent work (Veyseh et al., 2017) (See Figure 19). In order to encode this characteristics, authors in (Veyseh et al., 2017) introduced an attention-based model in which the representation of the current reply is obtained by attending at the previous and next replies in the same branch of the reply tree. They also combine this representation with other network-based features obtained for the reply or its author from the social network. Figure 18 shows the diagram of this model.

7.4 Rumor Resolution

For rumor resolution which is the task of if a rumor is true or not, the existing work attempt to collectively encode all replies in the tree structure of the posts (e.g., tweets). For instance, Ma et al. (2018) exerted Tree-LSTM to collectively encode the tree structure. In their model, the representation of the main tweet will be regulated by the representations from its direct replies. The replies themselves will be also regulated by the representation of their direct replies (i.e., their children in the tree structure (See Figure 20)). While this method could be helpful to incorporate the information from all replies to identify the validity of the main claim, the major limitation of this is that it is restricted to the reply structure enforced by the social network. In order to alleviate this issue, authors in (Veyseh et al., 2019c) introduced a semantic-based structure inference component. More specifically, the representations of the replies are employed to compute the pair-wise connection between them using self-attention mechanism. Combined with a novel regularization for emphasizing the content of the main tweet, this method achieves the SOTA performance on this task.

8 Text Structure in Other NLP Applications

This section studies the application of text structure and structure-based models for other natural language processing applications. Namely, we study the application of the text structure for sentiment analysis, question answering, document classification and text summarizing.

8.1 Sentiment Analysis

Sentiment analysis is one of the well known tasks in natural language processing. The goal of this task is to identify the attitude expressed in a piece of text. For instance, in the sentence *This restaurant is famous because of its high-quality Kebabs*, the author expresses a positive attitude. Identifying the polarity of the authors' attitude could be helpful for other downstream applications including opinion mining and recommending systems. Due to the importance of this task, there is a wealth of prior work for sentiment analysis. Furthermore, in addition to the main task, recent works have extensively studied the sub-task aspect-based sentiment analysis (ABSA). In this sub-task, the goal is to identify the attitude of the author toward a specific topic. For instance, in the sentence *The Kebabs were good but the service was terrible*, the author express a positive attitude toward *Kebab* and a negative attitude toward *service*. In prior works, topics of interest in which the attitude is evaluated against are categorized into two groups:

- Aspect Term: In this category, topic is one of the words appearing in the sentence. For instance, in the given example, there are two aspect terms *Kebab* and *service*.

- Aspect Category: This group of topics refer to subjects that are discussed in the text but they might not explicitly appear in it. For instance, the aspect category *quality of food* is discussed in the aforementioned example but it is not explicitly part of the sentence[2].

It is worth noting that ABSA might be used in a pipeline system or a joint model with aspect term extraction (ATE) and aspect category extraction (ACE) modules. More specifically, in a pipeline model, the aspect terms and aspect categories are separately extracted using a pre-trained model. Then, the extracted aspect terms or categories will be used by the final ABSA system. On the other hand, in a joint model, the system simultaneously predicts the aspect term, aspect category and the attitude of the author toward them.

In addition to the well-known SA and ABSA, recently another sub-task of sentiment analysis has been introduced. This task, called targeted opinion word extraction (TOWE), aims to identify the words in the text that convey the attitude of the author, i.e., opinion word, toward the specific topic expressed in the sentence. For instance, in the sentence *The keyboards of this laptop are quite well-designed, however, its screen is disappointing*, there are two aspect terms, i.e., *keyword* and *screen*, and the opinion word for the aspect term *keyboard* is *well-designed* and the opinion word for the aspect term *screen* is *disappointing*. By identifying the opinion words toward different aspect terms, TOWE could be helpful to increase the interpretability of an ABSA system. Similar to ABSA, some recent work propose a pipeline model or a joint model for TOWE. In the pipeline model, the aspect terms are predicted by a pre-trained model. On the other hand, a joint TWOE model predicts

[2]Note that in this case the word *Kebab* is implicitly referring to the aspect category *quality of food*

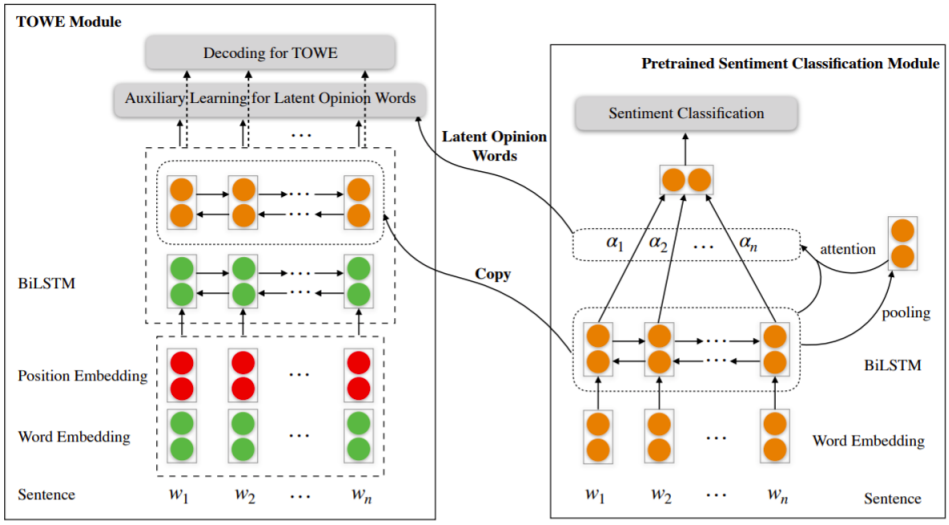

Figure 21: The architecture of Latent Opinions Transfer Network (Wu et al., 2020c)

both the aspect terms and the opinion words towards to each of them.

In this section, we study the progress of recent works on ABSA and TOWE, with an emphasis on the application of the text structure for these two tasks.

8.1.1 Aspect-based Sentiment Analysis

Prior works on aspect-based sentiment analysis range from feature-engineering methods (e.g., SVM) (Wagner et al., 2014) to the advanced deep learning models (Wagner et al., 2016; Johnson and Zhang, 2015; Tang et al., 2016). Despite the improvement obtained from the typical sequence-based deep models (e.g., LSTM) (Wagner et al., 2016) and other novel mechanism such as attention mechanism (Luong et al., 2015) and gating mechanism (He et al., 2018), recent work has shown that the graph-based models that encode the syntactic structure of the sentence achieve the state-of-the-art results for ABSA (Huang and Carley, 2019; Zhang et al., 2019a; Hou et al., 2019). This sections reviews the important recent works that employ the structure-aware models for ABSA.

Huang and Carley (2019) proposed a model based on graph attention network to encode the dependency tree of the sentence. More specifically, given the syntactic tree of the sentence, an adjacency matrix is computed. In this matrix, entries indicate whether the words in the two corresponding indices are connected in the dependency tree or not. As the dependency tree is a directed graph, the adjacency matrix is asymmetrical. Afterwards, a graph-attention network (GAT) (Veličković et al., 2017) is employed to update the initial representations of the words using the dependency tree adjacency matrix. Specifically, GAT consists of multiple layers to update node representations using attention-based neighbor aggregation. Note that the initial representation of the nodes are obtained from their corresponding pre-trained embeddings. While GAT could be more effective than Graph Convolution Network (GCN) to capture longer dependencies in the input graph, its performance still degrades by stacking too many layers of GAT. In order to address this issue, authors in (Huang and Carley, 2019) proposed to employ a Recurrent Neural Network (RNN) to encode the representations of the nodes from different layers of the GAT. In particular, all representations of the i-th node from l layers of GAT, are fed into a LSTM layer and the final hidden states of the LSTN neurons are employed as the final representation of the i-th word.

Despite the improvement obtained by the approach presented in (Huang and Carley, 2019), this method fails to capture the importance of the aspect term in the pair-wise interaction between words. In other words, the attention scores are ignorant of the given aspect term in the sentence. It could be problematic as the aspect term is the most important word in the sentence and it should be emphasized in the representations of the other words too. To alleviate this limitation, authors in (Veyseh et al., 2020d) proposed to compute aspect-aware gates to be applied to each layer of GCN. More specifically, the representation of the aspect term w_t is employed by a feed forward neural network to compute the gate vector g_t. The gate vector g_t

is multiplied to the word representations obtained from the l-th layer of the GCN operating on the dependency tree, i.e., $h'_i = g_t * h_i$. Although this gating mechanism could be helpful to incorporate the information about the aspect term to the representations of the other words, one drawback is that it exploits the same gate for different layers of the GCN that naturally represent different abstract information. To mitigate this problem, authors suggested to compute separate gate vectors for each layer of the GCN. Moreover, they introduced a diversity auxiliary loss to encourage the difference between gates of the different layers. Finally, in addition to the application of the syntactic tree in the gated graph convolution network, authors in (Veyseh et al., 2020d) proposed to employ the syntactic tree to guide the model emphasizing on the words that are syntactically more important to the aspect term. In particular, they compute importance scores for each word based on their distance to the aspect term in the dependency tree. They also compute another importance score for each word based on the similarity of its representations to the representation of the entire sentence. Finally, they encourage these two sets of scores to be similar to each other by using KL-divergence between them in the final loss function.

8.1.2 Targeted Opinion Word Extraction

Compared to the other sub-tasks of sentiment analysis, there are fewer prior works on the targeted opinion word extraction. Although some of the prior works have studied the task of opinion word extraction (OTE) (Htay and Lynn, 2013; Shamshurin, 2012), they extract the general opinion words expressed in the sentence regardless of the given target words. Another related task to TOWE is opinion target extraction (OTE) (Qiu et al., 2011; Liu et al., 2015a; Poria et al., 2016; Yin et al., 2016; Xu et al., 2018) whose goal is to identify the target words for which the author express his/her opinion in the sentence. Also, some prior works propose a joint model for opinion word extraction (OWE) and opinion target extraction (OTE). Howver, they do not pair the opinion words with their targets (Qiu et al., 2011; Liu et al., 2013a; Wang et al., 2016b, 2017; Li and Lam, 2017). The few existing works on TOWE employ either a rule-based method (Zhuang et al., 2006; Hu and Liu, 2004) or deep learning models (Fan et al., 2019; Wu et al., 2020c; Pouran Ben Veyseh et al., 2020). In the recent work by (Fan et al., 2019), authors annotated the the widely used ABSA datasets from the SemEval challenges, i.e., SemEval 2014, 2015 and 2016. These datasets contain reviews for restaurants and laptops. One limitation of the datasets prepared by (Fan et al., 2019) is their limited size compared to other exiting datasets for sentiment analysis. This limited size might hinder developing an effective deep learning model. To address this issue, authors in (Wu et al., 2020c) proposed a transfer learning based model. Specifically, they propose to learn the opinion word extraction knowledge from the existing sentiment classification datasets by pre-training an attention-based sentiment classifier. Note that the attention scores obtained for each word from the pre-trained sentiment classifier indicates the general opinion words in the input sentence. In order to transfer this information to the TOWE model, they propose two mechanism: (1) Incorporating the representation of the words from the pre-trained sentiment classifier into the TOWE model by concatenating them with the representations of the corresponding words from the TOWE model, (2) Re-scaling the attention scores obtained from the pre-trained sentiment classifier with respect to their distance to the target and use the re-scaled attention scores as auxiliary labels to be predicted by the TOWE model in a multi-task setting. The overall architecture of this model is shown in Figure 21. While this method achieves some improvement, the major limitation of this is that it totally ignores the syntactic information of the input sentence. To overcome this limitation, authors in (Pouran Ben Veyseh et al., 2020) proposed to employ the dependency tree of the sentence to infer the importance of the words of the sentence with respect to the given target. More specifically, the application of the dependency tree in this work is two-folded: (1) To induce the importance of the words and incorporate it into the model parameters, (2) Update pair-wise interactions of the words based on their connection in the dependency tree. To achieve the first goal, they compute the dependency-based importance scores of the words based on their distance to the target word in the dependency tree. In order to incorporate these scores to the model parameters, they suggest to first compute another set of scores based on the model parameters (called model-based scores). Afterwards, they minimize the difference between these two sets of scores, i.e., dependency-based and model-based scores, using KL-divergence. In or-

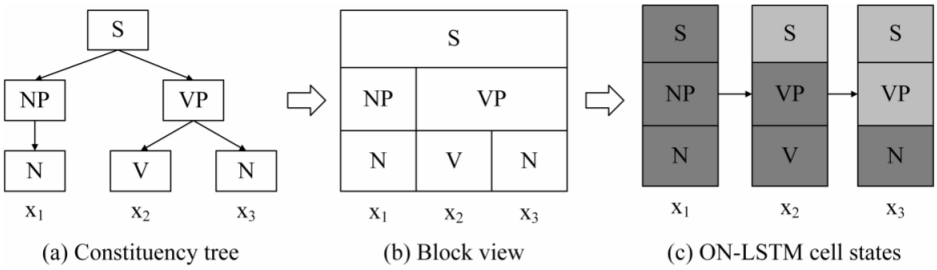

Figure 22: Correspondence between constituency parse tree and hidden states of ON-LSTM (Shen et al., 2018). Figure (a) shows the constituency tree for the sentence $S = x_1, x_2, x_3$, Figure (b) shows the block-view of the constituency tree and Figure (c) shows the hidden states of ON-LSTM at different time-steps (i.e., words). Note that in this symbolic sentence, the word x_1 has the highest importance so all hidden states of ON-LSTM are active in that time step. On the other hand, x_3 is the least important words and fewest neurons are active in the corresponding time step.

der to compute the model-based importance scores, they employ Ordered Neuron LSTM (ON-LSTM) architecture. This architecture is similar to the well-known LSTM architecture with the key difference of having two master input and forget gates. These gates control the frequency in which each neurons of ON-LSTM should be updated. More specifically, these gates are computed and employed in the ON-LSTM computations as follows:

$$\begin{aligned}
\hat{f}_i &= cummax(W_{\hat{f}} x_i + U_{\hat{f}} h_{i-1} + b_{\hat{f}}) \\
\hat{i}_i &= 1 - cummax(W_{\hat{i}} x_i + U_{\hat{i}} h_{i-1} + b_{\hat{i}}) \\
\bar{f}_i &= \hat{f}_i \circ (f_i \hat{i}_i + 1 - \hat{i}_i), \bar{i}_i = \hat{i}_i \circ (i_i \hat{f}_i + 1 - \hat{f}_i) \\
c_i &= \bar{f}_i \circ c_{i-1} + \bar{i}_i \circ \hat{c}_i
\end{aligned} \tag{1}$$

where h_{i-1} is the representation of the $i-i$-th word, $W_{\hat{f}}$, $U_{\hat{f}}$ and $b_{\hat{f}}$ are model parameters, f_i, i_i and c_{i-1} are the forget and input gates and the context vector for the i-th and $i-1$-th word, respectively, $\circ$ is element-wise multiplication and finally, $cummax$ is a new activation function defined as follows:

$$cumax(x) = cumsum(softmax(x)) \tag{2}$$

and $cumsum$ is defined as $cumsum(u_1, u_2, \ldots, u_n) = (u'_1, u'_2, \ldots, u'_n)$ where $u'_i = \sum_{j=1..i} u_j$

Note that ideally $cummax$ divides its input into two sections of 0's and 1's. Using this function as the activation function of the master forget or input gates, it controls how many neurons should be activated at the corresponding time step. Based on this, authors in (Shen et al., 2018) and (Pouran Ben Veyseh et al., 2020) suggest to infer the importance of the word w by $s_w = 1 - \sum_{j=1..D} \hat{f}_{ij}$. Note that although Shen et al. (2018) directly use this importance score to infer the constituency tree of the sentence (See Figure 22), authors in (Pouran Ben Veyseh et al., 2020) employ these scores to measure the difference between model-based and dependency-based importance scores via KL-divergence loss.

In order to realize the second aforementioned application of dependency tree (i.e., encoding the pair-wise interaction between words), authors in (Pouran Ben Veyseh et al., 2020) employ graph convolution network (Kipf and Welling, 2017). Since the pair-wise interaction between words in the dependency tree might be ignorant of the target word, they suggest to combine the dependency tree adjacency matrix with another adjacency matrix induced by a feed forward layer consuming the distance of the words to the target in the dependency tree.

8.2 Definition Extraction

Automatically extracting terms and definitions from text is one of the natural language processing tasks related to information extraction. This task aims to identify the symbols or phrases, i.e., terms, for which a definition is provided in text. For instance, in the sentence *The phrase "atoms and molecules" is explained in the dictionary by the expression of building blocks of materials* the definition *building blocks of materials* is provided for the term *atoms and molecules*. Identifying the terms and definitions in text could be helpful for question answering, knowledge base population, and text summarization.

Prior work in Definition Extraction (DE) takes two step to fulfill this task: (1) Identifying the sentences in which a term or a definition is provided,

Relation Name	Description
Direct-defines	Links definition to term.
Indirect-defines	Links definition to referential term or term to referential definition.
Refers-to	Links referential term to term or referential definition to definition.
AKA	Links alias term to term.
Supplements	Links secondary definition to definition.

Figure 23: Relation schema in DEFT dataset (Spala et al., 2019)

Dataset	# of positive annotations	Size (in sentences)
WCL	1,871	4,718
W00	731	2,185
DEFT	**11,004**	**23,746**

Figure 24: Statistics of existing definition extraction datasets (Spala et al., 2019)

(2) Span extraction whose goal is to recognize the spans in text corresponding to term or definition. While early models provide a solution for the former task (Klavans and Muresan, 2001; Cui et al., 2004, 2005; Fahmi and Bouma, 2006), the latter gained more attention in recent years (Li et al., 2016b; Veyseh et al., 2020a; Kang et al., 2020).

For this task, there are three major datasets:

- WCL: This dataset, contributed by (Navigli and Velardi, 2010), annotates Wikipedia articles with the terms and definitions in general domain.
- W00: This dataset is introduced by (Jin et al., 2013). It provides annotations for terms and definitions in scientific domain from the papers in ACL-ARC anthology.
- DEFT: This is the largest available definition extraction dataset (Spala et al., 2019). It provides annotations for terms and definitions in legal documents, i.e., contracts, and scientific documents, i.e., text books. Moreover, it also annotates the different type of relations between terms or definitions in the document. The description of the relation schema is provided in Figure 23. Also, Figure 24 shows the statistics for all three datasets WCL, W00 and DEFT.

Prior works on DE extends from rule-based methods (Klavans and Muresan, 2001; Cui et al., 2004, 2005; Fahmi and Bouma, 2006) to feature-based (Jin et al., 2013; Westerhout, 2009) and recently advanced deep learning models (Anke and Schockaert, 2018; Veyseh et al., 2020a; Kang et al., 2020). Although recent sequential deep learning models such as LSTM and CNN achieve promising results on this task (Anke and Schockaert, 2018), the state-of-the-art results are obtained by models employing the structure of the input text. More specifically, authors in (Veyseh et al., 2020a) proposed a model in which the dependency tree of the sentence is utilized by graph convolution network (GCN) to encode long-range dependencies between words in the sentence. Moreover, in their model they encourage the consistency between the term and the definition by ensuring the similarity between the term and the definition representations. This work is further analyzed and improved in the recent work (Kang et al., 2020) by utilizing a transformer-based encoder to model the syntactic structure of the sentence.

A special case for definition extraction is acronym meaning extraction. Acronyms and abbreviations are shorter forms of technical terms and they are prevalent in scientific and legal writing. Acronym meaning extraction consist of two major sub-tasks:

- Acronym Identification: In this sub-task the goal is to identify the spans that represent an acronym or phrases which are abbreviated in text. For instance, in the sentence *The main key performance indicator, herein referred to as KPI, is the E2E throughput*, there are two acronyms, i.e., *KPI* and *E2E*, and one phrase with an abbreviated form, i.e., *key performance indicator*. An acronym identification systems (AI) aims to identify both acronyms and the phrases in text. This task is normally formulated as sequence labeling. The predominant approach used in prior work on AI is based on heuristics rules. For instance, the ap-

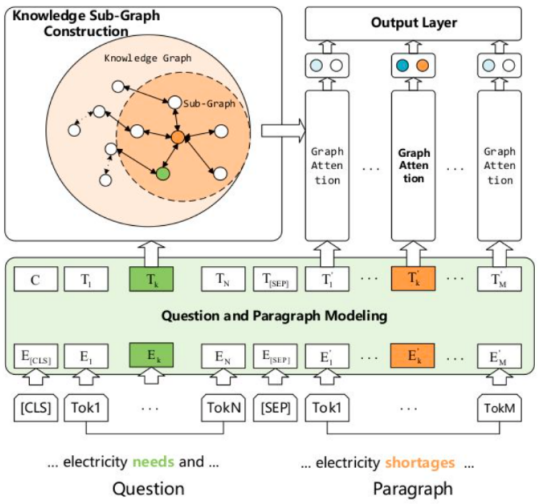

Figure 25: Incorporating external knowledge into transformer-based machine reading comprehension (Qiu et al., 2019)

proach proposed in (Charbonnier and Wartena, 2018) identities the acronyms if 60% of their characters are upper-cased. To identify the phrases they propose to compare the initials of the words surrounding an acronym with the letters of the acronym itself; phrases that their initials could form the acronym are labeled as abbreviated phrases. Despite the simplicity of the rule-based methods, they achieve promising results on various acronym identification datasets (Veyseh et al., 2020c). Recently, authors in (Veyseh et al., 2020c) proposed a deep learning model which utilizes a sequence-based encoder (i.e., BiLSTM) followed by conditional random field (CRF) layer to predict the acronyms and their long-forms in text.

- Acronym Disambiguation: Acronyms and abbreviations might have multiple expanded forms. For instance, *PDF* could refer to *Probability Density Function* or *Portable Document Format*. The correct meaning of an acronym depends on the context in which the acronym appears. As such, acronym disambiguation (AD), aims for identifying the correct meaning of an ambiguous acronym, i.e., an acronym with multiple long-forms. For this task, prior work employs both feature-based models (Li et al., 2018) and deep learning approaches (Charbonnier and Wartena, 2018; Ciosici et al., 2019). For example, authors in (Charbonnier and Wartena, 2018) propose to pre-train a language model in which the acronyms are represented with the special token *[Acronym]_[Meaning]*. Note that for each meaning of an acronym, one special token is created. Afterwards, in inference time, the acronym is replaced with blank and the special token with the highest probability to fill the blank is predicted by a language model. The meaning corresponding to the predicted special token is selected as the expanded form of the given acronym.

Employing the structure of the text has been shown to be effective for acronym extraction too (Veyseh et al., 2020c). More specifically, authors in (Veyseh et al., 2020c) propose to employ the dependency tree of the input sentence to capture the interactions between words of the sentence. This interaction is encoded by graph convolution network (GCN). They show that this model could significantly improve the performance over a sequence-based model. For instance, consider the sentence *Words that are not compatible with our pre-defined rules are excluded from SDP*, with the acronym *SDP* and the correct expanded form *Shortest Dependency Path*. In this example, the clue for identifying the correct meaning of the acronym is *Words* at the beginning of the sentence. As there is a long distance between *SDP* and *Words* a sequence-based model might fail to capture their dependencies. On the other hand, these two words are close to each other in the dependency tree. Thereby, incorporating the information obtained by dependency tree into the model could improve the performance of acronym disambiguation. Despite this improvement, authors in (Veyseh et al., 2020c) warn that

the direct incorporation of the dependency tree might also involve some noisy dependencies into the model which results in performance degradation compared to a sequence-based acronym disambiguation model. Hence, enriching a structure-based model with some mechanism to control the contribution of the dependency tree and filter out the noisy dependencies seems to be promising direction for future improvement on this task. This can be achieved by graph attention network or more sophisticated structure-based models for inferring the semantic structure instead of relying on the syntactic trees.

8.2.1 Question Answering and Machine Reading Comprehension

One of the well-known tasks in natural language processing is question answering. This task backs to 1960s (Green Jr et al., 1961) when early systems were designed to extract answers to questions from a database. Since then, several formulations and settings for question answering has been proposed including open domain (Yang et al., 2015), knowledge base (Veyseh, 2016), or community-based (Zhao et al., 2017) question answering. Some other tasks such as machine reading comprehension (Qiu et al., 2019), relation extraction (Li et al., 2019c), or event extraction (Du and Cardie, 2020) has been also modeled and approached using question answering paradigm. For a complete survey on question answering, refer to (Gupta and Gupta, 2012; Bouziane et al., 2015; Fu et al., 2020).

In this study, we review the application of structure-aware models for question answering (QA). Some of the prior work employs the structure in a knowledge base (KB) to extract answers to the questions. For instance, authors in (Qiu et al., 2019) proposed to incorporate the external knowledge encoded in a Knowledge base into a transformer-based QA system. More specifically, for a given paragraph from which the answer is expected to be extracted and a question, they first extract the triples in the knowledge base whose tail or head share the same lemma with one of the words in the given paragraph. Afterwards, for each extracted triple, the neighbors of the heads or tails that share the same lemma with one of the words in the questions are selected too. Finally, using the extracted triples and their neighbors, author create a graph which represent a sub-graph of the KB. This graph is later encoded by a graph attention network (GAT). The representations of the words obtained from the GAT will be finally concatenated with the representations of them from the transformer to be used in a span labeling model. Figure 25 shows the diagram of this model.

Knowledge bases could be also directly used to extract answers from them. For instance, authors in (Veyseh, 2016) proposed a feature-based model to extract answers from DBpedia which is a knowledge base constructed from Wikipedia. In particular, they first extract the keywords of the questions that would be further exploited to extract entities, i.e., candidate answers, from the knowledge base. Afterwards, using the semantic similarity between the entity relations and the question, the triple with the highest semantic similarity is selected as the answer to the question.

In addition to the application of the knowledge base for question answering, some recent works employ the structure-aware models for document-level question answering. For instance, authors in (De Cao et al., 2018) proposed a graph-based model to extract answers from a collection of documents. More specifically, they first create a graph of entities and semantic relations between them from a set of documents. The representations of the entities is obtained from the sequential encoding of the words of each document. Next, to perform reasoning across multiple documents and update the entity representations, they employ graph convolution network (GCN). Finally, the representations of the question obtained from a sequence-based encoder is concatenated with the representations of the entity nodes to predict the answer.

8.2.2 Text Summarization

Text summarization is one of the established task in natural language processing dating back to 1950s (El-Kassas et al., 2020). The goal of this task is to summarize a long piece of text, e.g., a document, into a shorter version. To this end, prior work takes three different approaches: (1) Extractive summary: In this approach the summary is constructed by selecting the salient sentences or other text blocks from the original document (Murray et al., 2005), (2) Abstractive summary: This method aims to generate a summary consisting of sentences which might not be explicitly in the document (Gehrmann et al., 2018), and finally (3) Hybrid summary: In this method both extractive and abstractive techniques are utilized to obtain the final summary (Kirmani et al., 2019). For a comprehensive review of the prior work on text

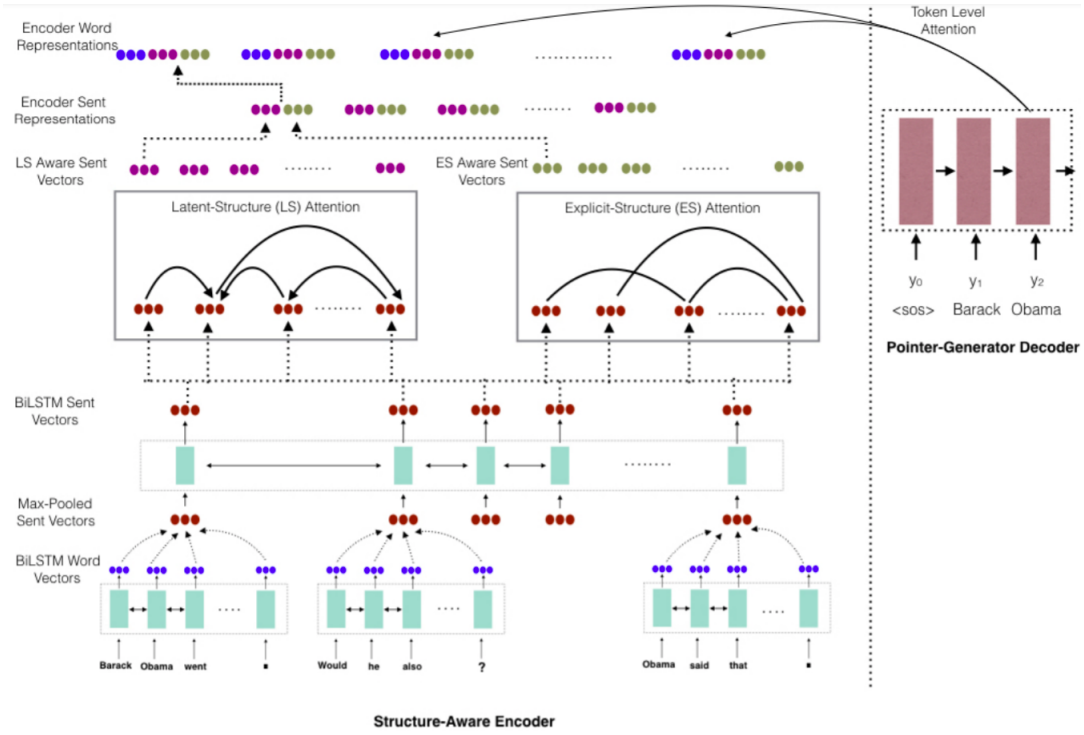

Figure 26: Document summarization using latent and explicit structure encoding (Balachandran et al., 2020)

summarization, we refer the reader to the recent survey (El-Kassas et al., 2020). In this study we review some of the recent works that utilize the text structure to obtain the summary.

For text summarization (TS), traditionally graph-based architectures have played an important role to encode the interaction between different parts of the document, thereby, improving the representation of the document for summarization. These methods includes sentence-sentence compatibility (Erkan and Radev, 2004), abstract meaning representation (AMR) (Liu et al., 2018) or discourse based methods such as coreference graphs (Durrett et al., 2016). In addition to these traditional structures employed for TS, recently the structure induction has gained attention too. Structure induction aims to employ the semantic representation of the words/sentences of the document to infer pair-wise interactions between them. For instance, in the recent work by Balachandran et al. (2020), authors proposed an attention-based model to incorporate the semantics of the sentences into the syntax-based structure of the document. In particular, the sentences of the document are first encoded by a sequence-based model (i.e., BiLSTM). Afterwards, the max-pooled representations of these sentences are fed into two graph-based models. The first one is a graph convolution network which utilizes the coreference-based graph of the document to update the sentence representations. The second one is a self-attention component which assesses the pair-wise interaction of the sentences using their representations. The concatenation of the outputs of these two graph-based networks are finally fed into a decoder as the context representation to generate the text summary. The diagram of this model is shown in Figure 26

9 Future Works

Employing structural modeling is an important topic in various NLP tasks, especially information extraction whose goal is to create a structured knowledge from unstructured text. Despite all successes in leveraging syntactic or semantic structure of the text, external structures such as knowledge bases and innovative structure-based modeling (e.g., graph attention network), there are a lot of challenges remained for future research. One of the major limitations of the existing work is that they are limited to the existing structures (e.g., syntactic tree or knowledge bases) extracted using external tools. More specifically, a pre-trained model is required to create the structure used in the IE model. This requirement has two drawbacks: (1) In domains and settings in which an efficient structure could not be extracted using external tools (e.g.,

in cross-lingual setting that one of the languages lacks efficient syntactic parser) the existing models fail to decently work. (2) The external tools are pre-trained for the general task (e.g., constituency parsing), thus ignorant of the downstream task (e.g., relation extraction). This mismatch between the pre-trained model's task and the IE task might result in inefficiency of the extracted structure using these tools. In order to address these two limitations, one direction is to simultaneously train the IE model for the task in hand and also to infer the structure in a multi-task setting. To achieve this goal, several questions should be answered such as whether a sparse graph is suitable for the IE task or a dense graph; what elements should be used as the nodes and the edges of the inferred graph (i.e., words, entities, etc); and how the inferred structure should be involved in the model for the IE task?

Another limitation of the existing work is that they are mainly restricted to the sentence-level structures (e.g., dependency tree). The main goal of IE is to extract formations from document rather than one sentence. So it is crucial for the future research to explore the challenges of leveraging a document-level structure. Although there are some recent work for document level RE or EE, however these models mainly exploit heuristics to create the structure. For instance, they use the sentences or entities as the nodes and connect them in the graph if the entity is mentioned in the sentence. Such simple rules might not be able to capture all types of interactions between different parts of the document. Thereby, exploring efficient ways to encode the structure of the document for IE is another direction for future work.

Finally, exploiting external knowledge in an structured model is also another possible direction for future research in IE. Using external knowledge (e.g., knowledge base) for IE has a long history. However, incorporating these structures in the modern deep architectures (e.g., transformers like BERT) is not fully explored yet. It has been shown that the transformer based model pre-trained on large corpora are able to encode notions of textual structure. However, their capability to encode the factual knowledge in a knowledge base should be investigated in future.

10 Proposed Research Topic

Based on the expected future works elaborated in the previous section, I propose to work on the application of the document structure for multilingual event detection. In this task, the goal is to study how the structural information induced for a given document could be useful for detecting the event triggers and their arguments in text. This is a novel study as none of the prior works have considered the document structure for event extraction specifically for multilingual setting. Also, given the recent work on structure induction, this work is expected to have a substantial contribution to the field by providing more insight into the applicability of a deep learning model to induce a structure which is useful across multiple languages. To this end, due to the lack of existing resources, i.e., a dataset, we should first attempt to collect training data for document level multilingual event extraction. In addition, we expect to have some analysis on how the existing deep architectures, such as transformer-based pre-trained language models, could perform in document level event extraction. Finally, we will work on a novel model to efficiently infer the structure of a given document and to exert this in the final event extraction model. The proposed model will be evaluated on multiple languages including English, Spanish, Persian and Vietnamese.

References

2016. Rich ere annotation guidelines overview. In *Linguistic Data Consortium,Philadelphia*.

Gustavo Aguilar and Thamar Solorio. 2019. Dependency-aware named entity recognition with relative and global attentions. *arXiv preprint arXiv:1909.05166*.

Wasi Uddin Ahmad, Nanyun Peng, and Kai-Wei Chang. 2020. Gate: Graph attention transformer encoder for cross-lingual relation and event extraction. *arXiv preprint arXiv:2010.03009*.

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

Thien Huu Nguye Amir Pouran Ben Veyseh, Tuan Ngo Nguyen. 2020. Graph transformer networks with syntactic and semantic structuresfor event argument extraction.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Neural module networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 39–48.

Luis Espinosa Anke and Steven Schockaert. 2018. Syntactically aware neural architectures for definition extraction. In *NAACL-HLT*.

Vidhisha Balachandran, Artidoro Pagnoni, Jay Yoon Lee, Dheeraj Rajagopal, Jaime Carbonell, and Yulia Tsvetkov. 2020. Structsum: Incorporating latent and explicit sentence dependencies for single document summarization. *arXiv preprint arXiv:2003.00576*.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R Devon Hjelm. 2018. Mine: mutual information neural estimation. *arXiv preprint arXiv:1801.04062*.

Anders Björkelund and Jonas Kuhn. 2014. Learning structured perceptrons for coreference resolution with latent antecedents and non-local features. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 47–57.

Jari Björne and Tapio Salakoski. 2018. Biomedical event extraction using convolutional neural networks and dependency parsing. In *Proceedings of the BioNLP 2018 workshop*, pages 98–108, Melbourne, Australia. Association for Computational Linguistics.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795.

Abdelghani Bouziane, Djelloul Bouchiha, Noureddine Doumi, and Mimoun Malki. 2015. Question answering systems: survey and trends. *Procedia Computer Science*, 73:366–375.

Razvan Bunescu and Raymond Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 724–731.

Yixin Cao, Lei Hou, Juanzi Li, and Zhiyuan Liu. 2018. Neural collective entity linking. *arXiv preprint arXiv:1811.08603*.

Yee S. Chan and Dan Roth. 2010. Exploiting background knowledge for relation extraction. In *COLING*.

Jean Charbonnier and Christian Wartena. 2018. Using word embeddings for unsupervised acronym disambiguation. In *Proceedings of the 27th International Conference on Computational Linguistics*, Santa Fe, New Mexico, USA. 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 *ACL-IJCNLP*.

Zheng Chen and Heng Ji. 2009. Language specific issue and feature exploration in chinese event extraction. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 209–212.

Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. 2019. Connecting the dots: Document-level neural relation extraction with edge-oriented graphs. In *EMNLP-IJCNLP*.

Manuel R Ciosici, Tobias Sommer, and Ira Assent. 2019. Unsupervised abbreviation disambiguation. *arXiv preprint arXiv:1904.00929*.

Kevin Clark and Christopher D Manning. 2016. Improving coreference resolution by learning entity-level distributed representations. *arXiv preprint arXiv:1606.01323*.

Michael Collins and Yoram Singer. 1999. Unsupervised models for named entity classification. In *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*.

Hang Cui, Min-Yen Kan, and Tat-Seng Chua. 2004. Unsupervised learning of soft patterns for generating definitions from online news. In *WWW*.

Hang Cui, Min-Yen Kan, and Tat-Seng Chua. 2005. Generic soft pattern models for definitional question answering. In *SIGIR*. ACM.

Shiyao Cui, Bowen Yu, Tingwen Liu, Zhenyu Zhang, Xuebin Wang, and Jinqiao Shi. 2020. Event detection with relation-aware graph convolutional neural networks.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2018. Question answering by reasoning across documents with graph convolutional networks. *arXiv preprint arXiv:1808.09920*.

Franck Dernoncourt, Ji Young Lee, and Peter Szolovits. 2017. Neuroner: an easy-to-use program for named-entity recognition based on neural networks. *arXiv preprint arXiv:1705.05487*.

George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie M Strassel, and Ralph M Weischedel. 2004. The automatic content extraction (ace) program-tasks, data, and evaluation. In *Lrec*, volume 2, pages 837–840. Lisbon.

Xinya Du and Claire Cardie. 2020. Event extraction by answering (almost) natural questions. *arXiv preprint arXiv:2004.13625*.

Greg Durrett, Taylor Berg-Kirkpatrick, and Dan Klein. 2016. Learning-based single-document summarization with compression and anaphoricity constraints. *arXiv preprint arXiv:1603.08887*.

Seth Ebner, Patrick Xia, Ryan Culkin, Kyle Rawlins, and Benjamin Van Durme. 2019. Multi-sentence argument linking. *arXiv preprint arXiv:1911.03766*.

Tome Eftimov, Barbara Koroušić Seljak, and Peter Korošec. 2017. A rule-based named-entity recognition method for knowledge extraction of evidence-based dietary recommendations. *PloS one*, 12(6):e0179488.

Wafaa S El-Kassas, Cherif R Salama, Ahmed A Rafea, and Hoda K Mohamed. 2020. Automatic text summarization: A comprehensive survey. *Expert Systems with Applications*, page 113679.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Ismail Fahmi and Gosse Bouma. 2006. Learning to identify definitions using syntactic features. In *Proceedings of the Workshop on Learning Structured Information in Natural Language Applications*.

Zhifang Fan, Zhen Wu, Xinyu Dai, Shujian Huang, and Jiajun Chen. 2019. Target-oriented opinion words extraction with target-fused neural sequence labeling. In *NAACL-HLT*.

Kong Fang and Fu Jian. 2019. Incorporating structural information for better coreference resolution. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 5039–5045. AAAI Press.

Zheng Fang, Yanan Cao, Qian Li, Dongjie Zhang, Zhenyu Zhang, and Yanbing Liu. 2019. Joint entity linking with deep reinforcement learning. In *The World Wide Web Conference*, pages 438–447.

Zheng Fang, Yanan Cao, Ren Li, Zhenyu Zhang, Yanbing Liu, and Shi Wang. 2020. High quality candidate generation and sequential graph attention network for entity linking. In *Proceedings of The Web Conference 2020*, pages 640–650.

Hongliang Fei, Xu Li, Dingcheng Li, and Ping Li. 2019. End-to-end deep reinforcement learning based coreference resolution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 660–665.

Jun Feng, Minlie Huang, Li Zhao, Yang Yang, and Xiaoyan Zhu. 2018. Reinforcement learning for relation classification from noisy data. *Proceedings of the Thirty-Second AAAI Con- ference on Artificial Intelligence*.

Bin Fu, Yunqi Qiu, Chengguang Tang, Yang Li, Haiyang Yu, and Jian Sun. 2020. A survey on complex question answering over knowledge base: Recent advances and challenges. *arXiv preprint arXiv:2007.13069*.

Tsu-Jui Fu, Peng-Hsuan Li, and Wei-Yun Ma. 2019. Graphrel: Modeling text as relational graphs for joint entity and relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1409–1418.

Octavian-Eugen Ganea and Thomas Hofmann. 2017. Deep joint entity disambiguation with local neural attention. *arXiv preprint arXiv:1704.04920*.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for nlg micro-planning. In *55th annual meeting of the Association for Computational Linguistics (ACL)*.

Sebastian Gehrmann, Yuntian Deng, and Alexander M Rush. 2018. Bottom-up abstractive summarization. *arXiv preprint arXiv:1808.10792*.

Genevieve Gorrell, Kalina Bontcheva, Leon Derczynski, Elena Kochkina, Maria Liakata, and Arkaitz Zubiaga. 2018. Rumoureval 2019: Determining rumour veracity and support for rumours. *arXiv preprint arXiv:1809.06683*.

Bert F Green Jr, Alice K Wolf, Carol Chomsky, and Kenneth Laughery. 1961. Baseball: an automatic question-answerer. In *Papers presented at the May 9-11, 1961, western joint IRE-AIEE-ACM computer conference*, pages 219–224.

Poonam Gupta and Vishal Gupta. 2012. A survey of text question answering techniques. *International Journal of Computer Applications*, 53(4).

Ben Hachey, Will Radford, Joel Nothman, Matthew Honnibal, and James R Curran. 2013. Evaluating entity linking with wikipedia. *Artificial intelligence*, 194:130–150.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2018. Effective attention modeling for aspect-level sentiment classification. In *ACL*.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid O Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2009. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals.

Amir Pouran Ben Veyseh Thien Huu Nguyen Hieu Man Duc Trong, Duc Trong Le. 2020. Introducing a new dataset for event detection in cybersecurity texts. In *EMNLP*.

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

Xiaochen Hou, Jing Huang, Guangtao Wang, Kevin Huang, Xiaodong He, and Bowen Zhou. 2019. Selective attention based graph convolutional networks for aspect-level sentiment classification. In *arXiv*.

Su Su Htay and Khin Thidar Lynn. 2013. Extracting product features and opinion words using pattern knowledge in customer reviews. *The Scientific World Journal*, 2013.

Linmei Hu, Luhao Zhang, Chuan Shi, Liqiang Nie, Weili Guan, and Cheng Yang. 2019. Improving distantly-supervised relation extraction with joint label embedding. 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 3812–3820.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177.

Binxuan Huang and Kathleen Carley. 2019. Syntax-aware aspect level sentiment classification with graph attention networks. In *EMNLP*.

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

Heng Ji and Ralph Grishman. 2011. Knowledge base population: Successful approaches and challenges. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pages 1148–1158.

Zhanming Jie and Wei Lu. 2019. Dependency-guided lstm-crf for named entity recognition. *arXiv preprint arXiv:1909.10148*.

Yiping Jin, Min-Yen Kan, Jun-Ping Ng, and Xiangnan He. 2013. Mining scientific terms and their definitions: A study of the acl anthology. In *EMNLP*.

Rie Johnson and Tong Zhang. 2015. Semi-supervised convolutional neural networks for text categorization via region embedding. In *NIPS*.

Dongyeop Kang, Andrew Head, Risham Sidhu, Kyle Lo, Daniel S Weld, and Marti A Hearst. 2020. Document-level definition detection in scholarly documents: Existing models, error analyses, and future directions. *arXiv preprint arXiv:2010.05129*.

Sammy Khalife and Michalis Vazirgiannis. 2018. Scalable graph-based individual named entity identification. *arXiv preprint arXiv:1811.10547*.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

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

Mahira Kirmani, Nida Manzoor Hakak, Mudasir Mohd, and Mohsin Mohd. 2019. Hybrid text summarization: a survey. In *Soft Computing: Theories and Applications*, pages 63–73. Springer.

Judith L Klavans and Smaranda Muresan. 2001. Evaluation of the definder system for fully automatic glossary construction. In *Proceedings of the AMIA Symposium*, page 324. American Medical Informatics Association.

Elena Kochkina, Maria Liakata, and Isabelle Augenstein. 2017. Turing at semeval-2017 task 8: Sequential approach to rumour stance classification with branch-lstm. *arXiv preprint arXiv:1704.07221*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Shalom Lappin and Herbert J Leass. 1994. An algorithm for pronominal anaphora resolution. *Computational linguistics*, 20(4):535–561.

Phong Le and Ivan Titov. 2019. Distant learning for entity linking with automatic noise detection. *arXiv preprint arXiv:1905.07189*.

Kenton Lee, Yoav Artzi, Yejin Choi, and Luke Zettlemoyer. 2015. Event detection and factuality assessment with non-expert supervision. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1643–1648.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. *arXiv preprint arXiv:1707.07045*.

Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. *arXiv preprint arXiv:1804.05392*.

Diya Li, Lifu Huang, Heng Ji, and Jiawei Han. 2019a. 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.

Jiao Li, Yueping Sun, Robin J Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Thomas C Wiegers, and Zhiyong Lu. 2016a. Biocreative v cdr task corpus: a resource for chemical disease relation extraction. *Database*, 2016.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2020a. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*.

Manling Li, Alireza Zareian, Qi Zeng, Spencer Whitehead, Di Lu, Heng Ji, and Shih-Fu Chang. 2020b. Cross-media structured common space for multimedia event extraction. *arXiv preprint arXiv:2005.02472*.

Manling Li, Qi Zeng, Ying Lin, Kyunghyun Cho, Heng Ji, Jonathan May, Nathanael Chambers, and Clare Voss. 2020c. Connecting the dots: Event graph schema induction with path language modeling. In *Proc. The 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP2020)*.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *ACL*.

SiLiang Li, Bin Xu, and Tong Lee Chung. 2016b. Definition extraction with lstm recurrent neural networks. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*.

Wei Li, Dezhi Cheng, Lei He, Yuanzhuo Wang, and Xiaolong Jin. 2019b. Joint event extraction based on hierarchical event schemas from framenet. *IEEE Access*, 7:25001–25015.

Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. 2019c. Entity-relation extraction as multi-turn question answering. *arXiv preprint arXiv:1905.05529*.

Xin Li and Wai Lam. 2017. Deep multi-task learning for aspect term extraction with memory interaction. In *EMNLP*.

Yang Li, Bo Zhao, Ariel Fuxman, and Fangbo Tao. 2018. Guess me if you can: Acronym disambiguation for enterprises. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1308–1317, Melbourne, Australia. Association for Computational Linguistics.

Shasha Liao and Ralph Grishman. 2010a. Filtered ranking for bootstrapping in event extraction. In *COLING*.

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

Thomas Lin, Oren Etzioni, et al. 2012. Entity linking at web scale. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*, pages 84–88.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2124–2133.

Fei Liu, Jeffrey Flanigan, Sam Thomson, Norman Sadeh, and Noah A Smith. 2018. Toward abstractive summarization using semantic representations. *arXiv preprint arXiv:1805.10399*.

Kang Liu, Heng Li Xu, Yang Liu, and Jun Zhao. 2013a. Opinion target extraction using partially-supervised word alignment model. In *Twenty-Third International Joint Conference on Artificial Intelligence*.

Pengfei Liu, Shafiq Joty, and Helen Meng. 2015a. Fine-grained opinion mining with recurrent neural networks and word embeddings. In *EMNLP*.

Tianyu Liu, Kexiang Wang, Baobao Chang, and Zhifang Sui. 2017. A soft-label method for noise-tolerant distantly supervised relation extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1790–1795.

Xiaohua Liu, Yitong Li, Haocheng Wu, Ming Zhou, Furu Wei, and Yi Lu. 2013b. Entity linking for tweets. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1304–1311.

Yang Liu, Furu Wei, Sujian Li, Heng Ji, Ming Zhou, and Houfeng Wang. 2015b. A dependency-based neural network for relation classification. In *ACL*.

Michal Lukasik, Kalina Bontcheva, Trevor Cohn, Arkaitz Zubiaga, Maria Liakata, and Rob Procter. 2019. Gaussian processes for rumour stance classification in social media. *ACM Transactions on Information Systems (TOIS)*, 37(2):1–24.

Michal Lukasik, PK Srijith, Duy Vu, Kalina Bontcheva, Arkaitz Zubiaga, and Trevor Cohn. 2016. Hawkes processes for continuous time sequence classification: an application to rumour stance classification in twitter. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 393–398.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *EMNLP*.

Chunpeng Ma, Akihiro Tamura, Masao Utiyama, Eiichiro Sumita, and Tiejun Zhao. 2019. Improving neural machine translation with neural syntactic distance. 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 2032–2037.

Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J Jansen, Kam-Fai Wong, and Meeyoung Cha. 2016. Detecting rumors from microblogs with recurrent neural networks.

Jing Ma, Wei Gao, and Kam-Fai Wong. 2018. Rumor detection on twitter with tree-structured recursive neural networks. 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.

Inderjeet Mani, Eric Bloedorn, and Barbara Gates. 1998. Using cohesion and coherence models for text summarization. In *Intelligent Text Summarization Symposium*, pages 69–76.

David McClosky, Mihai Surdeanu, and Christopher Manning. 2011. Event extraction as dependency parsing. In *BioNLP Shared Task Workshop*.

Andrei Mikheev, Marc Moens, and Claire Grover. 1999. Named entity recognition without gazetteers. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*.

Bonan Min, Shuming Shi, Ralph Grishman, and Chin-Yew Lin. 2012. Ensemble semantics for large-scale unsupervised relation extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1027–1037.

Mike Mintz, Steven Bills, Rion Snow, and Dan 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.

Verginica Barbu Mititelu, Dan Tufiş, and Elena Irimia. 2018. The reference corpus of the contemporary romanian language (corola). In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using lstms on sequences and tree structures. In *ACL*.

Makoto Miwa, Paul Thompson, Ioannis Korkontzelos, and Sophia Ananiadou. 2014. Comparable study of event extraction in newswire and biomedical domains. In *COLING*.

Gabriel Murray, Steve Renals, and Jean Carletta. 2005. Extractive summarization of meeting recordings.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

Aakanksha Naik and Carolyn Rosé. 2020. Towards open domain event trigger identification using adversarial domain adaptation. *arXiv preprint arXiv:2005.11355*.

Guoshun Nan, Zhijiang Guo, Ivan Sekulic, and Wei Lu. 2020. Reasoning with latent structure refinement for document-level relation extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1546–1557, Online. Association for Computational Linguistics.

Roberto Navigli and Paola Velardi. 2010. Learning word-class lattices for definition and hypernym extraction. In *ACL*.

Claire Nédellec, Robert Bossy, Jin-Dong Kim, Jung-Jae Kim, Tomoko Ohta, Sampo Pyysalo, and Pierre Zweigenbaum. 2013. Overview of bionlp shared task 2013. In *Proceedings of the BioNLP shared task 2013 workshop*, pages 1–7.

Vincent Ng and Claire Cardie. 2002. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 104–111.

Minh Nguyen and Thien Huu Nguyen. 2018b. Who is killed by police: Introducing supervised attention for hierarchical lstms. In *Proceedings of COLING*.

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.

Thien Huu Nguyen and Ralph Grishman. 2015a. Combining neural networks and log-linear models to improve relation extraction. *arXiv preprint arXiv:1511.05926*.

Thien Huu Nguyen and Ralph Grishman. 2015b. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 39–48.

Thien Huu Nguyen and Ralph Grishman. 2015a. Relation extraction: Perspective from convolutional neural networks. In *The NAACL Workshop on Vector Space Modeling for NLP (VSM)*.

Thien Huu Nguyen, Avirup Sil, Georgiana Dinu, and Radu Florian. 2016b. Toward mention detection robustness with recurrent neural networks. *arXiv preprint arXiv:1602.07749*.

Truc-Vien T Nguyen and Alessandro Moschitti. 2011. End-to-end relation extraction using distant supervision from external semantic repositories. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 277–282.

Trung Minh Nguyen and Thien Huu Nguyen. 2019. One for all: Neural joint modeling of entities and events. In *AAAI*.

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

Sachin Pawar, Girish K Palshikar, and Pushpak Bhattacharyya. 2017. Relation extraction: A survey. *arXiv preprint arXiv:1712.05191*.

Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen-tau Yih. 2017. Cross-sentence n-ary relation extraction with graph lstms. *Transactions of the Association for Computational Linguistics*, 5:101–115.

Soujanya Poria, Erik Cambria, and Alexander Gelbukh. 2016. Aspect extraction for opinion mining with a deep convolutional neural network. *Knowledge-Based Systems*, 108:42–49.

Amir Pouran Ben Veyseh, Thien Huu Nguyen, and Dejing Dou. 2019. Graph based neural networks for event factuality prediction using syntactic and semantic structures. In *ACL*.

Amir Pouran Ben Veyseh, Nasim Nouri, Franck Dernoncourt, Dejing Dou, and Thien Huu Nguyen. 2020. Introducing syntactic structures into target opinion word extraction with deep learning. page EMNLP.

James Pustejovsky. 2006. Timebank 1.2. In *LDC*.

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.

Vahed Qazvinian, Emily Rosengren, Dragomir Radev, and Qiaozhu Mei. 2011. Rumor has it: Identifying misinformation in microblogs. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1589–1599.

Pengda Qin, Weiran Xu, and William Yang Wang. 2018. Robust distant supervision relation extraction via deep reinforcement learning. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics.*

Delai Qiu, Yuanzhe Zhang, Xinwei Feng, Xiangwen Liao, Wenbin Jiang, Yajuan Lyu, Kang Liu, and Jun Zhao. 2019. Machine reading comprehension using structural knowledge graph-aware network. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5898–5903.

Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2011. Opinion word expansion and target extraction through double propagation. *Computational linguistics*, 37(1):9–27.

Chris Quirk and Hoifung Poon. 2016. Distant supervision for relation extraction beyond the sentence boundary. *arXiv preprint arXiv:1609.04873*.

Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher D Manning. 2010. A multi-pass sieve for coreference resolution. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 492–501.

Altaf Rahman and Vincent Ng. 2009. Supervised models for coreference resolution. In *Proceedings of the 2009 conference on empirical methods in natural language processing*, pages 968–977.

Sudha Rao, Daniel Marcu, Kevin Knight, and Hal Daumé III. 2017. Biomedical event extraction using abstract meaning representation. In *BioNLP 2017*, pages 126–135.

Lev Ratinov and Dan Roth. 2012. Learning-based multi-sieve co-reference resolution with knowledge. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1234–1244.

Sebastian Riedel and Andrew McCallum. 2011. Robust biomedical event extraction with dual decomposition and minimal domain adaptation. In *BioNLP Shared Task 2011 Workshop*.

Ellen Riloff and Jay Shoen. 1995. Automatically acquiring conceptual patterns without an annotated corpus. In *Third Workshop on Very Large Corpora*.

Alan Ritter, Oren Etzioni, and Sam Clark. 2012. Open domain event extraction from twitter. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1104–1112.

Rachel Rudinger, Aaron Steven White, and Benjamin Van Durme. 2018. Neural models of factuality. *arXiv preprint arXiv:1804.02472*.

Sriparna Saha, Amit Majumder, Md Hasanuzzaman, and Asif Ekbal. 2011. Bio-molecular event extraction using support vector machine. In *2011 Third International Conference on Advanced Computing*, pages 298–303. IEEE.

Taneeya Satyapanich, Francis Ferraro, and Tim Finin. 2020. Casie: Extracting cybersecurity event information from text. *UMBC Faculty Collection*.

Roser Saurí and James Pustejovsky. 2009. Factbank: a corpus annotated with event factuality. *Language resources and evaluation*, 43(3):227.

Ozge Sevgili, Artem Shelmanov, Mikhail Arkhipov, Alexander Panchenko, and Chris Biemann. 2020. Neural entity linking: A survey of models based on deep learning. *arXiv preprint arXiv:2006.00575*.

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 *AAAI*.

Ivan Shamshurin. 2012. Extracting domain-specific opinion words for sentiment analysis. In *Mexican International Conference on Artificial Intelligence*, pages 58–68.

Yuming Shang, Heyan Huang, Xin Sun, and Xianling Mao. 2020. Learning relation ties with a force-directed graph in distant supervised relation extraction. *arXiv preprint arXiv:2004.10051*.

Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. 2018. Ordered neurons: Integrating tree structures into recurrent neural networks. *arXiv preprint arXiv:1810.09536*.

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.

Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational linguistics*, 27(4):521–544.

Sasha Spala, Nicholas A Miller, Yiming Yang, Franck Dernoncourt, and Carl Dockhorn. 2019. Deft: A corpus for definition extraction in free-and semi-structured text. In *Proceedings of the 13th Linguistic Annotation Workshop*, pages 124–131.

Nikolaos Stylianou and Ioannis Vlahavas. 2019. A neural entity coreference resolution review. *arXiv preprint arXiv:1910.09329*.

Ang Sun, Ralph Grishman, and Satoshi Sekine. 2011. Semi-supervised relation extraction with large-scale word clustering. In *ACL*.

Duyu Tang, Bing Qin, and Ting Liu. 2016. Aspect level sentiment classification with deep memory network. In *EMNLP*.

Jorge A Vanegas, Sérgio Matos, Fabio González, and José L Oliveira. 2015. An overview of biomolecular event extraction from scientific documents. *Computational and mathematical methods in medicine*, 2015.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Amir Pouran Ben Veyseh. 2016. Cross-lingual question answering using common semantic space. In *Proceedings of TextGraphs-10: the workshop on graph-based methods for natural language processing*, pages 15–19.

Amir Pouran Ben Veyseh, Franck Dernoncourt, Dejing Dou, and Thien Huu Nguyen. 2020a. A joint model for definition extraction with syntactic connection and semantic consistency. In *AAAI*.

Amir Pouran Ben Veyseh, Franck Dernoncourt, My Tra Thai, Dejing Dou, and Thien Huu Nguyen. 2020b. Multi-view consistency for relation extraction via mutual information and structure prediction. In *AAAI*, pages 9106–9113.

Amir Pouran Ben Veyseh, Franck Dernoncourt, Quan Hung Tran, and Thien Huu Nguyen. 2020c. What does this acronym mean? introducing a new dataset for acronym identification and disambiguation.

Amir Pouran Ben Veyseh, Javid Ebrahimi, Dejing Dou, and Daniel Lowd. 2017. A temporal attentional model for rumor stance classification. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 2335–2338.

Amir Pouran Ben Veyseh, Thien Huu Nguyen, and Dejing Dou. 2019a. Improving cross-domain performance for relation extraction via dependency prediction and information flow control. In *IJCAI*.

Amir Pouran Ben Veyseh, Nasim Nour, Franck Dernoncourt, Quan Hung Tran, Dejing Dou, and Thien Huu Nguyen. 2020d. Improving aspect-based sentiment analysis with gated graph convolutional networks and syntax-based regulation. *EMNLP*.

Amir Pouran Ben Veyseh, My T Thai, Thien Huu Nguyen, and Dejing Dou. 2019b. Rumor detection in social networks via deep contextual modeling. In *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 113–120.

Amir Pouran Ben Veyseh, My T. Thai, Thien Huu Nguyen, and Dejing Dou. 2019c. Rumor detection in social networks via deep contextual modeling. In *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*.

Joachim Wagner, Piyush Arora, Santiago Cortes, Utsab Barman, Dasha Bogdanova, Jennifer Foster, and Lamia Tounsi. 2014. NRC-canada-2014: Detecting aspects and sentiment in customer reviews. In *SemEval*.

Joachim Wagner, Piyush Arora, Santiago Cortes, Utsab Barman, Dasha Bogdanova, Jennifer Foster, and Lamia Tounsi. 2016. Effective LSTMs for target-dependent sentiment classification. In *COLING*.

Difeng Wang, Wei Hu, Ermei Cao, and Weijian Sun. 2020a. Global-to-local neural networks for document-level relation extraction. *arXiv preprint arXiv:2009.10359*.

Haoyu Wang, Muhao Chen, Hongming Zhang, and Dan Roth. 2020b. Joint constrained learning for event-event relation extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 696–706, Online. Association for Computational Linguistics.

Haoyu Wang, Muhao Chen, Hongming Zhang, and Dan Roth. 2020c. Joint constrained learning for event-event relation extraction. *arXiv preprint arXiv:2010.06727*.

Linlin Wang, Zhu Cao, Gerard de Melo, and Zhiyuan Liu. 2016a. Relation classification via multi-level attention cnns. In *EMNLP*.

Rui Wang, Deyu Zhou, and Yulan He. 2019a. Open event extraction from online text using a generative adversarial network. *arXiv preprint arXiv:1908.09246*.

Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2016b. Recursive neural conditional random fields for aspect-based sentiment analysis. In *EMNLP*.

Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2017. Coupled multi-layer attentions for co-extraction of aspect and opinion terms. In *AAAI*.

Xiaozhi Wang, Ziqi Wang, Xu Han, Zhiyuan Liu, Juanzi Li, Peng Li, Maosong Sun, Jie Zhou, and Xiang Ren. 2019b. Hmeae: Hierarchical modular event argument 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 5781–5787.

Eline Westerhout. 2009. Definition extraction using linguistic and structural features. In *Proceedings of the 1st Workshop on Definition Extraction*.

Aaron Steven White, Drew Reisinger, Keisuke Sakaguchi, Tim Vieira, Sheng Zhang, Rachel Rudinger, Kyle Rawlins, and Benjamin Van Durme. 2016. Universal decompositional semantics on universal dependencies. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1713–1723.

Sam Joshua Wiseman, Alexander Matthew Rush, Stuart Merrill Shieber, and Jason Weston. 2015. Learning anaphoricity and antecedent ranking features for coreference resolution. 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)*. Association for Computational Linguistics.

Junshuang Wu, Richong Zhang, Yongyi Mao, Hongyu Guo, Masoumeh Soflaei, and Jinpeng Huai. 2020a. Dynamic graph convolutional networks for entity linking. In *Proceedings of The Web Conference 2020*, pages 1149–1159.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2019. Zero-shot entity linking with dense entity retrieval. *arXiv preprint arXiv:1911.03814*.

Wei Wu, Fei Wang, Arianna Yuan, Fei Wu, and Jiwei Li. 2020b. Corefqa: Coreference resolution as query-based span prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6953–6963.

Zhen Wu, Fei Zhao, Xin-Yu Dai, Shujian Huang, and Jiajun Chen. 2020c. Latent opinions transfer network for target-oriented opinion words extraction. *arXiv preprint arXiv:2001.01989*.

Wei Xiang and Bang Wang. 2019. A survey of event extraction from text. *IEEE Access*, 7:173111–173137.

Hu Xu, Bing Liu, Lei Shu, and Philip S Yu. 2018. Double embeddings and cnn-based sequence labeling for aspect extraction. In *ACL*.

Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015. Classifying relations via long short term memory networks along shortest dependency paths. In *EMNLP*.

Ikuya Yamada and Hiroyuki Shindo. 2019. Pretraining of deep contextualized embeddings of words and entities for named entity disambiguation. *arXiv preprint arXiv:1909.00426*.

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

Bishan Yang and Tom M. Mitchell. 2016. Joint extraction of events and entities within a document context. In *NAACL-HLT*.

Sen Yang, Dawei Feng, Linbo Qiao, Zhigang Kan, and Dongsheng Li. 2019. Exploring pre-trained language models for event extraction and generation. In *ACL*.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 2013–2018.

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. 2019. Docred: A large-scale document-level relation extraction dataset. *arXiv preprint arXiv:1906.06127*.

Yichun Yin, Furu Wei, Li Dong, Kaimeng Xu, Ming Zhang, and Ming Zhou. 2016. Unsupervised word and dependency path embeddings for aspect term extraction. In *IJCAI*.

Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020. Named entity recognition as dependency parsing. *arXiv preprint arXiv:2005.07150*.

Mo Yu, Matthew R Gormley, and Mark Dredze. 2015. Combining word embeddings and feature embeddings for fine-grained relation extraction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1374–1379.

Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. 2019. Graph transformer networks. In *Advances in Neural Information Processing Systems*, pages 11983–11993.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *Journal of machine learning research*, 3(Feb):1083–1106.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *COLING*.

Shuang Zeng, Runxin Xu, Baobao Chang, and Lei Li. 2020. Double graph based reasoning for document-level relation extraction.

Chen Zhang, Qiuchi Li, and Dawei Song. 2019a. Aspect-based sentiment classification with aspect-specific graph convolutional networks. In *EMNLP*.

Junchi Zhang, Yanxia Qin, Yue Zhang, Mengchi Liu, and Donghong Ji. 2019b. Extracting entities and events as a single task using a transition-based neural model. In *IJCAI*.

Tongtao Zhang, Spencer Whitehead, Hanwang Zhang, Hongzhi Li, Joseph Ellis, Lifu Huang, Wei Liu, Heng Ji, and Shih-Fu Chang. 2017a. Improving event extraction via multimodal integration. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 270–278.

Yufeng Zhang, Xueli Yu, Zeyu Cui, Shu Wu, Zhongzhen Wen, and Liang Wang. 2020a. Every document owns its structure: Inductive text classification via graph neural networks. *arXiv preprint arXiv:2004.13826*.

Yuhao Zhang, Peng Qi, and Christopher D Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *acl*.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. 2017b. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45.

Yunyan Zhang, Guangluan Xu, Yang Wang, Daoyu Lin, Feng Li, Chenglong Wu, Jingyuan Zhang, and Tinglei Huang. 2020b. A question answering-based framework for one-step event argument extraction. In *IEEE Access, vol 8, 65420-65431*.

Zhou Zhao, Hanqing Lu, Vincent W Zheng, Deng Cai, Xiaofei He, and Yueting Zhuang. 2017. Community-based question answering via asymmetric multi-faceted ranking network learning. In *AAAI*, volume 17, pages 3532–3539.

GuoDong Zhou and Jian Su. 2002. Named entity recognition using an hmm-based chunk tagger. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 473–480.

GuoDong Zhou, Jian Su, Jie Zhang, and Min Zhang. 2005a. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd annual meeting of the association for computational linguistics (acl'05)*, pages 427–434.

Guodong Zhou, Jian Su, Jie Zhang, and Min Zhang. 2005b. Exploring various knowledge in relation extraction. In *ACL*.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *ACL*.

Li Zhuang, Feng Jing, and Xiao-Yan Zhu. 2006. Movie review mining and summarization. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 43–50.

Arkaitz Zubiaga, Elena Kochkina, Maria Liakata, Rob Procter, and Michal Lukasik. 2016. Stance classification in rumours as a sequential task exploiting the tree structure of social media conversations. *arXiv preprint arXiv:1609.09028*.

Stefan Zwicklbauer, Christin Seifert, and Michael Granitzer. 2016. Robust and collective entity disambiguation through semantic embeddings. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 425–434.