IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received September 17, 2020, accepted October 20, 2020, date of publication October 30, 2020, date of current version November 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3034907

A History and Theory of Textual Event Detection and Recognition

YANPING CHEN[1,2], **ZEHUA DING**[1], **QINGHUA ZHENG**[3], **(Member, IEEE)**, **YONGBIN QIN**[1,2], **RUIZHANG HUANG**[1,2], **AND NAZARAF SHAH**[4]

[1]Guizhou Provincial Key Laboratory of Public Big Data, College of Computer Science and Technology, Guizhou University, Guiyang 550025, China
[2]Key Laboratory of Intelligent Medical Image Analysis and Precise Diagnosis of Guizhou Province, Guizhou University, Guiyang 550025, China
[3]Department of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049, China
[4]School of Computing, Electronics and Maths, Coventry University, Coventry CV1 5FB, U.K.

Corresponding author: Yanping Chen (ypench@gmail.com)

This work was supported in part by the National Natural Science Foundation of China through the Joint Funds under Grant U1836205, in part by the National Natural Science Foundation of China through the Major Research Program under Grant 91746116, in part by the National Natural Science Foundation of China under Grant 62066007 and Grant 62066008, in part by the Major Special Science and Technology Projects of Guizhou Province under Grant [2017]3002, and in part by the Key Projects of Science and Technology of Guizhou Province under Grant [2020] 1Z055.

ABSTRACT There is large and growing amounts of textual data that contains information about human activities. Mining interesting knowledge from this textual data is a challenging task because it consists of unstructured or semistructured text that are written in natural language. In the field of artificial intelligence, event-oriented techniques are helpful in addressing this problem, where information retrieval (IR), information extraction (IE) and graph methods (GMs) are three of the most important paradigms in supporting event-oriented processing. In recent years, due to information explosions, textual event detection and recognition have received extensive research attention and achieved great success. Many surveys have been conducted to retrospectively assess the development of event detection. However, until now, all of these surveys have focused on only a single aspect of IR, IE or GMs. There is no research that provides a complete introduction or a comparison of IR, IE, and GMs. In this article, a survey about these techniques is provided from a broader perspective, and a convenient and comprehensive comparison of these techniques is given. The hallmark of this article is that it is the first survey that combines IR, IE and GMs in a single frame and will therefore benefit researchers by acting as a reference in this field.

INDEX TERMS Event detection, event recognition, information extraction, information retrieval.

I. INTRODUCTION

Written language is the most convenient format for expressing human thought and behavior. It has been widely adopted for recording human knowledge. For example, the SiKu Quan Shu is the largest collection of Chinese literature compiled during the years 1773-1782 in the Qianlong period of the Qing dynasty. The collection contains 36,304 volumes and approximately 0.8 billion Chinese characters. Due to the emergence of the Internet, textual data have been accumulating rapidly. For example, the indexed Web contains at least 6.04 billion pages,[1] Google Scholar has 160 million indexed documents, and PubMed,[2] a search engine accessing the MEDLINE database, contains more than 29 million citations.

Besides the large volume, there are two important aspects of the current textual data: accessibility and interactivity. Accessibility is supported by the development of the Internet. High-speed Internet infrastructure and high-performance computing terminals enable frequent and convenient browsing and downloading at anytime and anywhere. Various kinds of data (e.g., news, comments, papers, encyclopedias or novels) are available to anyone who is searching for information through the Internet. The interactivity prompts individuals to post their opinions. When exploring information, we are not just information consumers but also information producers or more accurately *information prosumers*. Diverse social media, such as Facebook and Twitter, make it easier for individuals to express opinions easily and freely. Accelerated by accessibility and interactivity, information is exploded, developed and spread more quickly and more influentially than ever before. This information explosion provokes the desire to mine valuable information from the mass of textual

[1]Sunday, 09 February, 2020. https://www.worldwidewebsize.com/
[2]https://www.ncbi.nlm.nih.gov/pubmed/

The associate editor coordinating the review of this manuscript and approving it for publication was Gianmaria Silvello.

data. The data can be beneficial for various applications, e.g., information acquisition, collaborative recommendation, knowledge management or decision making.

Event-oriented methods provide effective ways to support the requirements for handling the surge of information. They are helpful for understanding the when, where and what of events that have happened around the world. However, the definition of an ''event'' is very vague. It has been adopted in many fields and has been used in reference to many semantic concepts. Various definitions have been presented by researchers. In this survey, we focus on ''events'' expressed through text. We named these events as ''textual events''. In the field of artificial intelligence, information retrieval (IR) and information extraction (IE) and graphic methods (GMs) are three of the most important paradigms to support textual event detection and recognition. Before presenting our taxonomy about event detection, these paradigms are introduced briefly as follows.

Traditionally, an IR system ranks documents relative to a query [1]. In this paradigm, textual data is processed at the document level. A document is seen as a bag-of-words and is represented by a word vector. Elements of the vector are assessed with term weighting, e.g., the term frequency–inverse document frequency (TF-IDF) [2]. The model is known as the vector space model (VSM) [3]. It maps a document into a high-dimensional space. The similarity among documents is calculated as the distance between vectors. In this paradigm, an event is defined as a cluster of documents, which describes a particular ''story'' that occurred at a specific time and place [4]–[6]. The IR-based system is effective for retrieving similar documents. The main shortcoming of this paradigm is that it does not support content-based recognition. Users still need to skim through the document content, which is a laborious and time-consuming task. To reduce ''information overload'' caused by the flooding of documents, IE provides an effective complementary solution.

Instead of processing textual data at the document level, IE systems extract designated information from the content of documents, e.g., named entities, temporal expressions, and entity relationships. In this field, an event is often defined as a frame or a template with slots, which can be filled with parameters such as actors, location, and time. Every event has a predefined structure. For example, a birthday event may contain parameters such as host, guest, location, and time. An event is triggered by special words (anchor words, e.g., a verb) [7]. When an event is triggered, the relevant parameters are extracted according to a predefined structure. In this paradigm, the detection task is often modelled as a sequence labeling process, where the hidden Markov model (HMM) [8], conditional random fields (CRF) [9] or long short-term memory (LSTM) [10] are commonly adopted. The main problem is that processing at the sentence level suffers from a serious sparse feature problem because a sentence often contains a limited number of words.

There are many knowledge bases (e.g., Freebase and Yago) that have been developed to support event-oriented information exploration [11], [12]. In this paradigm, linguistic units or semantic concepts (e.g., named entities and entity relations) are extracted and organized into a graph representation [12]–[14]. A graph contains topological information about documents. In this paradigm, an event can be defined as a subgraph (or a node). A graph representation is helpful when mining the underlying structure of events in documents and it gives rise to novel solutions for event detection and recognition. Event detection and recognition can be implemented by graph mining algorithms (e.g., community detection [15]). It can be used to support many analytical methods (e.g., statistical relational learning and entity disambiguation) and to provide a visual interface for human-oriented information exploration.

As discussed above, according to the characteristics of IR, IE and GMs, we divide definitions of events into three categories: *documental events*, *frame events* and *graphic events*. A documental event is a cluster of documents that contains ample information about an event. A frame event is a template with a predefined semantic structure. A graphic event is a graph representation with coupled semantic information from a semantic graph. In this article, we conduct a survey about event detection and recognition. The rest of this article is organized as follows. In Section II, the information retrieval, information extraction and graphic method are introduced as a background of event detection and recognition. The discussions of documental events, frame events and graphic events are presented in Section III, Section IV and Section V, respectively. We provide the conclusion in Section VI.

II. BACKGROUND

Three types of event definitions are presented in this article. They are relevant to three fields: IR, IE and GMs. In the following, we introduce these fields as the background for textual event detection and recognition.

A. INFORMATION RETRIEVAL

The idea of IR originated from the discipline of library science. It dates back to approximately 3000 BC. It was used by Sumerians for accessing archived clay tablets [16]. At its early stages, IR was mainly used by librarians to retrieve indexed items, such as books or documents [17]. The catalogues are often alphabetically ordered by author, title and subject. For example, catalogue cards were widely used to index archived items. After the catalogue compilation, the retrieval task was performed by human labor. At the beginning of the nineteenth century, mechanical devices were designed to operate catalogue cards automatically [18].

The phrase, ''information retrieval'', was first coined in March of 1950 by Calvin Mooers at a conference [19]. The reliability of automatically retrieving documents has been proven by Cleverdon and Keen [20]. It has been shown that automatic indexing is comparable to manual indexing. The conclusion is induced from the Cranfield collection corpus constructed in the late 1960s', which contains 1400 documents and 225 queries [21]. Research on

IR has been accelerated by the development of the Internet. The Internet is full of unstructured or semi-structured documents [22], [23], which poses a challenge for traditional IR methods.

In traditional IR systems, a query is required as an input. The query can be structured (e.g., regular expression) or unstructured (e.g., noun phrase or natural language). An IR system retrieves relative documents by measuring the similarity between queries and documents. Various approaches have been proposed to measure the similarity between queries and documents, e.g., Boolean models [24], natural language processing (NLP) models [25] and vector space models (VSMs) [16], [22].

In Boolean models, documents are indexed by a set of terms (or keywords) collected manually or automatically extracted from titles, abstracts or documents. The match between queries and documents is based on the satisfaction of the indexed terms. It is mainly implemented by a string matching method or Boolean operations. In addition to indexed terms, syntax or semantic information are explored by NLP methods. NLP methods have the advantage of utilizing techniques developed in natural language processing. The VSM represents a document (paragraph or sentence) as a term vector with a fixed-length dimension [3]. The similarity between documents is calculated by using the distances between the term vectors. Compared with the Boolean model, the VSM has robust performance, and no manually annotated keywords are required.

According to Carpineto and Romano [26], the average query length is 2.30 words. Many of these queries are poorly defined. Therefore, researchers have developed many techniques for improving the quality of queries, for example, *relevance feedback*, *interactive query refinement* or *word sense disambiguation* [27]–[30].

There are many evaluation communities that aim to promote state-of-the-art technologies to support IR research. We introduce two of them as examples: the Text REtrieval Conference (TREC)[3] and topic detection and tracking (TDT).[4]

TREC has been co-sponsored by DARPA and NIST since 1992. It is an on-going series of workshops organized by the retrieval group under the TIPSTER Text Program [21]. Every TREC workshop consists of a set of tracks in which particular tasks are defined, such as the web track, microblog track, medical records track, and legal track. For each round, the TREC publishes a large-scale evaluation collection and provides the evaluation techniques in advance. All participants receive a static set of documents and are asked to return ranked documents, where the documents are ordered by probabilities according to whether they are relevant to queries. It enables the results to be compared across different systems. In the early stage, TREC mainly focuses on retrieving relevant documents in a given corpus. At later stages, many tracks were defined for extracting factual information from documents, e.g., the knowledge base acceleration track or question answering track.

TDT is another evaluation community belonging to the TIDES[5] project of the DARPA. The TDT started with a pilot study in 1997 and ended in 2004. Under the TDT setting, topics (or events) are detected to enable users to manage text streams in real time (e.g., newswire and broadcast news). To process text streams, new methodologies are required, e.g., segmenting a stream of data, finding new topics or keeping track of topics. In each conference, the community proposed several tasks to evaluate the methodologies developed in this field. For example, for the TDT evaluation in 2004, there are five research tasks: story link detection, new event detection, topic detection,[6] topic tracking and hierarchical topic detection.

B. INFORMATION EXTRACTION

Information extraction (IE) focuses on extracting semantic or syntax units from documents [31]–[36]. The first IE model was proposed by Schank [37] and is known as *conceptual dependency theory* (CDT). It assumes that the main conceptualizations in a sentence are expressed by concepts (actions and concrete nouns) and the dependencies among them. It assumes that an action is the focus of a linguistic structure. The structure defines the dependency relationships among concepts. Given an input string, scripts (stereotyped causal chains) are used to extract the relevant ‘‘conceptual cases’’ of each action (e.g., objective, recipient or instrumental). Based on this theory, Yale University designed a system named SAM. It modifies scripts of CDT by using sketchy scripts to extract important events with lower-level text analyses [38].

Frame theory is another popular theory proposed by Minsky [39], where ‘‘a frame is a data structure representing a stereotyped situation’’ Minsky [39]. A frame defines the structure of an event with a certain number of slots. The slots can be filled with information about a specific event. For example, all children's birthday parties have hosts, guests and a birthday cake. These objects and the relationship among them constitute a birthday party event. Therefore, a frame of a birthday party event has slots that can be filled with parameters such as host, guest, and action. To define a frame, the number of slots and the type of each parameter should be predetermined according to a common structure of a specific event.

Lehnert [40] presented a *plot unit connectivity graph* for story summaries. Plot units are conceptual structures used to represent the conceptual content. Four types of relations are defined to link plot units, e.g., *motivation*, *termination*, *actualization* and *equivalence*. The relations have three types of affect states: negative events, positive events and mental states. When all plot units of a story are connected, the graph

[3]http://trec.nist.gov/

[4]http://www.itl.nist.gov/iad/mig//tests/tdt/

[5]The *Translingual Information Detection, Extraction, and Summarization* project.

[6]It was replaced by hierarchical topic detection which is a new research task in TDT 2004.

may grow to a complex network. To reduce the complexity, Rumelhart [41] proposed a story grammar for plotting human narrative stories. Sager [42] introduced a sublanguage analysis method to extract information from clinical reporting. Many systems were designed between the 1980s and 1990s. An overview of these systems can be found in Hahn [43].

IE research has been prompted by a series of evaluation conferences [44]. Five conferences (or communities) are influential: SemEval (semantic evaluation),[7] the Message Understanding Conference (MUC),[8] Automatic Content Extraction (ACE),[9] the Text Analysis Conference (TAC)[10] and the BioNLP shared task series.[11]

SemEval is an ongoing series of evaluation conferences. It is developed from the word sense evaluation series.[12] SemEval focuses on evaluating semantic analysis systems. It tries to promote mechanisms and methodologies for linguistic computation with meaning. From Senseval-1 to Senseval-3, *word sense disambiguation* received special attention. The recent workshops focus on areas regarding *textual similarity and question answering*, *detecting sentiment*, *coreference*, *information extraction*, etc. [45].

During the 1990s, the MUC was supported by the Science Applications International Corporation (SAIC) to foster the development of novel and improved methods for IE. The MUC saw the development of information extraction. In the first MUC, there was no definition for the format of the output and evaluation criterion. Participants were free to determine the output format according to their understanding of the task. Then, the community summarized the results and defined the direction of the following conference. The MUC-2 crystallized event recognition as frame filling tasks, whereas the MUC-6 coined the task of ''named entity'' recognition to support sophisticated extraction tasks. From 1987 to 1997, the MUC was held seven times and was then replaced by the ACE program.

ACE was inherited from the MUC. The tasks of ACE were more complex and subtle than those of the MUC. The ACE program extracted linguistic units from the point of natural language understanding. Five linguistic units were defined by the ACE evaluation criteria: entities, times, values, relations and events. Four tasks were proposed: entity detection and tracking (EDT), relation detection and characterization (RDC), entity linking (LNK), and event detection and characterization (VDC) [7]. ACE was held annually from 1999 to 2008. It was replaced by a track in the Text Analysis Conference (TAC) in 2009.

The Text Analysis Conference (TAC) was initiated and organized by the retrieval group of the Information Access Division (IAD) in the National Institute of Standards and Technology (NIST). TAC has four tracks: recognizing textual entailment, question answering, summarization and knowledge base population (KBP), where KBP is similar to ACE event recognition. KBP aims to populate knowledge bases through systems that automatically extract information from texts. This track has three areas: slot filling, entity linking and cold start knowledge base population.

The BioNLP refers to a series of shared tasks and is characterized by the evaluation data [46]–[49]. It focuses on accelerating methods and strategies developed for biomedical and molecular biology texts. The BioNLP shared tasks incorporate theories and methodologies developed from natural language processing, computational linguistics and medical informatics. The tasks of the BioNLP are similar to those of the ACE, e.g.; named entity recognition, relation recognition and event recognition. In contrast to named entities such as *person*, *organization* or *location*, BioNLP focuses on bacterial molecules, cells, proteins and genes [50]. Relation types among them are *implicit renaming*, *biological proof*, and *protein encoding* [50]. Event types are *transcription*, *binding* and *regulation* [46].

C. GRAPHIC METHOD

Extracting linguistic units into networks or graphs is helpful to reveal the potential relationships among linguistic units in a document (or documents). A graphic representation enables topological analyses such as social networks and complex networks. It can give rise to novel solutions for a variety of NLP tasks. Based on graphic representations, the topological properties of linguistic units show the macroscopic characteristics of languages (e.g., Zipf's law [51]). They are useful for modeling language with a flexible approach.

Graphic representations show an increased interest in the field of natural language processing. Various networks or graphs are proposed to represent knowledge in a structured form. In this article, according to the abstract level of nodes in a network, we roughly class them into three categories: *co-occurrence network*, *syntactic network* and *semantic network*.

1) CO-OCCURRENCE NETWORK

In co-occurrence networks, nodes are terms (e.g., hyperlinks, author names or words). Edges between nodes indicate that the linked nodes co-occur in a given window, e.g., a sentence, a page or a document. The co-occurrence information (e.g., cocitation, coword, and colink) between terms can be used to explore and understand the structures of the underlying documents [52]. For example, the PageRank algorithm ranks Web pages by the colinks between them. Author cocitations can be used to analyze potential patterns among papers [53]. In addition, lexical (or word) co-occurrence networks show meaningful characteristics of languages [54]–[56].

Approaches to analyse co-occurrence networks are mainly based on statistical characteristics. The foundation to utilizing a co-occurrence network is that terms do not occur in a random way. Co-occurred terms share some syntactic or semantic relations. For example, a coauthor relation indicates that

[7] http://www.senseval.org/index.html
[8] http://www-nlpir.nist.gov/related_projects/muc/
[9] http://www.itl.nist.gov/iad/mig/tests/ace/
[10] http://www.nist.gov/tac/
[11] http://2011.bionlp-st.org/
[12] https://en.wikipedia.org/wiki/SemEval

they belong to the same field. Co-occurring words in a sentence show a syntactic dependency [55]. For a co-occurrence network, two issues are very important. The first is the size of the window where terms co-occurred. In a lexical co-occurrence network, the window can be set as a sentence, a paragraph or a document. A smaller granularity means a stronger coherence between terms, but it increases the sparsity of a network. Another issue regarding co-occurrence networks is the object that a term refers to. In the PageRank algorithm, a hyperlink is unambiguously referred to as a unique Web page. Therefore, co-occurrence networks based on colinks truly reflect the underlying structure of Web pages. However, because of homonyms, in an author cocitation network, noise will be introduced with the ambiguity problem. The problem is worse in a lexical co-occurrence network. Depending on context, the same word can refer to different objects.

2) SYNTACTIC NETWORK

The constraint in a co-occurrence network is very loose and only depends on the co-occurrence information. Because word co-occurrences in a language seem evidently arbitrary [57], it is difficult to reveal the underlying structure of a language with a co-occurrence network [58]. Depending on the context of a sentence, many words are polysemic and ambiguous. They cannot reflect the true structure of a language. To capture the syntactic dependency in a sentence, syntactic information is introduced to strengthen the relationship among words. It is also known as a syntactic network. To identify a link between two nodes, it is required that the linked words are syntactically dependent.

Syntactic dependency networks have been widely discussed to analyze the characteristics of a language [59]–[62]. In a traditional syntactic network, nodes are denoted as words. They are linked by edges indicating the syntactic dependency between them. Syntactic networks are constructed by organizing the words with syntactical dependencies in a corpus. Every sentence is first parsed into a syntactic tree. Then, the words with syntactic dependencies are linked by labelled edges [59], [61], [63]. In syntactic networks, many types of edges have been defined. The Boolean relation is the most widely used edge type I Cancho *et al.* [59], [61]. Some researchers use grammatical relations, where every edge has a label indicating the type of grammatical dependence between words [60]. Researchers used a thesaurus dictionary to link words that express similar concepts (e.g., the conceptual network Motter *et al.* [64]).

Based on syntactic networks, studies have shown that syntactic networks have a small-world effect and scale-free distribution of degrees [55], [59]. These conclusions are helpful for understanding the characteristics of a language. The main problem of syntactic networks is that because the current techniques used to parse a sentence are difficult and error-prone, it is difficult to construct a network precisely and automatically.

3) SEMANTIC NETWORK

Semantic networks organize knowledge in graph representations. In this network, nodes represent semantic units, e.g., names of people, locations or organizations. Edges between them represent semantic relations, e.g., *location*, *part-whole*, and *physical*. Semantic networks have received extensive research attention. They can be roughly divided into three paradigms: logic-based semantic networks (e.g., ontology), scalable knowledge databases (e.g., Freebase [65]) and semantic networks constructed by open information extraction.

Logic-based semantic networks refer to networks that have a coherent knowledge representation, e.g., Cyc [66] and WordNet [67]. Because the qualities of the network (e.g., consistency, completeness, independence or decidability) are important for a logic-based semantic network, they are often constructed by domain experts and focused on a closed domain. For example, Masterman [68] proposed a *semantic net*, which encodes language semantic features for the purpose of machine understanding. Ceccato and Maretti [69] presented a *correlational net*, which defined 56 different relations, e.g., kinship relations, case relations, and different kinds of attributes. Sowa [70] designed a *conceptual graph*. A conceptual graph represents logic as a graph representation. A conceptual graph supports logic operators directly. Other semantic networks (e.g., a dependency graph [71], WordNet [67], and Cyc [66]) have also been proposed. Logic-based networks have strong constraints on network quality. They are very beneficial for natural language processing.

In the second paradigm, instead of focusing on a closed domain, networks are extended into open fields. These networks try to merge diverse knowledge in a unified framework. Therefore, instead of a rigid structure, these networks try to define a graphical representation with high extensibility and scalability. In this paradigm, systems such as Yago [72] and Freebase [65] are presented to support tasks, such as information retrieval or semisupervised information extraction. Many of these networks are built from semi-structured databases (e.g., Wikipedia) with direct or indirect human intervention (e.g., collaborative methods or searching logs) or with the support of ontologies (e.g., WordNet, OpenCyc [73]). NLP techniques have also been explored for constructing these networks, e.g., Xlike [74] and ECKGs [75]. This type of network is often adopted to represent "static" knowledge (e.g., encyclopedic knowledge), where knowledge is not changed rapidly.

The third paradigm (open semantic networks) integrates knowledge in a dynamic domain. These networks process unstructured data (e.g., plaint texts) from heterogeneous resources. In these networks, nodes are named entities. Edges are relations between them. The number of relation types can be predefined or dynamically created from the input. They can generate a huge network containing more than thousands or millions of nodes and edges. For example, Zhang *et al.* [76] extracted knowledge from cross-lingual data. Angeli *et al.* [77] focused on canonically structured sentences with a few patterns. To construct a network, distant (or weak) supervision [78], [79] and bootstrapping [80]–[83] are commonly used. They often adopt a knowledge database to guide the construction process. Researchers have proposed many systems, e.g., T EXTR UNNER [84], K NOWI TA LL [85], [86], WOE [87], and StatSnowBall [88].

III. DOCUMENTAL EVENT

A documental event refers to a cluster of documents describing the same event. In this field, the term "topic" is also widely used by researchers to represent a documental event [89]. The main difference between them is that documental events have a specific time stamp or a location to indicate the occurrence of the event. In this section, the definitions and techniques in documental event recognition are first presented. Then, three scenarios, (*segmentation*, *event detection and tracking* and *event organization*), are discussed in detail.

A. DEFINITIONS AND NOTATIONS

In the task of event detection and tracking, three terms are often mentioned: "story", "event" and "topic". In many studies, they have been used indiscriminately. However, in this field, there are subtle differences among them. A "story" represents a stream of text in a news article that transmits specific information, e.g., a newswire. An event is defined as "a particular thing that happens at a specific time and place" [90], e.g., a plane crash or a meeting. A topic is a series of related events or activities. In the TDT evaluation corpus, every story belongs to a topic, and a story may contains one or more event(s). They are all defined as clusters of documents representing different granularities of documental contents.

Traditionally, documental events are defined as a flattened structure, where documents in different documental events are mutually exclusive. However, in real applications, a hierarchical representation is effective in expressing the relations among events. Therefore, event organization is introduced to model the relations among documental events. Organized events are often represented in a tree structure. The parent-child relation between nodes denotes the different granularities of events in a corpus. This strategy is also helpful in supporting tasks, such as new event detection or event tracking.

An IR system also outputs clustered documents, but it is different from documental event detection. The latter has an obvious difference: chronology. In documental events, documents are chronological. They tend to occur in a range of times. Therefore, a time gap between similar documents may represent different events [5].

Compared to traditional IR systems, documental event recognition has many unique characteristics. In online event detection, a specific phenomenon is that new vocabularies intermittently occur in arriving documents. The event representation should update the new terms accordingly. An "incremental IDF" method was proposed to handle this problem. It adapts to new terms dynamically [91], [92]. In a multilanguage environment, a "language-specific comparison" method was proposed to compensate for the heterogenicity among languages [92]. To recognize a documental event, the chronological order is an important characteristic. A time span between two events suggests that they are different. For example, two earthquakes may have a time span of approximately thousands of years. As time elapsed, the heterogeneity between the events increases. Therefore, a time window or a decaying function can be adopted to capture this phenomenon [91].

B. TECHNIQUES

The techniques used to extract documental events are mainly rooted in IR [93]. Most IR systems can be adapted to implement documental event recognition. For example, at the TDT evaluation conference, the CMUDIR team presented a system that was initially designed for the TREC-style task. It also achieved high performance in the event detection task [94].

Many documental event detection and recognition systems are based on VSM directly [4]–[6], [92], [94]–[98]. These systems represent a document as a term vector. The values of a vector are set by term weighting. A matrix is used to represent a corpus, where each column of the matrix denotes a document vector. Based on this model, documents are mapped into a measure space. The similarity between two documents can be evaluated by a predefined distance function, e.g., *Manhattan distance*, *cosine*, Hellinger distance [96], Kullback-Leibler divergence, Clarity-based distance or Jensen-Shannon distance [99], [100].

Based on VSM, various techniques are developed to improve system performance, e.g., techniques to improve clustering or evaluation, techniques to link the dependence among events and techniques to fuse results [17], [23], [101]. Among them, the term weighting and space transformation are two important issues.

Term weighting quantifies the significance of terms [102], [103]. Many techniques have been proposed, such as term frequency (TF) [104], inverse document frequency (IDF) [105], TF-IDF [103] and term relevance weight [106]. Term weighting decides the distribution of documents in a measure space. Different term weighting methods lead to different cutting planes in a measure space. Term weighting is also used to select or filter terms that are noisy. Various feature selection methods have been proposed, such as chi-squared statistics [107], pairwise mutual information [57], log-likelihood ratios [108] and Jaccard similarity measures [109].

In real applications, the dimension of VSM may grow exponentially, which leads to the "curse of dimensionality" problem. A large dimension is expensive in memory and computation. Furthermore, capturing the semantic information of terms is also important. To resolve these problems, space transformation has been proposed, such as latent semantic indexing (LSA) [110], probabilistic latent semantics indexing (PLSA) [111] and latent Dirichlet allocation (LDA) [112].

Based on the document matrix, latent semantic indexing (LSI) implements a singular-value decomposition. It maps documents into a latent semantic space represented by a singular value matrix [113]. Probabilistic latent semantics indexing (PLSI) utilizes the phenomenon that terms with different occurrences across documents may indicate latent topics. The latent topics can be modelled by hidden variables [114], [115]. To control the number of parameters and the overfitting problem in PLSI, latent Dirichlet allocation (LDA) proposes several assumptions to restrict the distribution of random variables [112]. These techniques (LSA, PLSA, LDA, etc.) map documents into to a reduced semantic space. They have advantages in exploring the latent semantics of documents and identifying the hidden semantic structure in documents [110], [116].

C. SEGMENTATION

In real applications, many documents contain several relevant events. For example, an airplane crash event usually includes subevents such as smoke escaping from the engine, airplane malfunctions, scared and shaken passengers, etc. They are expressed in a text stream. Because there is no clear boundary between them, it is necessary to segment it into coherent partitions.

Various approaches have been proposed to implement segmentation tasks. They fall into two categories: structure-based approaches and the content-based approaches [6]. There are systems combining two approaches in a unified framework (e.g., [117]).

A structure-based approach defines the segmentation task as a classification problem. It gives a predication based on features extracted from a possible boundary. For example, Beeferman *et al.* [117] proposed an exponential model to extract features correlated with the presence of boundaries. In addition to classifying each boundary independently, a sequence model (e.g., CRF, HMM or LSTM) was introduced to capture the dependency among segmentation boundaries [118]. The model tries to find the most likely tagging sequence for a text stream. McCallum *et al.* [119] used maximum entropy Markov models (MEMMs). These models have reported better performance compared to other models, such as ME-stateless, TokenHMM, and FeatureHMM. Barrow *et al.* [120] proposed a segment pooling model. It jointly segments a document and labels the content of segmentations.

Content-based approaches utilize the meanings of words to find coherent blocks in a document or a text stream. A coherent block refers to a segment with highly related semantic meanings. A change in semantic similarities among contexts may indicate an inhomogeneous block. In this field, two approaches have been proposed to compute the similarities among words: syntagmatic relation and paradigmatic relation [121]. The syntagmatic similarity is based on the co-occurrences of words in a corpus. It represents how words are arranged in sequential texts. The paradigmatic similarity is based on the associative data in a thesaurus. It represents how words are associated. In recent work, Glavaš *et al.* [122] proposed an unsupervised algorithm that uses word embeddings to measure the semantic relationships among segments.

To evaluate a segmentation method, two approaches have been presented: *indirect evaluation* and *direct evaluation* [6]. Indirect evaluation emphasizes support for other tasks. In this approach, a system is first conducted to segment documents. Then, an event detection task is implemented on the segmented output for an evaluation. In the direct evaluation approach, a metric (e.g., precision and recall) is adopted to evaluate the performance of a system. This approach is easy to implement and can evaluate systems across different platforms. However, there are still some shortcomings in this approach because a hard benchmark cannot access the degree of error. To resolve this problem, Beeferman *et al.* [123] proposed a probabilistically motivated error metric.

D. EVENT DETECTION AND TRACKING

Event detection and tracking monitors documents or text streams, attempts to find new events or tracks existing events. In the TDT evaluation conference, four related tasks were defined: *new event detection*, *story link detection*, *topic detection* and *topic tracking*. The *new event detection* task monitors unseen events. Two scenarios are considered. First, participants are given a corpus and are required to divide it into different event-specific clusters. It is also named *retrospective event detection*. In the second scenario, known as *online event detection*, documents are arranged in chronological order. Participants are required to predict which event every document belongs to or which event contains a new event. The *story link detection* task estimates whether two given stories are discussing the same event. The *topic detection* task incrementally clusters stories that discuss the same event. In the *topic tracking* task, a system is first trained on a number of stories. Then, when a story arrives, the system ranks which story it belongs to [6].

In the pilot studies of the *new event detection* task, three systems were proposed: the CMU system, the dragon system and the UMass system. The CMU system adopted a single-pass incremental clustering algorithm (INCR) [5]. In this algorithm, every event is denoted by a centroid vector. It is induced from all document vectors in an event. The CMU system makes a flat partition of input documents. When a new document arrives, it is assigned to the nearest event. The dragon system also uses a single-pass clustering algorithm [93]. The first document is set as an initial event. Arrived documents are exclusively added into the closest event. If the similarity among existing events is larger than a predefined threshold, a new event is created by using the new document as a centroid vector. In this system, the threshold for creating a new event is dynamically calculated, which is helpful for capturing the temporal features of events [124]. The Umass system presented two models: a vector space model and a relevance model [92]. A characteristic of the system is that after documents are added, the centroid of an event can be updated, which enables it to adapt to the new documents.

In the event tracking task, a set of target events is given in advance. They are used to train a tracking system. It estimates the arrived documents and determines which events they belong to. Event tracking systems can be divided into two categories: unsupervised tracking and supervised tracking [94], [97]. The difference between them is that unsupervised tracking systems predict arrived documents independently without human intervention [5], [93]. On the other hand, in supervised tracking systems, after a document is processed, humans give a confidence score to the document. The information is used as feedback to adjust the tracking system Connell *et al.* [92], Zhang and Callan [94].

Supported by neural networks, event detection and tracking has been accelerated from many aspects. Hu *et al.* [125] presented an online news event detection model that adopts word embedding to represent documents. It reduces the influence caused by the out-of-vocabulary problem. Based on a deep neural network, Chen *et al.* [126] proposed an encoder-memory-decoder framework for subevent detection. It learns document and subevent representations. Subevents are detected by selecting the most proper subevent representation.

To evaluate an event detection and tracking system, a confusion matrix is commonly used. Two other techniques are the detection cost function and the decision error tradeoff (DET) curve. They are discussed in Fiscus and Doddington [127], Martin *et al.* [128].

E. EVENT ORGANIZATION

The output of event detection and tracking is a flattened partition of a corpus, where a document is not allowed to belong to multiple events. A flattening structure is not effective in supporting the multiple granularity of event detection and recognition. On the other hand, event organization gives a hierarchical view of documents. The task is similar to the *subject-based information organization* task proposed in the IR field [129], [130]. The main difference is that the later outputs a hierarchical representation but ignores the chronological property.

In the event organization task, the computational complexity is the main problem in organizing events into a hierarchical representation [92], [97]. To reduce the computational complexity, many models make use of the time locality phenomenon. Nallapati *et al.* [4] has shown that the temporal characteristic is very useful in event organization. From this aspect, Cutting *et al.* [131] provided a document hierarchical structure model (known as the scatter/gather model). This approach was also employed by the CMU event detection system [91]. There are also studies that generate hierarchical event-based social media, e.g., Twitter [132], [133]. The neural network-based methods for organizing events can also benefit from integrating deep semantic features and external knowledge. For example, Peng *et al.* [134] proposed a pairwise popularity graph convolutional network to integrate information from an external knowledge base for fine-grained social event categorization.

In the following, three typical approaches are discussed: the UMass approach [92], the TNO approach [97] and the ICT approach [135].

The UMass approach has two steps [92]. In the first step, all documents are sorted by timestamps. Each story is chronologically compared to a number of documents ahead of it. The *bounded k-NN* algorithm is implemented to calculate the similarity among them. If the similarity is below a predefined threshold, then a new event is created. Otherwise, it is assigned to the closest event. The algorithm outputs a list of events. Each contains one or more documents. In the second step, a *bounded agglomerative clustering* method is implemented to cluster the event list. Because the event list is sorted chronologically, it takes a certain number of events from the front of the list and combines the closest event pair with similarity. The process is iteratively run until only a specified number of events remain. However, even though the motivation of the UMass approach is reasonable, it suffers from poor performance. The reason is that, in this approach, documents are not allowed to belong to multiple events.

The TNO approach provides a scalable architecture for hierarchical event detection [97]. It has four steps: *sampling*, *clustering*, *optimizing* and *merging*. In the *sampling* step, random samples are collected from the corpus. In the *clustering*[13] step, documents are clustered by similarities among them. Then, the hierarchical structure is built with a *basic hierarchic agglomerative clustering* method. The *optimizing* step is necessary because the output of the clustering step is usually an unbalanced binary tree. Therefore, in the *merging* step, if a document has not been in the sample, it is used as a query to match the sampled documents. Then, the retrieved documents are added into the event that contains the top ten matched documents. TNO allows a document belonging to more than one event. The results showed better performance than the UMass approach.

The ICT approach was provided by Yu *et al.* [135]. It is similar to the UMass approach. In this approach, a whole corpus is first partitioned. A traditional clustering method is implemented layer by layer. Then, ICT uses a *bucket container* (a time window) to capture the time locality characteristic. In each bucket, a *agglomerative hierarchical clustering* method is implemented to produce microclusters and uses different thresholds to generate different granularities. Thresholds were used to control the depth of hierarchical events, where lower clusters are combined into one or more high layer clusters.

To evaluate a hierarchical event detection model, two problems are considered: detection cost and travel cost. The detection cost consists of a penalty for false alarms and detection misses. The travel cost refers to the expense of travelling through a hierarchical structure to find an optimal node. allan *et al.* [137] compared several approaches to evaluate a hierarchical event clustering system. They also proposed a minimal

[13] A symmetrical version of the *cross-entropy reduction scoring function* [136].

cost method, which was adopted by the TDT program to evaluate hierarchical event detection tasks.

IV. FRAME EVENT

The methodologies for recognizing frame events are mainly based on information extraction, where semantic units or linguistic units are identified from documents. Recognizing frame events is implemented with a cascading framework (or a multistage pipeline). The recognition task is usually divided into related subtasks, e.g., named entity recognition, coreference resolution and relation recognition. The cascading framework may suffer from a serious cascading failure problem, where errors are propagated into the following task. Another two frameworks are the joint approach and end-to-end approach. The joint approach combines two or more tasks in a single model. It can share features between tasks. For example, Fu *et al.* [138] extracted joint entities and relations using graph convolutional networks. The end-to-end approach implements a task from scratch without depending on other tasks. For example, the task of entity relation extraction is assumed to depend on the output of named entity extraction. In an end-to-end model, the relation is extracted from raw sentences [139].

In the following, definitions of frame events are first given. Then, the techniques used to support frame events are discussed. Finally, four subtasks for supporting frame event recognition are discussed individually and are named entity recognition, coreference resolution, relation recognition and event recognition.

A. DEFINITIONS AND NOTATIONS

In the Message Understanding Conference (MUC),[14] an event (also known as a "scenario template") is represented as a template. Each event has predefined parameters that refer to its semantic roles, e.g., actor, object, and time. Various event types were defined in the MUCs, e.g., fleet operations (MUC-1, MUC-2), terrorist activities (MUC-3, MUC-4), airplane crashes (MUC-7), and rocket/missile launches (MUC-7). The differences among the structures of events are great. At the MUC-2, an event has at most 10 slots. In the MUC-4, the structure becomes more complex. Some events contain more than 47 slots. In the MUC-5, nested structures are allowed for some events. They increase the challenge of recognizing structures.

ACE defines an "event" as "a specific occurrence involving participants". Compared to the MUC, ACE emphasizes language understanding. In the ACE definition, "triggers" (or anchors) are used to identify an occurrence of events. A trigger can be a verb, noun, pronoun, adjective, etc. In addition to participant roles (the MUC template slots), events in ACE have four properties[15] and two attributes.[16] Participants and attributes are generally known as event arguments. Detection of an event is triggered by the occurrence of an event trigger (words that can arouse the associated event). If an event has been detected, entities in the same event sentence are considered arguments of the event. Because event arguments can be asymmetric, the order of arguments should be considered.

As discussed in Section II-B, in the BioNLP, an event is defined as a *n*-tuple in a sentence. Event detection is set as a shared task [46]. The event type of BioNLP follows the GENIA corpus[17] [140]. Arguments of a BioNLP event are denoted as type t-entities, which represent occurrences of entities in a sentence. An argument is referred to as a two-dimensional array with its entity type and its span in the sentence. In the BioNLP annotation, every event or t-entity is labelled with a prefix "E" label and "T" label, respectively.

The *markup language for temporal and event expressions* (TimeML)[18] is a specification language for events and temporal expressions in natural language, where an event is considered "a cover term for situations that happen or occur" [141]. An event denotes a state or a circumstance. In TimeML, every event has a unique ID called "an event ID number" (eid). It belongs to one of the seven classes, e.g., REPORTING, PERCEPTION, and ASPECTUAL.

B. TECHNIQUES

The frame event recognition task is processed at the sentence level. Because only a limited number of features are available for making a prediction, it causes a sparse feature representation. Techniques used in this field are mainly focused on capturing the extra structural information and semantic information of sentences. To make better use of sentence structural information, various techniques are proposed.

N-gram techniques combine adjacent words to capture the dependency between them [142]. Because adjacent words may have no dependency relationship, the n-gram features are fragmental and noisy, especially when *n* is large. To improve the quality of n-gram features, the features are often combined with other information (e.g., latent topic variables) for robust performance [143]–[145].

Parsing trees or dependency trees is a fine-grained method to model sentence structure. It is rooted in linguistic theory and provides a formalized method to study languages. For example, in relation extraction, tree kernel methods are widely used to capture the sentence structural information [146], [147]. The problem in parsing tree-based systems is that their performance is often hurt by inaccurate chunking or parsing [148]. Tree-based systems usually suffer from poor performance caused by heterogeneous, noisy and fragmented data. Therefore, instead of a "deeper" analysis of a whole sentence, local dependencies are more helpful [148]. Another problem in tree kernel methods is that because the function

[14] http://www-nlpir.nist.gov/related_projects/muc/

[15] Polarity, tense, genericity and modality.

[16] Two attributes are *event-specific attributes* (e.g., CRIME-ARG, SENTENCE-ARG and POSITION-ARG) and *general event attributes* (e.g., PLACE-ARG and TIME-ARG).

[17] http://www.geniaproject.org/

[18] http://www.timeml.org/site/index.html

to calculate the distance between parsing trees is manually designed, it may overfit the evaluation corpus.

Sequence models (or Markov models, e.g., HMM, CRF, and LSTM) are widely used to capture the dependencies among terms. In sequence models, the task of information extraction is modelled as a tagging problem. It outputs a maximized label sequence. The label indicates the semantic role of a word. For example, a label "B-PER" indicates that a word is the beginning of a person's name. The sequence model is effective in capturing the structural information of a sentence. However, some tasks (e.g., coreference resolution) require processing two or more linguistic units, which may be scattered across a document. In this condition, it is hard to model them with a sequence model. Another problem with the sequence model is that the output is heavily dependent on the local features of a sentence [149]. Furthermore, finding a label sequence is not helpful for recognizing nested linguistic units (e.g., nested named entity).

Combined features are effective in capturing structural information [150]–[152]. Combined features map the term space into a higher dimensional space, which can make the separating hyperplane more effective. Combining features also changes the distribution of features. Combining features results in skewed term distributions and improves the predictive power of the features. Most of the studies on combined features use a greedy method, which applies new and additional features to improve the performance. Chen *et al.* [153] proposed a systematized analysis on the influence of combined features, where constraint conditions were employed to generate combined features. This analysis shows considerable improvement in relation recognition. Based on Chen *et al.* [153], Chen *et al.* [154] and Chen *et al.* [155] proposed a formalized method, named feature calculus, to combine "atomic features". Another regularized method to combine features is the kernel method. Kernel substitution is used to increase the dimensionality of a measure space [156].

In neural network-based methods, position embedding is the most popular method for capturing the structural information of a neural network. For example, Santos *et al.* [157] encoded position information into sentence representations. Implementing a neural network on a dependency parsing tree is another strategy for capturing the sentential information of a sentence. For example, Xu *et al.* [158] implemented an RNN on the shortest dependency path between two named entities. For other methods, Zhang *et al.* [159] implemented an attention mechanism on graph convolutional networks to select the relevant partial structures from a dependency tree. Kalchbrenner *et al.* [160] used a K-max pooling method to encode different parts of a sentence divided by two named entities. Chen *et al.* [161] presented a multichannel deep neural network to learn the structural information of a sentence. Soares *et al.* [162] inserted entity markers to point out entity boundaries in a relation instance. Then, a transformer network was adopted to learn the semantic dependency in the relation instances.

In addition to structural information, capturing the semantic information in sentences is also important for information extraction. One way to obtain semantic information is to use external knowledge. In open information extraction, external knowledge is widely used to guide a semisupervised method, e.g., regular expression [163], [164] or ontology [165]. Various resources have been explored, such as domain knowledge [27], [166], bilingual information [30] and heuristic information [167]. The main problem when applying external resources is that many of them are heterogeneous, which can hurt performance.

C. NAMED ENTITY RECOGNITION

An *entity* is considered an object (or a set of objects). An *entity mention* denotes an occurrence of an entity in a sentence. Recognizing entities is a challenging task because sentences also suffer from serious feature sparsity problems. This task is often formalized as a sequence labelling process. The labels can be "B", "I", "S", "L" and "O", representing the *Beginning*, *Single*, *Inside*, *Last* and *Outside* of an entity name, respectively. There are two ways to support this task.

The first approach makes an assumption of independent and identical distributions among labels. It labels every word independently without considering the dependency between adjacent labels. Therefore, a classifier can be employed to implement the labelling process, such as maximum entropy [168], [169] and support vector machines [170]. After labels for every word are given, three inference algorithms can be used to induce a maxized label sequence: greedy matching, beamsearch and dynamic programming [149].

The second approach assumes that the labels in a sequence are dependent. Then, instead of labelling each word independently, the approach attempts to find a maximized label sequence [156]. In this respect, a generative model (e.g., CRF [9]) or a discrimination method (e.g., HMM [8]) can be implemented to label sequences. In this field, neural networks have also received great attention. Early models usually adopted a sequence model to output the flattened NEs (e.g., LSTM, Bi-LSTM or Bi-LSTM-CNN).

Another important issue for named entity recognition is the nestification problem, where two named entities may overlap mutually. For example, "University of Washington" is an organizational NE, where "Washington" is a nested NE indicating the location of the university. To handle the nestification problem, the sequence model can be redesigned into three variants: layering, cascading and joint model [171]. Parsing trees are also widely used to represent nested NEs in a tree structure [172]. For example, Finkel and Manning [173] used the internal and structural information of parsing trees to flatten nested NEs. Zhang *et al.* [174] adopted a transition-based parser. Jie *et al.* [175] tried to capture the global dependency of parsing trees. Chen *et al.* [176] and Chen *et al.* [177] proposed a boundary assembling method that detects entity boundaries first. Then, boundaries are assembled into candidates for further prediction.

To recognize named entities, various features have been proposed [167], [169], [170]. Lexical features are the most popular. Combined features, such as n-gram features or phrase chunkers, are also widely adopted. To capture the structure of a sentence, POS tags and dependency trees have shown robust performance Chen *et al.* [178]. Because the task is processed at the sentence level, recognizing the named entity also suffers from the feature sparsity problem. External knowledge is important for capturing semantic information. Various external knowledge has been presented, e.g., gazetteer, thesaurus, WordNet [167], [170], [179]–[181], bilingual information [182], heuristic information (e.g., family or transliterated names) [167], [183] and titles [184]. Even outputs of another named entity recognzing system are helpful [169]. In the early stage, hand-crafted rules were used for entity recognition [185]–[187]. However, for some languages without morphological features (e.g., Chinese), rule-based methods are ineffective.

Many models recognize all types of entities in a unified framework. However, for the same languages (e.g., Chinese), different types of named entities may have different word formations. For example, location names usually have a nested structure. Therefore, there are studies that have addressed different types of named entities with different strategies [167], [181], [188]. Instead of a unified framework, [176] proposed a cascading approach, named the boundary assembling method, where boundaries of named entities are first detected and then combined as candidate entities for further prediction. It is an effective way to take advantage of existing features.

D. RELATION RECOGNITION

The works related to recognizing entity relations can be roughly divided into the following categories: traditional relation extraction, distant relation extraction, multi-entity relation extraction, cross sentence extraction and joint relation extraction.

Traditional relation extraction usually adopts supervised methods to identify the predefined relations between two named entities [189]. Techniques used for TRR often come from the field of machine learning, e.g., kernel methods [146], [190], belief networks [191], linear programming [192], maximum entropy [150], SVMs [193] or deep neural networks [194]. TRR systems can be divided into feature-based systems and kernel-based systems. Feature-based systems follow the notion of feature engineering, which tries to explore various syntactic and semantic features [150], [155], [193]. Kernel-based methods are usually formalized as a shallow parse classification problem [146]. Various tree kernels have been proposed for capturing semantic information and local dependency information, such as semantic tree kernels [195] and feature-enriched tree kernels [196]. The main disadvantage of a traditional relation extraction system is that a manually annotated corpus is required, which has a high labor cost. Furthermore, the migration among different applications is difficult.

In distant relation extraction, a knowledge base is employed to guide the process. It dynamically generates relation types from an open domain. From this perspective, Vashishth *et al.* [197] adopted a graph convolutional network to encode syntactic information. Qin *et al.* [198] proposed a deep reinforcement learning strategy to reduce the influence of false positive problems. To reduce noise caused by distant supervision, Yuan *et al.* [199] paid more attention to entity pairs with higher qualities. To address the incorrect samples generated by distant supervision, Zeng *et al.* [200] proposed a piecewise convolutional neural network that was implemented on different parts of a sentence divided by two named entities. Zhou *et al.* [201] proposed a hierarchical selective attention network that takes sentence-level attention and word-level attention to construct sentence representations for relation extraction.

Instead of setting two named entities as arguments of a relation, in multientity relation extraction (or known as an *n*-ary relation), a relation instance can be linked to several named entities. Techniques to extract multientity relations are often based on traditional relation extraction. For example, McDonald *et al.* [202] proposed a graph from pairs of entities. Then, maximal cliques in the graph are evaluated as potential complex relation instances. Mandya *et al.* [203] combined LSTM and a CNN to support multientity relation extraction. Peng *et al.* [204] presented a graph-based LSTM model incorporating various intrasentential and intersentiential dependencies for multientity relation extraction. Akimoto *et al.* [205] decomposed the *n*-ary relation extraction task into lower-arity candidates, which are aggregated as multientity relation instances.

Cross sentence extraction focuses on extracting entity relations at the document level [204], [206]. It is similar to the coreference resolution task, which tries to find referential relations among named entities in a document [207]. In this task, Wang *et al.* [208] proposed a cross-sentence context-aware approach to sense feature information across sentences. Yao *et al.* [209] integrated information across documents, which gives a global inference for relation extraction. Peng *et al.* [204] encoded the cross sentence relation with a graph-based LSTM model. Gupta *et al.* [210] proposed an intersentiential dependency-based neural network.

In a traditional relation extraction task, information about named entities (e.g., entity types) is supposed to be known in advance. In real applications, relation extraction may suffer from the error propagation problem caused by false named entities. Joint relation extraction implements the relation extraction task from scratch. To support this task, structured labels (jointed from relation and entity labels) can be designed to supervise the training process. For example, Miwa and Sasaki [211] proposed an entity and relation table, which jointly represents entities and relations. Zhang *et al.* [212] built a globally optimized neural model to integrate syntactic information. Another strategy for implementing joint relation extraction is to conduct a multiobjective learning model. For example, Zheng *et al.* [213] output relation labels and entity labels with a piecewise convolutional neural network and a bidirectional LSTM network, respectively. Fu *et al.* [138] extracted joint entities and relations using graph convolutional networks. Structured-type labels increase the number of classification categories. The multiobjective learning strategy has the advantage of sharing parameters with joint entities and relation extraction.

E. COREFERENCE RESOLUTION

When an entity is mentioned in a sentence, it is referred to as an *entity mention* (or a mention in short). An entity may have several mentions in a document (or documents). Coreference resolution is the task of grouping different entity mentions according to whether they refer to the same entity [214]. This task was first presented at the MUC-6. Then, it was evaluated as a separate coreference subtask. The coreference resolution can be seen as a coreference relation extraction task. The difference between them is that the coreference relation among entity mentions is transitive. It may link many entity mentions within a document.

The coreference resolution task is often processed as a binary relation recognition problem [215]. All pairs of entity mentions in a document should be evaluated. A high probability between two entity mentions indicates that they are references of the same entity. Then, pairwise mentions are grouped into the same set if they satisfy a predefined threshold. This approach is known as the pairwise decision model. It makes the assumption that all pairwise coreference decisions have an independent identical distribution.

Various techniques have been proposed for coreference resolution, such as decision trees [216], conditional models [215], hidden Markov models [217], bootstrapping [218], unsupervised models [219], [220], inductive logic programming [221], logic probability [222], [223], semantic models [224], nonparametric Bayesian models [225] and neural network models [226].

The main problem of coreference resolution is caused by transitivity constraints. Let m_i, m_k and m_j be three entity mentions. Suppose m_i, m_k are coreferent and m_j, m_k are coreferent. Because the coreference relation is transitive, according to the pairwise decision model, m_i, m_j and m_k should be grouped in the same set. However, it is common that m_i and m_j have a low probability in a given context, which leads to conflicts in a grouped entity mention set. An early approach for solving this problem was the ''ostrich strategy'', also known as the transitive closure of pairwise decisions. This approach assumes that m_i and m_j have a coreferent relation, regardless of the true value between them [217]. Nicolae [227] proposed a graph cutting approach to unit entity pairs into a coherent cluster.

Other problems for coreference resolution are the large search space and unbalanced data. Unlike relation recognition, which only considers the entity pairs in a sentence, coreference resolution should partition all mentions into mutually exclusive sets. All entity pairs in a document should be considered, where most of the entity pairs are not coreferent. This leads to a large search space and unbalanced data. One strategy is to adopt a greedy left-to-right linkage decision. Nonetheless, these approaches are greedy agglomerative clustering algorithms that rely on pairwise models. Coreference resolution also suffers from the feature sparsity problem. To utilize more features, Culotta [222] proposed a first-order probabilistic model, which enables classification decisions over a set of noun phrases. McCallum and Wellner [228] uses a generative approach to incorporate various features.

F. EVENT RECOGNITION

In this section, event recognition refers to tasks defined by the ACE evaluation conference,[19] where eight event types are defined, e.g., life, movement, and conflict. As discussed in Section IV-A, in the ACE annotation, event recognition is a very complex task. It is usually implemented as a pipeline process that is divided into four steps: anchor identification, argument identification, attribute assignment and event coreference. Tasks such as named entity recognition, relation recognition and coreference resolution are seen as fundamental tasks for supporting event recognition. To eliminate the influence of dependent tasks, when evaluating an event recognition model, dependent tasks are often given as golden annotations.

Currently, two approaches are widely adopted for recognizing events: *pattern matching* and *machine learning*. Yangarber [229] proposed a pattern matching-based system, named ''ExDisco''. In this system, ''good'' event examples are first discovered from a large unannotated corpus. They are used as seeds to initialize the system, and then more events (or examples) are found iteratively. In Chieu and Ng [230], a maximum entropy approach is adopted to extract events from semistructured and free text. Ahn [231] combined two approaches (TiMBL and MegaM) proposed by [232] and [233], and the combined approach showed improved performance. Riloff and Shoen [234] presented an AutoSlog-TS system that uses conceptual patterns for event recognition. The patterns are acquired automatically using only a preclassified training corpus. In Chen and Ji [235], event recognition is modelled as a sequence labeling process, where a maximum entropy Markov model is adopted. Llorens *et al.* [236] analyzed the contribution of semantic roles to TimeML event recognition and classification, which showed improved performance, especially for nominal events. Zhang *et al.* [237] proposed an emergency event recognition model based on deep learning. Ramakrishnan *et al.* [238] presented an EMBERS system that focuses on forecasting ''civil unrest'' events in the open domain. Piskorski *et al.* [11] presented an event recognition system that was used to monitor online news. In addition to extracting the parameters of events from sentences, they are extracted from clustered documents.

[19]http://www.itl.nist.gov/iad/mig/tests/ace/

V. GRAPHIC EVENT

A graphic event is a subgraph of a semantic network, which is usually built by distant (or weak) supervision [78], [79] and bootstrapping [80]–[83] methods. The graphic representation enables topological analyses and can give rise to novel solutions for many NLP tasks. Supported by a variety of visualization methods, it is also helpful in providing human-oriented information exploring interfaces.

A. DEFINITIONS AND NOTATIONS

Unlike documental events and frame events, no agreed definition has been proposed by any committee for graphic events. Various graphic event definitions have been used in this field. Some of them are discussed as follows.

An event is represented as a subgraph of a network. This is a popular definition for graphic events. In this definition, nodes of the network can be semantic units (e.g., named entities [12], [239]), real world entities [13] or social actors [240]. The edge represents the semantic relationship or correlation strength between nodes. Graphic event recognition is implemented by techniques such as dense subgraphs based on tightly coupled entities Angel *et al.* [13], community detection [13], [240], and graph partitioning [241].

An event is represented as a cluster of keywords [242], [243]. The definition is similar to that of the feature pivot event [133], [241], [244], where an event is a group of key words. It assumes that, for massive streams of documents, a sharp change in word frequency indicates a burst of a new event. Because a keyword-based event is composed of words expressing the meaning of an event, it is helpful in understanding the content of an event. For example, based on KeyGraph [245], many experiments have been conducted to support keyword-based event extraction [240], [242], [244], [246]. Texts in social media often have a short range of informative key words. With the increasing popularity of social media (e.g., Twitter, Weixun), keyword events have attracted increasing attention [247].

An event mention[20] is represented as a node of a network. The edge between events (nodes) may represent a chronological ordering Kuzey *et al.* [248] or an inclusion relationship representing different granularities of events [249]. The event is often extracted with document clustering or classification methods Kuzey *et al.* [248], Liu *et al.* [249]. The relations between events are measured by semantic metrics, e.g., cosine distance. This type of network is effective in expressing the relation between events. It gives a hierarchical (or chronological) representation of events, which is valuable in supporting event evolution and event tracking.

Other graphic event definitions have been proposed in this field. For example, Chen *et al.* [14] defined a graphic event as a semantic network extracted from a documental event. Glavaš and Šnajder [250] defined a graphic event as a temporal structure. In the EVIN system [251], a graphic event is defined by their participators, e.g., persons and organizations. In the NewsReader system [252], [253], a graphic event is a large knowledge graph built from a large number of articles and consists of dynamically extracted information about the who, what, where, and when of events.

[20] An event mention is a phrase denoting the occurrence of an event.

B. TECHNIQUES

The techniques used to construct a graphic event relate to many tasks in information extraction. According to the outputs, the tasks can be divided into three categories: node extraction, edge extraction and subgraph generation.

According to the type of nodes, many techniques are available. If a node is defined as a named entity (e.g., [12], [62], [239]) or an event mention (e.g., [250]), the process to recognize them is usually defined as a sequence labelling task. If a node is defined as a documental event, document clustering and classification are adopted [248], [249]. Compared to traditional document clustering tasks, extracting event mentions is often implemented in an open domain, where multisource, heterogeneous and noisy data (e.g., news, Twitter, BBS, microblog, etc.) are processed, which brings a big challenge to the traditional methods.

The techniques used to extract edges depend on the relation type between nodes. The technique used to extract the co-occurrence relation between nodes is the simplest. It can be obtained from documents directly (e.g., Das Sarma *et al.* [12]). The edge usually has a value, which indicates the correlation strength between nodes [13], [254], [255]. For the semantic relations between nodes, unsupervised or semisupervised methods are widely used [256], [257] and include heuristic rules [258], bootstrapping algorithms [80]–[82], distant supervision [78], [79], [259], and matrix factorization [260]. Because graphic events are usually constructed on an open domain, the performance of edge recognition is heavily dependent on the quality of node recognition. Furthermore, it may lead to the ‘‘semantic drift’’ problem caused by error accumulation [261]. The chronological order relation is mainly used in text streams, where the time stamp is usually available [248], [250]. For the subevents (an inclusion relationship) between nodes, the edges are commonly measured by a similarity function.

Community detection is an effective method for mining graphic events in a semantic network [13], [240], [242]. Based on a semantic network, community detection tries to identify groups of cohesive subgraphs. For example, based on an entity network built from social media, Angel *et al.* [13] proposed a ‘‘streaming edge weight’’ method to mine tightly coupled subgraphs. Because graphic events are often built on the open domain, they usually contain thousands of nodes and edges. Therefore, techniques such as social networks and complex networks can be adopted to mine the topological structures of a graphic event. For example, Chen *et al.* [14] and Meladianos *et al.* [243] generate subgraphs with techniques such as the k-core subgraph and the shortest path [262]. The chronological ordering relation is also helpful when tracking events. It consists of a sequence of events, which supports the detection of event evolution [263].

The similarity calculation between nodes can also be used to generate semantic cohesive subevents. For example, Liu *et al.* [249] used grouped key words to track growing events from massive breaking news.

C. SUBGRAPH MINING

Finding subgraphs is a key issue in mining knowledge from a graphic event. Because graphic events are variant in their structures, the task of subgraph mining is different for each event. In the following, several typical models are discussed in brief.

Accelerated by the development of social media (e.g., blog posts and twitters), textual streams are valuable in identifying emerging events. Angel *et al.* [13] presented a framework for finding groups of cohesive subgraphs in a weighted entity graph. In this graph, nodes represent real world entities. The weighted edges suggest pairwise association strength between nodes. In online scenarios, textual streams are arriving constantly. The cohesiveness of a subgraph can be changed by updating the graph. When the weight is updated, to maintain the cohesiveness of a dense subgraph, researchers proposed a D YND ENS algorithm. In this algorithm, when a single edge weight is updated, the maximum possible change in the graph is evaluated. In real-time event applications, this method is effective for updating dense subgraphs incrementally.

Based on a social network graph constructed from the World Wide Web, Faloutsos *et al.* [239] presented a paradigm to mine cohesive subgraphs for capturing the relationship between nodes. The method is implemented on an edge-weighted undirected graph with nodes denoting named entities. Recognized subgraphs are named ‘‘connection subgraphs’’. It satisfies predefined constraints (e.g., a limitation on the number of nodes). To process a large graph with high computational complexity, Faloutsos *et al.* [239] proposed an approximate (but high-quality) connection subgraph mining method.

Edges of a graphic event may represent the relationships between nodes. The semantic types of edges are often static and predefined. This limitation disables the capacity for discovering emerging events with new semantic relations. Because nodes and the edges between them are recognized automatically, it also suffers from poor performance caused by heterogeneous data. In regard to these problems, Das Sarma *et al.* [12] presented an entity dynamic relation graph. In this graph, the edge represents the dynamic connections between entities. Recognizing the dynamic relation is based on the co-occurrence between named entities. This graph uses a time window to construct the temporal profile. When an emerging event is broken out, the temporal profile represents the ratio of the co-occurrence. Based on this graph, the authors proposed temporal constraint cluster methods to identify dynamic events. They are represented as subgraphs of a large relation graph with temporal constraints.

Zhao and Mitra [240] defined events as subgraphs identified from a hierarchical graphic representation. Event detection is divided into three steps. First, social text stream data (emails in this article) are clustered into topics with unsupervised methods. The email content is represented as a TF-IDF vector. Then, a similarity function is defined to cluster them. The output of the cluster is seen as the top level structure of the graphic representation. Second, the timestamps of the emails are used to partition the clustered topics into finer granularities (subtopics) to capture the chronological information. A subtopic is a set of emails with information regarding email addresses (senders and recipients). By setting the addresses as nodes and the communication relations as edges, the whole structure is represented as a hierarchical multigraph. Therefore, in the third step, an event is identified as a connected subgraph with topic and temporal constraints.

D. KEYWORDS MINING

Every social text stream has a timestamp. It can be used to detect a burst of data flow in a social network, which may indicate an emerging event. Because of the popularity of social media, such as weblogs, Twitter, and message boards, keyword-based methods for recognizing events have received extensive research attention. Representing an event as a set of keywords can reveal the content of an event directly, which is helpful to understand the event. Because these words are the key to expressing the meanings of a social media text, recognizing events from keywords is also useful in reducing the influence of noise and fragmental data in social text media.

Weng and Lee [241] presented an event detection paradigm, which is rooted in signal processing theory. In this model, occurrences of a word with its timeline are transformed into a signal by wavelet analysis. It has the ability to represent the frequency change in a word over time [264]. In this work, trivial words are filtered by a cross correlation algorithm, which denotes the similarities among signals. This algorithm is implemented on all pairs of signals (words). It generates a similarity matrix. The matrix can be transformed into a graphic representation. Then, the task of detecting events is modelled as a graph partitioning problem. Following this motivation, Dong *et al.* [265] presented a multiscale event detection method based on the properties of the wavelet analyses. It automatically captures the temporal and spatial information of events.

Parikh and Karlapalem [266] proposed an algorithm for extracting events from KeyGraph. In this work, the TF-IDF term weighting approach is used to collect a set of keywords from documents. Keywords are the nodes of KeyGraph. The edges between nodes are defined as the co-occurrence relation in documents. To obtain a robust graph quality, edges are removed if the co-occurring nodes have a conditional probability below a predefined threshold. Based on KeyGraph, community detection is implemented to detect events. In this process, a betweenness centrality score is computed from each edge. It indicates the number of the shortest paths passing through the KeyGraph. A high score indicates that it is an inter-community edge. Removing it can reduce the

TABLE 1. Summary of Event Detection and Recognition.

	Documental Event	Frame Event	Graphic Event
Definition	A cluster of documents describing the same event.	A frame or a template with slots to be filled.	A subgraph of a semantic graph.
Relative Tasks	1) Segmentation; 2) new event detection; 3) event tracking; 4) event organization;	1) Named entity recognition; 2) relation recognition; 3) coreference resolution; 4) event extraction;	1) Subgraph representation; 2) keyword representation; 3) supplemental representation
Methods	Focusing on document-level processing. Mainly based on vector space models. Two types of techniques are employed: term weighting (TF, IDF, TF-IDF, term relevance weight, etc.) and space transformation (LSA, PLSA and LDA, etc.)	Focusing on sentence-level processing. Sequence models are often employed to capture syntactic and semantic information. Many techniques are employed: n-gram, parse tree, dependency tree, HMM, CRF, tree kernel, LSTM, etc.	Mainly based on graphic representation. Topology-based analytical methods are employed: community detection, statistical relational learning, social network, complex network, graph mining algorithms, etc.
System Examples	Cieri et al. [90], Abujabal and Berberich [267], Zhang and Callan [94], Yang et al. [5], Beeferman et al. [117], Bishop and Nasrabadi [156], Kozima [268], Papka et al. [124], Connell et al. [92], Trieschingg and Kraaij [97], Fiscus and Doddington [127], Martin et al. [128], Jurafsky and Martin [129], Nallapati et al. [4], Cutting et al. [131], Ifrim et al. [132], Atefeh and Khreich [133], Allan et al. [137]	Yangarber [229], Chieu and Ng [230], Ahn [231], Daelemans et al. [232], Daume [233], Riloff and Shoen [234], Chen and Ji [235], Llorens et al. [236], Zhang et al. [237], Ramakrishnan et al. [238], Piskorski et al. [11], Hoffmann et al. [87], Brin [163], Curran et al. [261], Agichtein and Gravano [83], Fan et al. [269], Grave [270], Fader et al. [271], Florian et al. [169]	Das Sarma et al. [12], Faloutsos et al. [239], Angel et al. [13], Zhao and Mitra [240], Weng and Lee [241], Sayyadi et al. [242], Meladianos et al. [243], Atefeh and Khreich [133], Knotostathis et al. [244], Ohsawa et al. [245], Kleinberg [246], Kuzey et al. [248], Liu et al. [249], Chen et al. [14], Glavaš and Šnajder [250], Kuzey and Weikum [251], Vossen et al. [252], Vossen et al. [253], Ritter et al. [272]

relations between nodes located in different communities. A node duplication strategy is proposed to ensure that a keyword can be seen in different communities. After events are detected, document clustering can be implemented by using a similarity function defined between an event (a set of keywords) and a document.

Meladianos *et al.* [243] used a graph of words to support event detection. Instead of a sliding window to count the co-occurrence, this graph weights edges by dividing the length of the document (Twitter). The graph-of-words is decomposed into a k-score subgraph, which represents the cohesion of words. Each node in the graph is given a value, named the ''core number'', indicating the largest of k that belongs to the k-core. Then, event detection based on the graph-of-words is implemented as follows. Successive tweets within an interval are collected to build the graph-of-words. ''Random'' tweets will lead to a graph-of-words with low core numbers. An event is detected if the highest cores in the graph-of-words satisfy specified requirements.[21]

E. GRAPH INFERRING

Most graphic events are semantic networks, where nodes are concepts and edges represent the semantic relations between them. Another type of graphic event is known as a narrative event graph. Compared with a narrative event graph, a semantic network can be seen as a ''static'' graph. It represents the connections among knowledge. In a narrative event graph, the node is usually denoted as an activity, and the edge represents the temporal and causal relations between activities. Therefore, it can be used to represent the development of a narrative event. Because the narrative event graph contains internal logical connections among narrative events, it can be adopted to support event evolutionary analysis.

A narrative event chain is a typical narrative graph proposed by Chambers and Jurafsky [273]. It is defined as a partially ordered set of events central to a protagonist. In this representation, the definition of events follows Schank and Abelson [274], known as scripts or Fillmorean frames. Events are instantiated from texts with the Stanford parser to collect verbs with subject, object, or prepositional-type dependencies. The narrative relation between events is detected by unsupervised distributional methods. The quality of the constructed narrative event chain is improved by classifying partially ordered events and pruning self-contained chains. In this narrative graphic event, Chambers and Jurafsky [273] presented two tasks to support inference: a narrative cloze task and an order coherence task. The narrative cloze task predicates an event, which is randomly removed from a narrative event chain. This task can evaluate a system's knowledge of narrative relations and coherence. It uses pointwise mutual information to generate a ranked list of guesses. The order coherence task infers the partial temporal ordering of the events. Unlike temporal relation recognition, which tries to identify the relation between two events, the order coherence task emphasizes the coherence of a whole ordered narrative chain.

[21] The sum of the core numbers of d terms belonging to the highest cores exceed a predefined threshold.

Du *et al.* [275] proposed a recurrent marked temporal point process model to support event forecasting. This model uses a recurrent neural network to learn a representation of the nonlinear dependency of an event chain, which is used for predicting the next event. Trivedi *et al.* [276] proposed a deep learning architecture to model knowledge evolution and reason over dynamic knowledge graphs. For another approach, Kazeni *et al.* [277] conducted a survey about representation learning for dynamic graphs.

VI. CONCLUSION

In this article, we provide a comprehensive survey about textual event detection and recognition. The methodologies for building events for information exploration are divided into three types: documental events, frame events and graphic events. They are summarized in TABLE1.

Documental events are processed at the document level, where the document has the smallest granularity. A document is denoted as a vector. Then, the similarity between documents is computed as the distance between vectors. Recognizing documental events is based on the similarity among documents. Many effective tasks, e.g., new event detection and event tracking, have been defined to support documental event-based information retrieval. It provides a convenient approach to retrieve relevant documents. The main problem for documental event detection is that it does not support content-based retrieval, where no syntactic or semantic units contained in documents are identified. Users are required to skim through the contents of documents returned by systems.

A frame event is defined as a frame or a template with slots to be filled by event arguments. It is processed at the sentence level. The recognition of frame events is rooted in information extraction, which aims to extract linguistic units with concrete concepts or functions (e.g., named entities, relations, or quantifiers). It can be seen as a trade-off between information retrieval and text understanding. The output of frame event recognition can be seen as "knowledge" to populate a knowledge base directly. It is helpful in providing content-based information exploration. The main problem in frame event systems is that the performance of frame event recognition is still unsatisfactory.

Many systems automatically organize extracted linguistic units into graph-based representations. A graphic event is a subgraph of the representation. Event detection is implemented by mining cohesive subgraphs from the graph. Organizing linguistic units into a graphic event enables topological techniques such as social network analysis, complex network analysis, and statistical relation learning. They are very useful for revealing the underlying structure of documents.

REFERENCES

[1] M. Sanderson and W. B. Croft, "The history of information retrieval research," *Proc. IEEE*, vol. 100, no. Special Centennial Issue, pp. 1444–1451, May 2012.

[2] G. Salton and C. S. Yang, "On the specification of term values in automatic indexing," *J. Documentation*, vol. 29, no. 4, pp. 351–372, Apr. 1973.

[3] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Commun. ACM*, vol. 18, no. 11, pp. 613–620, Nov. 1975.

[4] R. Nallapati, A. Feng, F. Peng, and J. Allan, "Event threading within news topics," in *Proc. CIKM*, 2004, pp. 446–453.

[5] Y. Yang, J. G. Carbonell, R. D. Brown, T. Pierce, B. T. Archibald, and X. Liu, "Learning approaches for detecting and tracking news events," *IEEE Intell. Syst.*, vol. 14, no. 4, pp. 32–43, Jul. 1999.

[6] J. Allan, J. G. Carbonell, G. Doddington, J. Yamron, and Y. Yang, "Topic detection and tracking pilot study final report," in *Proc. DARPA Broadcast News Transcription Understand. Workshop*, 1998, pp. 194–218.

[7] G. Doddington, A. Mitchell, M. Przybocki, L. Ramshaw, S. Strassel, and R. Weischedel, "The automatic content extraction (ace) program–tasks, data, and evaluation," in *Proc. LREC*, 2004, pp. 837–840.

[8] K. Seymore, A. McCallum, and R. Rosenfeld, "Learning hidden Markov model structure for information extraction," in *Proc. AAAI*, 1999, pp. 37–42.

[9] J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. 18th ICML.* San Francisco, CA, USA: Morgan Kaufmann, 2001, pp. 282–289.

[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] J. Piskorski, H. Tanev, M. Atkinson, E. Van Der Goot, and V. Zavarella, "Online news event extraction for global crisis surveillance," in *Transactions on Computational Collective Intelligence V*. Cham, Switzerland: Springer, 2011, pp. 182–212.

[12] A. Das Sarma, A. Jain, and C. Yu, "Dynamic relationship and event discovery," in *Proc. 4th ACM Int. Conf. Web Search Data Mining (WSDM)*, 2011, 207–216.

[13] A. Angel, N. Sarkas, N. Koudas, and D. Srivastava, "Dense subgraph maintenance under streaming edge weight updates for real-time story identification," *Proc. VLDB Endowment*, vol. 5, no. 6, pp. 574–585, Feb. 2012.

[14] Y. Chen, Q. Zheng, F. Tian, H. Lliu, Y. Hao, and N. Shah, "Exploring open information via event network," *Natural Lang. Eng.*, vol. 24, no. 2, pp. 199–220, 2017.

[15] P. Rozenshtein and A. Gionis, "Mining temporal networks," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 3225–3226.

[16] A. Singhal, "Modern information retrieval: A brief overview," *IEEE Comput. Soc. Tech. Committee Data Eng.*, vol. 24, no. 4, pp. 35–43, Jan. 2001.

[17] R. Baeza-Yates, *Modern Information Retrieval*, vol. 463. New York, NY, USA: ACM Press, 1999.

[18] S. Eliot and J. Rose, *A Companion to the History of the Book*, vol. 98. Hoboken, NJ, USA: Wiley, 2009.

[19] C. N. Mooers, "The theory of digital handling of non-numerical information and its implications to machine economics," Zator Co., Cambridge, MA, USA, Tech. Bull. No. 48, 1950.

[20] C. W. Cleverdon and M. Keen, *Aslib Cranfield Research Project-Factors Determining the Performance of Indexing Systems; Test Results*, vol. 2. Cranfield, U.K.: College of Aeronautics, 1966, p. 299. [Online]. Available: http://hdl.handle.net/1826/863

[21] D. Harman, "Overview of the first TREC conference," in *Proc. 16th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 1993, pp. 36–47.

[22] E. Greengrass, "Information retrieval: A survey," Tech. Rep., 2000.

[23] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Inf. Retr.*, vol. 1., Cambridge university press Cambridge, 2008.

[24] W. R. Greiff, W. B. Croft, and H. Turtle, "Computationally tractable probabilistic modeling of Boolean operators," in *Proc. 20th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 1997, pp. 119–128.

[25] E. D. Liddy, "Enhanced text retrieval using natural language processing," *Bull. Amer. Soc. Inf. Sci. Technol.*, vol. 24, no. 4, pp. 14–16, Jan. 2005.

[26] C. Carpineto and G. Romano, "A survey of automatic query expansion in information retrieval," *ACM Comput. Surveys*, vol. 44, no. 1, pp. 1–50, Jan. 2012.

[27] N. Ding, Z. Li, Z. Liu, H. Zheng, and Z. Lin, "Event detection with trigger-aware lattice neural network," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 347–356.

[28] W. Lu and T. H. Nguyen, "Similar but not the same: Word sense disambiguation improves event detection via neural representation matching," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 4822–4828.
[29] J. Walker Orr, P. Tadepalli, and X. Fern, "Event detection with neural networks: A rigorous empirical evaluation," 2018, *arXiv:1808.08504*. [Online]. Available: http://arxiv.org/abs/1808.08504
[30] J. Liu, Y. Chen, K. Liu, and J. Zhao, "Event detection via gated multilingual attention mechanism," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 1–8.
[31] H. Ji and R. Grishman, "Knowledge base population: Successful approaches and challenges," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1, Jun. 2011, pp. 1148–1158.
[32] R. Grishman, "Information extraction: Techniques and challenges," in *Information Extraction A Multidisciplinary Approach to an Emerging Information Technology*. Cham, Switzerland: Springer, 1997, pp. 10–27.
[33] R. Gaizauskas and Y. Wilks, "Information extraction: Beyond document retrieval," *J. Documentation*, vol. 54, no. 1, pp. 70–105, Mar. 1998.
[34] M.-F. Moens, *Information Extraction: Algorithms and Prospects in a Retrieval Context*, vol. 21. Cham, Switzerland: Springer, 2006.
[35] J. Cowie and W. Lehnert, "Information extraction," *Commun. ACM*, vol. 39, no. 1, pp. 80–91, 1996.
[36] A. McCallum, "Information extraction: Distilling structured data from unstructured text," *Queue*, vol. 3, no. 9, pp. 48–57, Nov. 2005.
[37] R. C. Schank, "Conceptual dependency: A theory of natural language understanding," *Cognit. Psychol.*, vol. 3, no. 4, pp. 552–631, Oct. 1972.
[38] G. DeJong, "Skimming newspaper stories by computer," in *Proc. IJCAI*. San Mateo, CA, USA: Morgan Kaufmann, 1977, p. 16.
[39] M. Minsky, "A framework for representing knowledge," in *The Psychology of Computer Vision*, P. Winston, Ed. New York, NY, USA: McGraw-Hill, 1974.
[40] W. G. Lehnert, "Plot units and narrative Summarization*," *Cognit. Sci.*, vol. 5, no. 4, pp. 293–331, Oct. 1981.
[41] D. E. Rumelhart, "Notes on a schema for stories," in *Representation and Understanding: Studies in Cognitive Science*, vol. 211, D. Bobrow and A. Collins, Eds. New York, NY, USA: Academic, 1975, p. 236.
[42] N. Sager, *Natural Language Information Processing* (Advanced Book Program). Reading, MA, USA: Addison-Wesley, 1981.
[43] U. Hahn, "Making understanders out of parsers: Semantically driven parsing as a key concept for realistic text understanding applications," *Int. J. Intell. Syst.*, vol. 4, no. 3, pp. 345–393, Sep. 1989.
[44] J. R. Hobbs and E. Riloff, "Information extraction," in *Handbook of NLP*, vol. 2. London, U.K.: Chapman & Hall, 2010, pp. 511–532.
[45] I. Hendrickx, S. N. Kim, Z. Kozareva, P. Nakov, D. Ó Séaghdha, S. Padó, M. Pennacchiotti, L. Romano, and S. Szpakowicz, "SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals," in *Proc. Workshop Semantic Eval., Recent Achievements Future Directions (DEW)*, 2009, pp. 94–99.
[46] J.-D. Kim, T. Ohta, S. Pyysalo, Y. Kano, and J. Tsujii, "Overview of BioNLP'09 shared task on event extraction," in *Proc. Workshop BioNLP Shared Task (BioNLP)*, 2009, pp. 1–9.
[47] C. Nédellec, R. Bossy, J.-D. Kim, J.-J. Kim, T. Ohta, S. Pyysalo, and P. Zweigenbaum, "Overview of bionlp shared task 2013," in *Proc. BioNLP Shared Task Workshop*, 2013, pp. 1–7.
[48] J.-D. Kim, S. Pyysalo, T. Ohta, R. Bossy, N. Nguyen, and J. Tsujii, "Overview of bionlp shared task 2011," in *Proc. BioNLP Shared Task Workshop*, Association for Computational Linguistics, 2011, pp. 1–6.
[49] D. Li, L. Huang, H. Ji, and J. Han, "Biomedical event extraction based on knowledge-driven Tree-," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1, 2019, pp. 1421–1430.
[50] J. Jourde, A.-P. Manine, P. Veber, K. Fort, R. Bossy, E. Alphonse, and P. Bessieres, "Bionlp shared task 2011: Bacteria gene interactions and renaming," in *Proc. BioNLP Shared Task Workshop*, Association for Computational Linguistics, 2011, pp. 65–73.
[51] R. F. I. Cancho and R. V. Solé, "Zipf's law and random texts," *Adv. Complex Syst.*, vol. 5, no. 1, pp. 1–6, 2002.
[52] L. Leydesdorff and L. Vaughan, "Co-occurrence matrices and their applications in information science: Extending ACA to the Web environment," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 57, no. 12, pp. 1616–1628, 2006.
[53] H. D. White and K. W. McCain, "Visualizing a discipline: An author co-citation analysis of information science," *J. Amer. Soc. Inf. Sci.*, vol. 49, no. 4, pp. 327–355, 1998.
[54] P. Edmonds, "Choosing the word most typical in context using a lexical co-occurrence network," in *Proc. EACL*, 1997, pp. 507–509.
[55] R. F. I. Cancho and R. V. Solé, "The small world of human language," *Proc. Royal Soc. London B, Biol. Sci.*, vol. 268, no. 1482, pp. 2261–2265, Nov. 2001.
[56] J. Liu, Y. Chen, K. Liu, and J. Zhao, "Neural cross-lingual event detection with minimal parallel resources," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 738–748.
[57] K. W. Church and P. Hanks, "Word association norms, mutual information, and lexicography," *Comput. Linguistics*, vol. 16, no. 1, pp. 22–29, Mar. 1990.
[58] A. Kilgarriff, "Language is never, ever, ever, random," *Corpus Linguistics Linguistic Theory*, vol. 1, no. 2, pp. 263–276, Jan. 2005.
[59] R. Ferrer i Cancho, R. V. Solé, and R. Köhler, "Patterns in syntactic dependency networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, no. 5, May 2004, Art. no. 051915.
[60] H. Liu and F. Hu, "What role does syntax play in a language network?" *EPL (Europhysics Lett.)*, vol. 83, no. 1, p. 18002, Jul. 2008.
[61] R. Ferrer I Cancho, A. Capocci, and G. Caldarelli, "Spectral methods cluster words of the same class in a syntactic dependency network," *Int. J. Bifurcation Chaos*, vol. 17, no. 7, pp. 2453–2463, Jul. 2007.
[62] T. H. Nguyen and R. Grishman, "Graph convolutional networks with argument-aware pooling for event detection," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 5900–5907.
[63] S. Cui, B. Yu, T. Liu, Z. Zhang, X. Wang, and J. Shi, "Edge-enhanced graph convolution networks for event detection with syntactic relation," 2020, *arXiv:2002.10757*. [Online]. Available: http://arxiv.org/abs/2002.10757
[64] A. E. Motter, A. P. S. de Moura, Y.-C. Lai, and P. Dasgupta, "Topology of the conceptual network of language," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 65, no. 6, Jun. 2002, Art. no. 065102.
[65] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: A collaboratively created graph database for structuring human knowledge," in *Proc. ACM SIGMOD Int. Conf. Manage. Data (SIGMOD)*, 2008, pp. 1247–1250.
[66] D. B. Lenat, "CYC: A large-scale investment in knowledge infrastructure," *Commun. ACM*, vol. 38, no. 11, pp. 33–38, Nov. 1995.
[67] G. A. Miller, "WordNet: A lexical database for english," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995.
[68] M. Masterman, "Semantic message detection for machine translation, using an interlingua," in *Proc. Int. Conf. Mach. Transl. Lang. Appl. Lang. Anal.*, 1961, pp. 437–475.
[69] S. Ceccato and E. Maretti, "Linguistic analysis and programming for mechanical translation (mechanical translation and thought)," Feltrinelli New York, New York, NY, USA, Tech. Rep., 1960.
[70] J. F. Sowa, *Conceptual Structures: Information Processing in Mind and Machine*. Reading, MA, USA: Addison-Wesley, 1984.
[71] D. G. Hays, "Dependency theory: A formalism and some observations," *Language*, vol. 40, no. 4, pp. 511–525, 1964.
[72] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: A core of semantic knowledge," in *Proc. 16th Int. Conf. World Wide Web (WWW)*, 2007, pp. 697–706.
[73] C. Matuszek, J. Cabral, M. J. Witbrock, and J. DeOliveira, "An introduction to the syntax and content of cyc.," in *Proc. AAAI Spring Symp.*, 2006, pp. 44–49.
[74] L. Padró, Z. Agic, X. Carreras, B. Fortuna, E. García Cuesta, Z. Li, T. Stajner, and M. Tadic, "Language processing infrastructure in the Xlike project," in *Process. LREC*, 2014, pp. 3811–3816.
[75] M. Rospocher, M. van Erp, P. Vossen, A. Fokkens, I. Aldabe, G. Rigau, A. Soroa, T. Ploeger, and T. Bogaard, "Building event-centric knowledge graphs from news," *J. Web Semantics*, vols. 37–38, pp. 132–151, Mar. 2016.
[76] S. Zhang, K. Duh, and B. Van Durme, "MT/IE: Cross-lingual open information extraction with neural sequence-to-sequence models," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics, Short Papers*, vol. 2, 2017, p. 64.
[77] G. Angeli, M. J. Johnson Premkumar, and C. D. Manning, "Leveraging linguistic structure for open domain information extraction," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Lang. Process. Long Papers*, vol. 1, 2015, pp. 344–354.

[78] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, "Distant supervision for relation extraction without labeled data," in *Proc. Joint Conf. 47th Annu. Meeting ACL 4th Int. Joint Conf. Natural Lang. Process. (ACL-IJCNLP)*, vol. 2, 2009, pp. 1003–1011.

[79] Y. Liu, K. Liu, L. Xu, and J. Zhao, "Exploring fine-grained entity type constraints for distantly supervised relation extraction," in *Proc. 25th Int. Conf. Comput. Linguistics*, 2014, pp. 2107–2116.

[80] Z. Kozareva and E. Hovy, "Learning arguments and supertypes of semantic relations using recursive patterns," in *Proc. ACL*, 2010, pp. 1482–1491.

[81] T. McIntosh, L. Yencken, J. Curran, and T. Baldwin, "Relation guided bootstrapping of semantic lexicons," in *Proc. ACL*, 2011, pp. 266–270.

[82] D. S. Weld, R. Hoffmann, and F. Wu, "Using Wikipedia to bootstrap open information extraction," *ACM SIGMOD Rec.*, vol. 37, no. 4, pp. 62–68, Mar. 2009.

[83] E. Agichtein and L. Gravano, "Snowball: Extracting relations from large plain-text collections," in *Proc. DL*, 2000, pp. 85–94.

[84] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni, "Open information extraction for the Web," in *Proc. IJCAI*, vol. 7, 2007, pp. 2670–2676.

[85] O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates, "Unsupervised named-entity extraction from the Web: An experimental study," *Artif. Intell.*, vol. 165, no. 1, pp. 91–134, 2005.

[86] O. Etzioni, A. Fader, J. Christensen, S. Soderland, and M. Mausam, "Open information extraction: The second generation," in *Proc. IJCAI*, vol. 11, 2011, pp. 3–10.

[87] R. Hoffmann, C. Zhang, and D. S. Weld, "Learning 5000 relational extractors," in *Proc. ACL*, 2010, pp. 286–295.

[88] J. Zhu, Z. Nie, X. Liu, B. Zhang, and J.-R. Wen, "Statsnowball: A statistical approach to extracting entity relationships," in *Proc. 18th Int. Conf. World Wide Web*, 2009, pp. 101–110.

[89] J. Fiscus, G. Doddington, J. Garofolo, and A. Martin, "Nist's 1998 topic detection and tracking evaluation (TDT2)," in *Proc. DARPA Broadcast News Workshop*, 1999, pp. 19–24.

[90] C. Cieri, S. Strassel, D. Graff, N. Martey, K. Rennert, and M. Liberman, "Corpora for topic detection and tracking," in *Topic Detection Tracking*. Cham, Switzerland: Springer, 2002, pp. 33–66.

[91] Y. Yang, T. Pierce, and J. Carbonell, "A study of retrospective and on-line event detection," in *Proc. 21st Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 1998, pp. 28–36.

[92] M. Connell, A. Feng, G. Kumaran, H. Raghavan, C. Shah, and J. Allan, "Umass at TDT 2004," in *Proc. Topic Detection Tracking Workshop Rep.*, 2004, pp. 2–3.

[93] J. Allan, R. Papka, and V. Lavrenko, "On-line new event detection and tracking," in *Proc. SIGIR*, 1998, pp. 37–45.

[94] Y. Zhang and J. Callan, "CMU dir supervised tracking report," in *Proc. DARPA Workshop*, 2004.

[95] J. Allan, V. Lavrenko, and H. Jin, "First story detection in TDT is hard," in *Proc. 9th Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2000, pp. 374–381.

[96] T. Brants and F. Chen, "A system for new event detection," in *Proc. 26th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2003, pp. 330–337.

[97] D. Trieschnigg and W. Kraaij, "TNO hierarchical topic detection report at TDT 2004," in *Proc. Topic Detection Tracking Workshop Rep.*, 2004, vol. 99, no. 7, pp. 1–8.

[98] S. Duan, R. He, and W. Zhao, "Exploiting document level information to improve event detection via recurrent neural networks," in *Proc. 8th Int. Joint Conf. Natural Lang. Process.*, vol. 1, Nov. 2017, pp. 352–361

[99] W. B. Croft, S. Cronen-Townsend, and V. Lavrenko, "Relevance feedback and personalization: A language modeling perspective," in *Proc. DELOS Workshop, Personalisation Recommender Syst. Digit. Libraries*, vol. 3, 2001, pp. 49–54.

[100] V. Lavrenko, J. Allan, E. DeGuzman, D. LaFlamme, V. Pollard, and S. Thomas, "Relevance models for topic detection and tracking," in *Proc. 2nd Int. Conf. Hum. Lang. Technol. Res.* San Mateo, CA, USA: Morgan Kaufmann, 2002, pp. 115–121.

[101] W. B. Frakes and R. Baeza-Yates, *Information Retrieval: Data Structures and Algorithms*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1992.

[102] A. Aizawa, "An information-theoretic perspective of TF–IDF measures," *Inf. Process. Manage.*, vol. 39, no. 1, pp. 45–65, Jan. 2003.

[103] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Inf. Process. Manage.*, vol. 24, no. 5, pp. 513–523, Jan. 1988.

[104] H. P. Luhn, "A statistical approach to mechanized encoding and searching of literary information," *IBM J. Res. Develop.*, vol. 1, no. 4, pp. 309–317, Oct. 1957.

[105] K. Sparck Jones, "A statistical interpretation of term specificity and its application in retrieval," *J. Documentation*, vol. 28, no. 1, pp. 11–21, Jan. 1972.

[106] S. E. Robertson and K. S. Jones, "Relevance weighting of search terms," *J. Amer. Soc. Inf. Sci.*, vol. 27, no. 3, pp. 129–146, May 1976.

[107] H. I. M. Nagao and M. Mizutani, "An automated method for the extraction of important words from Japanese scientific documents," in *Proc. Trans. Inf. Process. Soc. Jpn.*, 1976, vol. 17, no. 2, p. 110.

[108] T. Dunning, "Accurate methods for the statistics of surprise and coincidence," *Comput. Linguistics*, vol. 19, no. 1, pp. 61–74, 1993.

[109] G. Grefenstette, *Explorations in Automatic Thesaurus Discover*. Boston, MA, USA: Kluwer, 1994.

[110] T. K. Landauer and S. T. Dumais, "A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge.," *Psychol. Rev.*, vol. 104, no. 2, p. 211, 1997.

[111] T. Hofmann, "Unsupervised learning by probabilistic latent semantic analysis," *Mach. Learn.*, vol. 42, no. 1, pp. 177–196, Jan. 2001.

[112] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.

[113] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman, "Indexing by latent semantic analysis," *JASIS*, vol. 41, no. 6, pp. 391–407, 1990.

[114] T. Hofmann, "Probabilistic latent semantic indexing," in *Proc. 22nd Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 1999, pp. 50–57.

[115] S. Zheng, W. Cao, W. Xu, and J. Bian, "Doc2EDAG: An end-to-end document-level framework for chinese financial event extraction," 2019, *arXiv:1904.07535*. [Online]. Available: http://arxiv.org/abs/1904.07535

[116] M. Steyvers and T. Griffiths, "Probabilistic topic models," in *Handbook of LSA*, vol. 427, no. 7. 2007, pp. 424–440.

[117] D. Beeferman, A. Berger, and J. Lafferty, "Statistical models for text segmentation," *Mach. Learn.*, vol. 34, nos. 1–3, pp. 177–210, 1999.

[118] X. Du and C. Cardie, "Document-level event role filler extraction using multi-granularity contextualized encoding," 2020, *arXiv:2005.06579*. [Online]. Available: http://arxiv.org/abs/2005.06579

[119] A. McCallum, D. Freitag, and F. C. Pereira, "Maximum entropy Markov models for information extraction and segmentation.," in *Proc. ICML*, 2000, pp. 591–598.

[120] J. Barrow, R. Jain, V. Morariu, V. Manjunatha, D. Oard, and P. Resnik, "A joint model for document segmentation and segment labeling," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 313–322.

[121] H. Kozima and T. Furugori, "Similarity between words computed by spreading activation on an english dictionary," in *Proc. 6th Conf. Eur. Chapter Assoc. Comput. Linguistics*, 1993, pp. 232–239.

[122] G. Glavaš, F. Nanni, and S. P. Ponzetto, "Unsupervised text segmentation using semantic relatedness graphs," Assoc. Comput. Linguistics, Stroudsburg, PA, USA, Tech. Rep., 2016, pp. 125–130.

[123] D. Beeferman, A. Berger, and J. Lafferty, "Text segmentation using exponential models," in *Proc. 2nd Conf. EMNLP*, 1997, pp. 35–46.

[124] R. Papka *et al.*, "On-line new event detection using single pass clustering," Univ. Massachusetts, Amherst, MA, USA, Tech. Rep. 10(290941.290954), 1998.

[125] L. Hu, B. Zhang, L. Hou, and J. Li, "Adaptive online event detection in news streams," *Knowl.-Based Syst.*, vol. 138, pp. 105–112, Dec. 2017.

[126] G. Chen, N. Xu, and W. Mao, "An encoder-memory-decoder framework for sub-event detection in social media," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2018, pp. 1575–1578.

[127] J. G. Fiscus and G. R. Doddington, "Topic detection and tracking evaluation overview," in *Topic Detection Tracking*. Cham, Switzerland: Springer, 2002, pp. 17–31.

[128] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, "The DET curve in assessment of detection task performance," in *Proc. Eur. Conf. Speech Commun. Technol.*, Rhodes, Greece, vol. 4, 1997, pp. 1895–1898.

[129] D. Jurafsky and J. H. Martin, *Speech and Language Processing*. Noida, India: Pearson Education India, 2000.

[130] R. Dale, H. Moisl, and H. Somers, *Handbook of Natural Language Processing*. Boca Raton, FL, USA: CRC Press, 2000.

[131] D. R. Cutting, D. R. Karger, J. O. Pedersen, and J. W. Tukey, "Scatter/gather: A cluster-based approach to browsing large document collections," in *Proc. 15th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 1992, pp. 318–329.
[132] G. Ifrim, B. Shi, and I. Brigadir, "Event detection in Twitter using aggressive filtering and hierarchical tweet clustering," in *Proc. 2nd Workshop Social News Web (SNOW)*, 2014, pp. 1–34.
[133] F. Atefeh and W. Khreich, "A survey of techniques for event detection in Twitter," *Comput. Intell.*, vol. 31, no. 1, pp. 132–164, Feb. 2015.
[134] H. Peng, J. Li, Q. Gong, Y. Song, Y. Ning, K. Lai, and P. S. Yu, "Fine-grained event categorization with heterogeneous graph convolutional networks," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, p. 3238.
[135] M.-Q. Yu, W.-H. Luo, Z.-T. Zhou, and S. Bai, "ICT's approaches to HTD and tracking at TDT2004," in *Proc. Topic Detection Tracking Workshop Rep.*, 2004.
[136] W. Kraaij, "Variations on language modeling for information retrieval," *SIGIR Forum*, vol. 39, no. 1, p. 61, 2005.
[137] J. Allan, A. Feng, and A. Bolivar, "Flexible intrinsic evaluation of hierarchical clustering for TDT," in *Proc. 12th Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2003, pp. 263–270.
[138] T.-J. Fu, P.-H. Li, and W.-Y. Ma, "GraphRel: Modeling text as relational graphs for joint entity and relation extraction," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 1409–1418.
[139] M. Miwa and M. Bansal, "End-to-End relation extraction using LSTMs on sequences and tree structures," 2016, *arXiv:1601.00770*. [Online]. Available: http://arxiv.org/abs/1601.00770
[140] J.-D. Kim, T. Ohta, and J. Tsujii, "Corpus annotation for mining biomedical events from literature," *BMC Bioinf.*, vol. 9, no. 1, p. 10, Dec. 2008.
[141] R. Saurí, J. Littman, B. Knippen, R. Gaizauskas, A. Setzer, and J. Pustejovsky, "Timeml annotation guidelines," *Version*, vol. 1, no. 1, p. 31, 2006.
[142] M. Damashek, "Gauging similarity with n-grams: Language-independent categorization of text," *Science*, vol. 267, no. 5199, pp. 843–848, Feb. 1995.
[143] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1188–1196.
[144] H. M. Wallach, "Topic modeling: Beyond bag-of-words," in *Proc. 23rd Int. Conf. Mach. Learn. (ICML)*, 2006, pp. 977–984.
[145] D. Yogatama and N. A. Smith, "Linguistic structured sparsity in text categorization," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics, Long Papers*, vol. 1, 2014, pp. 786–796.
[146] D. Zelenko, C. Aone, and A. Richardella, "Kernel methods for relation extraction," *J. Mach. Learn. Res.*, vol. 3, pp. 1083–1106, Feb. 2003.
[147] M. Sahami and T. D. Heilman, "A Web-based kernel function for measuring the similarity of short text snippets," in *Proc. 15th Int. Conf. World Wide Web (WWW)*, 2006, pp. 377–386.
[148] S. Zhao and R. Grishman, "Extracting relations with integrated information using kernel methods," in *Proc. 43rd Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2005, pp. 419–426.
[149] L. Ratinov and D. Roth, "Design challenges and misconceptions in named entity recognition," in *Proc. 13th Conf. Comput. Natural Lang. Learn. (CoNLL)*, 2009, pp. 147–155.
[150] N. Kambhatla, "Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations," in *Proc. ACL Interact. Poster Demonstration Sessions*, 2004, p. 22.
[151] Y. Chen, Q. Zheng, and W. Zhang, "Omni-word feature and soft constraint for chinese relation extraction," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics Long Papers*, vol. 1, 2014, pp. 572–581.
[152] H. Lin, Y. Lu, X. Han, and L. Sun, "Nugget proposal networks for chinese event detection," 2018, *arXiv:1805.00249*. [Online]. Available: http://arxiv.org/abs/1805.00249
[153] Y. Chen, Q. Zheng, and P. Chen, "Feature assembly method for extracting relations in chinese," *Artif. Intell.*, vol. 228, pp. 179–194, Nov. 2015.
[154] Y. Chen, Q. Zheng, and P. Chen, "A set space model for feature calculus," *IEEE Intell. Syst.*, vol. 32, no. 5, pp. 36–42, Sep. 2017.
[155] Y. Chen, G. Wang, Q. Zheng, Y. Qin, R. Huang, and P. Chen, "A set space model to capture structural information of a sentence," *IEEE Access*, vol. 7, pp. 142515–142530, 2019.
[156] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer-Verlag, 2006.
[157] C. Nogueira dos Santos, B. Xiang, and B. Zhou, "Classifying relations by ranking with convolutional neural networks," 2015, *arXiv:1504.06580*. [Online]. Available: http://arxiv.org/abs/1504.06580
[158] Y. Xu, L. Mou, G. Li, Y. Chen, H. Peng, and Z. Jin, "Classifying relations via long short term memory networks along shortest dependency paths," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1785–1794.
[159] Z. Guo, Y. Zhang, and W. Lu, "Attention guided graph convolutional networks for relation extraction," 2019, *arXiv:1906.07510*. [Online]. Available: http://arxiv.org/abs/1906.07510
[160] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," 2014, *arXiv:1404.2188*. [Online]. Available: http://arxiv.org/abs/1404.2188
[161] Y. Chen, K. Wang, W. Yang, Y. Qing, R. Huang, and P. Chen, "A multi-channel deep neural network for relation extraction," *IEEE Access*, vol. 8, pp. 13195–13203, 2020.
[162] L. Baldini Soares, N. FitzGerald, J. Ling, and T. Kwiatkowski, "Matching the blanks: Distributional similarity for relation learning," 2019, *arXiv:1906.03158*. [Online]. Available: http://arxiv.org/abs/1906.03158
[163] S. Brin, "Extracting patterns and relations from the world wide Web," in *Proc. World Wide Web Databases*, 1999, pp. 172–183.
[164] A. Moro, H. Li, S. Krause, F. Xu, R. Navigli, and H. Uszkoreit, "Semantic rule filtering for Web-scale relation extraction," in *Proc. ISWC*, 2013, pp. 347–362.
[165] T. P. Mohamed, E. R. Hruschka, Jr., and T. M. Mitchell, "Discovering relations between noun categories," in *Proc. EMNLP*, 2011, pp. 1447–1455.
[166] Y. Lu, H. Lin, X. Han, and L. Sun, "Distilling discrimination and generalization knowledge for event detection via delta-representation learning," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 4366–4376.
[167] Y. Wu, J. Zhao, B. Xu, and H. Yu, "Chinese named entity recognition based on multiple features," in *Proc. Conf. Hum. Lang. Technol. Empirical Methods Natural Lang. Process. (HLT)*, 2005, pp. 427–434.
[168] N. Xue, "Chinese word segmentation as character tagging," in *Proc. Comput. Linguistics Chin. Lang. Process.*, 2003, pp. 29–48.
[169] R. Florian, H. Hassan, A. Ittycheriah, H. Jing, N. Kambhatla, X. Luo, N. Nicolov, S. Roukos, and T. Zhang, "A statistical model for multilingual entity detection and tracking," in *Proc. HLT-NAACL*, 2004, pp. 1–8.
[170] K. Hacioglu, B. Douglas, and Y. Chen, "Detection of entity mentions occurring in english and Chinese text," in *Proc. Conf. Hum. Lang. Technol. Empirical Methods Natural Lang. Process. (HLT)*, 2005, pp. 379–386.
[171] B. Alex, B. Haddow, and C. Grover, "Recognising nested named entities in biomedical text," in *Proc. Workshop BioNLP Biol., Translational, Clin. Lang. Process. (BioNLP)*, 2007, pp. 65–72.
[172] J. R. Finkel and C. D. Manning, "Nested named entity recognition," in *Proc. EMNLP*, 2009, pp. 141–150.
[173] J. R. Finkel and C. D. Manning, "Joint parsing and named entity recognition," in *Proc. Hum. Lang. Technol., Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics (NAACL)*, 2009, pp. 326–334.
[174] X. Zhang, D. Li, and X. Wu, "Parsing named entity as syntactic structure," in *Proc. ISCA*, 2014, pp. 278–282.
[175] Z. Jie, A. O. Muis, and W. Lu, "Efficient dependency-guided named entity recognition.," in *Proc. AAAI*, 2017, pp. 3457–3465.
[176] Y. Chen, Q. Zheng, and P. Chen, "A boundary assembling method for Chinese entity-mention recognition," *IEEE Intell. Syst.*, vol. 30, no. 6, pp. 50–58, Nov. 2015.
[177] Y. Chen, Y. Wu, Y. Qin, Y. Hu, Z. Wang, R. Huang, X. Cheng, and P. Chen, "Recognizing nested named entity based on the neural network boundary assembling model," *IEEE Intell. Syst.*, vol. 35, no. 1, pp. 74–81, Jan. 2020.
[178] J. Chen, N. Xue, and M. Palmer, "Using a smoothing maximum entropy model for Chinese nominal entity tagging," in *Proc. IJCNLP*. Cham, Switzerland: Springer, 2005, pp. 493–499.
[179] Y. Wu, J. Zhao, and B. Xu, "Chinese named entity recognition combining a statistical model with human knowledge," in *Proc. ACL Workshop Multilingual Mixed-Lang. Named Entity Recognit.*, 2003, pp. 65–72.
[180] F. Peng, F. Feng, and A. McCallum, "Chinese segmentation and new word detection using conditional random fields," in *Proc. 20th Int. Conf. Comput. Linguistics (COLING)*, 2004, p. 562.
[181] J. Gao, M. Li, C.-N. Huang, and A. Wu, "Chinese word segmentation and named entity recognition: A pragmatic approach," *Comput. Linguistics*, vol. 31, no. 4, pp. 531–574, Dec. 2005.
[182] M. Wang, W. Che, and C. D. Manning, "Joint word alignment and bilingual named entity recognition using dual decomposition," in *Proc. ACL*, 2013, pp. 1073–1082.

[183] J. Sun, J. Gao, L. Zhang, M. Zhou, and C. Huang, "Chinese named entity identification using class-based language model," in *Proc. 19th Int. Conf. Comput. Linguistics*, 2002, pp. 1–7.
[184] L.-J. Wang, W.-C. Li, and C.-H. Chang, "Recognizing unregistered names for mandarin word identification," in *Proc. 14th Conf. Comput. Linguistics*, 1992, pp. 1239–1243.
[185] J. Aberdeen, J. Burger, D. Connolly, S. Roberts, and M. Vilain, "MITRE-bedford: Description of the ALEMBIC system as used for MUC-4," in *Proc. 4th Conf. Message Understand. (MUC)*, 1992, pp. 141–155.
[186] G. R. Krupka and K. Hausman, "IsoQuest Inc.: Description of the NetOwl extractor system as used for MUC-7," in *Proc. MUC*, vol. 7, 1998, pp. 1–10.
[187] W. J. Black, F. Rinaldi, and D. Mowatt, "Facile: Description of the ne system used for MUC-7," in *Proc. 7th MUC*, 1998, pp. 1–10.
[188] K.-J. Chen and M.-H. Bai, "Unknown word detection for Chinese by a corpus-based learning method," *Int. J. Comput. Linguistics Chin.*, vol. 3, no. 1, pp. 27–44, Feb. 1998.
[189] M. Banko and O. Etzioni, "The tradeoffs between open and traditional relation extraction," in *Proc. ACL-HLT*, 2008, pp. 28–36.
[190] M. Zhang, J. Zhang, J. Su, and G. Zhou, "A composite kernel to extract relations between entities with both flat and structured features," in *Proc. 21st Int. Conf. Comput. Linguistics 44th Annu. Meeting (ACL)*, 2006, pp. 825–832.
[191] D. Roth and W.-T. Yih, "Probabilistic reasoning for entity & relation recognition," in *Proc. 19th Int. Conf. Comput. Linguistics*, 2002, pp. 1–7.
[192] D. Roth and W.-T. Yih, "Global inference for entity and relation identification via a linear programming formulation," in *Introduction to Statistical Relational Learning*, 2007, pp. 553–580.
[193] Z. GuoDong, S. Jian, Z. Jie, and Z. Min, "Exploring various knowledge in relation extraction," in *Proc. 43rd Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2005, pp. 427–434.
[194] D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao, "Relation classification via convolutional deep neural network," in *Proc. 25th Int. Conf. Comput. Linguistics, Tech. Papers (COLING)*, 2014, pp. 2335–2344.
[195] M. Collins and N. Duffy, "Convolution kernels for natural language," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, pp. 625–632.
[196] L. Sun and X. Han, "A feature-enriched tree kernel for relation extraction," in *Proc. 52nd ACL*, vol. 2, 2014, pp. 61–67.
[197] S. Vashishth, R. Joshi, S. Suman Prayaga, C. Bhattacharyya, and P. Talukdar, "RESIDE: Improving distantly-supervised neural relation extraction using side information," 2018, *arXiv:1812.04361*. [Online]. Available: http://arxiv.org/abs/1812.04361
[198] P. Qin, W. Xu, and W. Yang Wang, "Robust distant supervision relation extraction via deep reinforcement learning," 2018, *arXiv:1805.09927*. [Online]. Available: http://arxiv.org/abs/1805.09927
[199] Y. Yuan, L. Liu, S. Tang, Z. Zhang, Y. Zhuang, S. Pu, F. Wu, and X. Ren, "Cross-relation cross-bag attention for distantly-supervised relation extraction," in *Proc. AAAI Artif Intell*, vol. 33, 2019, pp. 419–426.
[200] D. Zeng, K. Liu, Y. Chen, and J. Zhao, "Distant supervision for relation extraction via piecewise convolutional neural networks," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1753–1762.
[201] P. Zhou, J. Xu, Z. Qi, H. Bao, Z. Chen, and B. Xu, "Distant supervision for relation extraction with hierarchical selective attention," *Neural Netw.*, vol. 108, pp. 240–247, Dec. 2018.
[202] R. McDonald, F. Pereira, S. Kulick, S. Winters, Y. Jin, and P. White, "Simple algorithms for complex relation extraction with applications to biomedical IE," in *Proc. 43rd Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2005, pp. 491–498.
[203] A. Mandya, D. Bollegala, F. Coenen, and K. Atkinson, "Combining long short term memory and convolutional neural network for cross-sentence n-ary relation extraction," 2018, *arXiv:1811.00845*. [Online]. Available: http://arxiv.org/abs/1811.00845
[204] N. Peng, H. Poon, C. Quirk, K. Toutanova, and W.-T. Yih, "Cross-sentence N-ary relation extraction with graph LSTMs," in *Proc. TACL*, vol. 5, 2017, pp. 101–115.
[205] K. Akimoto, T. Hiraoka, K. Sadamasa, and M. Niepert, "Cross-sentence N-ary relation extraction using lower-arity universal schemas," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 6226–6232.
[206] R. Jia, C. Wong, and H. Poon, "Document-level N-ary relation extraction with multiscale representation learning," 2019, *arXiv:1904.02347*. [Online]. Available: http://arxiv.org/abs/1904.02347
[207] R. Sukthanker, S. Poria, E. Cambria, and R. Thirunavukarasu, "Anaphora and coreference resolution: A review," *Inf. Fusion*, vol. 59, pp. 139–162, Jul. 2020.
[208] L. Wang, Z. Tu, A. Way, and Q. Liu, "Exploiting cross-sentence context for neural machine translation," 2017, *arXiv:1704.04347*. [Online]. Available: http://arxiv.org/abs/1704.04347
[209] L. Yao, S. Riedel, and A. McCallum, "Collective cross-document relation extraction without labelled data," in *Proc. EMNLP*, 2010, pp. 1013–1023.
[210] P. Gupta, S. Rajaram, H. Schütze, and T. Runkler, "Neural relation extraction within and across sentence boundaries," in *Proc. AAAI*, vol. 33, 2019, pp. 6513–6520.
[211] M. Miwa and Y. Sasaki, "Modeling joint entity and relation extraction with table representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1858–1869.
[212] M. Zhang, Y. Zhang, and G. Fu, "End-to-End neural relation extraction with global optimization," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 1730–1740.
[213] H. Liu, P. Wang, F. Wu, P. Jiao, W. Wang, X. Xie, and Y. Sun, "Reet: Joint relation extraction and entity typing via multi-task learning," in *Proc. NLPCC*. Cham, Switzerland: Springer, 2019, pp. 327–339.
[214] V. Ng, "Supervised noun phrase coreference research: The first fifteen years," in *Proc. ACL*, 2010, pp. 1396–1411.
[215] A. McCallum and B. Wellner, "Toward conditional models of identity uncertainty with application to proper noun coreference," in *Proc. IJCAI Workshop Inf. Integr. Web*, Acapulco, Mexico, 2003, pp. 76–86.
[216] J. F. McCarthy and W. G. Lehnert, "Using decision trees for coreference resolution," in *Proc. IJCAI*, 1995, pp. 1–7.
[217] W. M. Soon, H. T. Ng, and D. C. Y. Lim, "A machine learning approach to coreference resolution of noun phrases," *Comput. Linguistics*, vol. 27, no. 4, pp. 521–544, Dec. 2001.
[218] F. Laws, F. Heimerl, and H. Schütze, "Active learning for coreference resolution," in *Proc. ACL-HLT*, 2012, pp. 508–512.
[219] V. Ng, "Unsupervised models for coreference resolution," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2008, pp. 640–649.
[220] C. A. Bejan and S. Harabagiu, "Unsupervised event coreference resolution with rich linguistic features," in *Proc. ACL*, 2010, pp. 1412–1422.
[221] X. Yang, J. Su, J. Lang, C. L. Tan, T. Liu, and S. Li, "An entity-mention model for coreference resolution with inductive logic programming.," in *Proc. ACL*, 2008, pp. 843–851.
[222] A. Culotta, M. Wick, R. Hall, and A. McCallum, "First-order probabilistic models for coreference resolution," in *Proc. Main Conf. Hum. Lang. Technol., Conf. North Amer. Chapter ACL*, 2007, pp. 81–88.
[223] H. Poon and P. Domingos, "Joint unsupervised coreference resolution with Markov logic," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2008, pp. 650–659.
[224] A. Haghighi and D. Klein, "Coreference resolution in a modular, entity-centered model," in *Proc. ACL*, 2010, pp. 385–393.
[225] A. Haghighi and D. Klein, "Unsupervised coreference resolution in a nonparametric Bayesian model," in *Proc. ACL*, vol. 45, 2007, p. 848.
[226] K. Lee, L. He, M. Lewis, and L. Zettlemoyer, "End-to-end neural coreference resolution," 2017, *arXiv:1707.07045*. [Online]. Available: http://arxiv.org/abs/1707.07045
[227] C. Nicolae and G. Nicolae, "BestCut: A graph algorithm for coreference resolution," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2006, pp. 275–283.
[228] A. McCallum and B. Wellner, "Conditional models of identity uncertainty with application to noun coreference," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 905–912.
[229] R. Yangarber, "Scenario customization for information extraction," Defense Adv. Res. Projects Agency, Arlington, VA, USA, Tech. Rep., 2001.
[230] H. L. Chieu and H. T. Ng, "A maximum entropy approach to information extraction from semi-structured and free text," in *Proc. AAAI/IAAI*, 2002, pp. 786–791.
[231] D. Ahn, "The stages of event extraction," in *Proc. Workshop Annotating Reasoning About Time Events (ARTE)*, 2006, pp. 1–8.
[232] W. Daelemans, J. Zavrel, K. van der Sloot, and A. Van den Bosch, "Timbl: Tilburg memory-based learner," Tilburg Univ., Tilburg, The Netherlands, Tech. Rep. ILK 04-02, 2004.
[233] H. Daumé, III. (2004). *Notes on CG and LM-BFGS Optimization of Logistic Regression*. vol. 198, p. 282. [Online]. Available: http://pub.hal3.name#daume04cg-bfgs and http://hal3.name/megam

[234] E. Riloff and J. Shoen, "Automatically acquiring conceptual patterns without an annotated corpus," in *Proc. 3rd Workshop Very Large Corpora*, vol. 3, 1995, pp. 1–14.
[235] Z. Chen and H. Ji, "Language specific issue and feature exploration in Chinese event extraction," in *Proc. Hum. Lang. Technol., Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Companion Short Papers (NAACL)*, 2009, pp. 209–212.
[236] H. Llorens, E. Saquete, and B. Navarro-Colorado, "Timeml events recognition and classification: Learning crf models with semantic roles," in *Proc. 23rd Int. Conf. Comput. Linguistics*, 2010, pp. 725–733.
[237] Y. Zhang, Z. Liu, W. Zhou, and N. Natural Science Foundation of China (NSFC), "Event recognition based on deep learning in chinese texts," *PLoS ONE*, vol. 11, no. 8, Aug. 2016, Art. no. e0160147.
[238] N. Ramakrishnan *et al.*, "'Beating the news' with EMBERS: Forecasting civil unrest using open source indicators," 2014, *arXiv:1402.7035*. [Online]. Available: http://arxiv.org/abs/1402.7035
[239] C. Faloutsos, K. S. McCurley, and A. Tomkins, "Fast discovery of connection subgraphs," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2004, pp. 118–127.
[240] Q. Zhao and P. Mitra, "Event detection and visualization for social text streams," in *Proc. ICWSM*, 2007, pp. 1–4.
[241] J. Weng and B.-S. Lee, "Event detection in Twitter," *Proc. ICWSM*, vol. 11, 2011, pp. 401–408.
[242] H. Sayyadi, M. Hurst, and A. Maykov, "Event detection and tracking in social streams," in *Proc. ICWSM*, 2009, pp. 1–4.
[243] P. Meladianos, G. Nikolentzos, F. Rousseau, Y. Stavrakas, and M. Vazirgiannis, "Degeneracy-based real-time sub-event detection in Twitter stream," in *Proc. ICWSM*, 2015, pp. 248–257.
[244] A. Kontostathis, L. M. Galitsky, W. M. Pottenger, S. Roy, and D. J. Phelps, "A survey of emerging trend detection in textual data mining," in *Survey of Text Mining*. Cham, Switzerland: Springer, 2004, pp. 185–224.
[245] Y. Ohsawa, N. E. Benson, and M. Yachida, "Keygraph: Automatic indexing by co-occurrence graph based on building construction metaphor," in *Proc. ADL IEEE Int. Forum*, 1998, pp. 12–18.
[246] J. M. Kleinberg, "Bursty and Hierarchical Structure in Streams," *Data Mining Knowl. Discovery*, vol. 7, no. 4, pp. 373–397, 2003.
[247] M. Imran, C. Castillo, F. Diaz, and S. Vieweg, "Processing social media messages in mass emergency: Survey summary," in *Proc. Companion The Web Conf. Web Conf. (WWW)*, 2018, pp. 507–511.
[248] E. Kuzey, J. Vreeken, and G. Weikum, "A fresh look on knowledge bases: Distilling named events from news," in *Proc. 23rd ACM Int. Conf. Conf. Inf. Knowl. Manage. (CIKM)*, 2014, pp. 1689–1698.
[249] B. Liu, D. Niu, K. Lai, L. Kong, and Y. Xu, "Growing story forest online from massive breaking news," in *Proc. ACM Conf. Inf. Knowl. Manage.*, Nov. 2017, pp. 777–785.
[250] G. Glavaš and J. Šnajder, "Event graphs for information retrieval and multi-document summarization," *Expert Syst. Appl.*, vol. 41, no. 15, pp. 6904–6916, Nov. 2014.
[251] E. Kuzey and G. Weikum, "EVIN: Building a knowledge base of events," in *Proc. 23rd Int. Conf. World Wide Web (WWW) Companion*, 2014, pp. 103–106.
[252] P. Vossen, G. Rigau, L. Serafini, P. Stouten, F. Irving, and W. R. van Hage, "Newsreader: Recording history from daily news streams.," in *Proc. LREC*, 2014, pp. 2000–2007.
[253] P. Vossen, R. Agerri, I. Aldabe, A. Cybulska, M. van Erp, A. Fokkens, E. Laparra, A.-L. Minard, A. Palmero Aprosio, G. Rigau, M. Rospocher, and R. Segers, "NewsReader: Using knowledge resources in a cross-lingual reading machine to generate more knowledge from massive streams of news," *Knowl.-Based Syst.*, vol. 110, pp. 60–85, Oct. 2016.
[254] R. Blanco and C. Lioma, "Graph-based term weighting for information retrieval," *Inf. Retr.*, vol. 15, no. 1, pp. 54–92, Feb. 2012.
[255] A. Subburathinam, D. Lu, H. Ji, J. May, S.-F. Chang, A. Sil, and C. Voss, "Cross-lingual structure transfer for relation and event extraction," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 313–325.
[256] W. Wang, R. Besançon, O. Ferret, and B. Grau, "Filtering and clustering relations for unsupervised information extraction in open domain," in *Proc. 20th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2011, pp. 1405–1414.
[257] D. Downey, S. Schoenmackers, and O. Etzioni, "Sparse information extraction: Unsupervised language models to the rescue," Dept. Comput. Sci. Eng., Univ. Washington, Seattle, WA, USA, Tech. Rep., 2007.
[258] Y. Xu, M.-Y. Kim, K. Quinn, R. Goebel, and D. Barbosa, "Open information extraction with tree kernels," in *Proc. HLT-NAACL*, 2013, pp. 1–10.
[259] S. Takamatsu, I. Sato, and H. Nakagawa, "Reducing wrong labels in distant supervision for relation extraction," in *Proc. ACL*, 2012, pp. 721–729.
[260] S. Riedel, L. Yao, A. McCallum, and B. M. Marlin, "Relation extraction with matrix factorization and universal schemas," in *Proc. Conf. North Amer. Chapter ACL, Hum. Lang. Technol.*, 2013, pp. 74–84.
[261] J. R. Curran, T. Murphy, and B. Scholz, "Minimising semantic drift with mutual exclusion bootstrapping," in *Proc. PACLING*, 2007, pp. 172–180.
[262] J. Scott, *Social Network Analysis*. Newbury Park, CA, USA: Sage, 2017.
[263] R. Ahirrao and S. Patel, "An overview on event evolution technique," *Int. J. Comput. Appl.*, vol. 77, no. 10, pp. 7–11, 2013.
[264] G. Kaiser, *A Friendly Guide to Wavelets*. Cham, Switzerland: Springer, 2010.
[265] X. Dong, D. Mavroeidis, F. Calabrese, and P. Frossard, "Multiscale event detection in social media," *Data Mining Knowl. Discovery*, vol. 29, no. 5, pp. 1374–1405, Sep. 2015.
[266] R. Parikh and K. Karlapalem, "ET: Events from tweets," in *Proc. 22nd Int. Conf. World Wide Web (WWW) Companion*, 2013, pp. 613–620.
[267] A. Abujabal and K. Berberich, "Important events in the past, present, and future," in *Proc. 24th Int. Conf. World Wide Web (WWW) Companion*, 2015, pp. 1315–1320.
[268] H. Kozima, "Text segmentation based on similarity between words," in *Proc. 31st Annu. Meeting Assoc. Comput. Linguistics*, 1993, pp. 286–288.
[269] M. Fan, D. Zhao, Q. Zhou, Z. Liu, T. F. Zheng, and E. Y. Chang, "Distant supervision for relation extraction with matrix completion," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics Long Papers*, 2014, pp. 839–849.
[270] E. Grave, "A convex relaxation for weakly supervised relation extraction," in *Proc. Conf. EMNLP, ACL*, 2014, pp. 1580–1590.
[271] A. Fader, S. Soderland, and O. Etzioni, "Identifying relations for open information extraction," in *Proc. EMNLP*, 2011, pp. 1535–1545.
[272] A. Ritter, Mausam, O. Etzioni, and S. Clark, "Open domain event extraction from Twitter," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2012, pp. 1104–1112.
[273] N. Chambers and D. Jurafsky, "Unsupervised learning of narrative event chains," in *Proc. ACL HLT*, 2008, pp. 789–797.
[274] R. C. Schank and R. P. Abelson, *Scripts, Plans, Goals and Understanding: An Inquiry Into Human Knowledge Structures*. Hove, U.K.: Psychology Press, 2013.
[275] N. Du, H. Dai, R. Trivedi, U. Upadhyay, M. Gomez-Rodriguez, and L. Song, "Recurrent marked temporal point processes: Embedding event history to vector," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 1555–1564.
[276] R. Trivedi, H. Dai, Y. Wang, and L. Song, "Know-evolve: Deep temporal reasoning for dynamic knowledge graphs," 2017, *arXiv:1705.05742*. [Online]. Available: http://arxiv.org/abs/1705.05742
[277] S. M. Kazemi, R. Goel, K. Jain, I. Kobyzev, A. Sethi, P. Forsyth, and P. Poupart, "Representation learning for dynamic graphs: A survey," *J. Mach. Learn. Res.*, vol. 21, no. 70, pp. 1–73, 2020.

YANPING CHEN is currently an Associate Professor with the College of Computer Science and Technology, Guizhou University, Guiyang. His research interests include artificial intelligence and natural language processing.

ZEHUA DING is currently pursuing the degree with the College of Computer Science and Technology, Guizhou University, Guiyang. His research interest includes natural language processing.

QINGHUA ZHENG (Member, IEEE) is currently a Professor with the Department of Computer Science and Technology, Xi'an Jiaotong University. His research interests include multimedia distance education and computer network security.

RUIZHANG HUANG is currently an Associate Professor with the College of Computer Science and Technology, Guizhou University, Guiyang. Her research interests include information retrieval and text mining.

YONGBIN QIN is currently a Professor with the College of Computer Science and Technology, Guizhou University, Guiyang. His research interests include big data processing, cloud computing, and text mining.

NAZARAF SHAH is a Senior Lecturer at Coventry University, Coventry, U.K. He received the Ph.D. degree in multi-agent systems from Coventry University, in 2006. He has more than 90 publications in various international conference proceedings and journals. His research interests include intelligent agents, service oriented computing, and cloud computing.