World Wide Web (2020) 23:2043–2077
https://doi.org/10.1007/s11280-020-00790-2

# Multi-document semantic relation extraction for news analytics

**Yongpan Sheng[1] · Zenglin Xu[1] · Yafang Wang[2] · Gerard de Melo[3]**

Received: 8 November 2019 / Revised: 7 January 2020 / Accepted: 15 January 2020 /
Published online: 18 May 2020


**Abstract**
Given the overwhelming amounts of information in our current 24/7 stream of new incoming articles, new techniques are needed to enable users to focus on just the key entities and concepts along with their relationships. Examples include news articles but also business reports and social media. The fact that relevant information may be distributed across diverse sources makes it particularly challenging to identify relevant connections. In this paper, we propose a system called *MuReX* to aid users in quickly discerning salient connections and facts from a set of related documents and viewing the resulting information as a graph-based visualization. Our approach involves open information extraction, followed by a careful transformation and filtering approach. We rely on integer linear programming to ensure that we retain only the most confident and compatible facts with regard to a user query, and finally apply a graph ranking approach to obtain a coherent graph that represents meaningful and salient relationships, which users may explore visually. Experimental results corroborate the effectiveness of our proposed approaches, and the local system we developed has been running for more than one year.

**Keywords** Multi-document semantic extraction system · Open information extraction · Graph-based visualization

## 1 Introduction

In today's digital and highly interconnected world, there is an endless 24/7 stream of new articles appearing online, including news reports, business transactions, digital media, etc.

This article belongs to the Topical Collection: *Special Issue on Web and Big Data 2019*
Guest Editors: Jie Shao, Man Lung Yiu, and Toyoda Masashi

✉ Gerard de Melo
gdm@demelo.org

[1] SMILE Lab, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan, China

[2] Ant Financial Services Co., Hangzhou, Zhejiang, China

[3] Department of Computer Science, Rutgers University–New Brunswick, Piscataway, NJ, USA

[51]. Faced with these overwhelming amounts of information, people may desire to have a better understanding of what the key topics and facts in the news are, what user comments focus on, and how new articles are related to previous news. In some cases, they may desire more detailed information about key entities and concepts and their relationships. However, such topic-specific key facts in the news are often spread across a number of disparate articles and sources. Not only do different outlets often cover different aspects of a story, but typically, also new information only becomes available over time, so new articles in a developing story need to be connected to previous ones, or to historic documents providing relevant background information. Therefore, it is crucial to analyze and explore different aspects of news data to satisfy people's information needs.

Topic Detection and Tracking (TDT) has been attracting substantial attention in academia and industry, and addresses many problems that relate to the structure of the news stream, news modeling, topic detection, and so on. Google, for instance, organizes news in terms of collections of similar articles. However, TDT operates at the document level. It does not involve identifying individual facts and tracking their connections across articles.

Given a query topic, a user is often expected to grasp the key information by scouring through a long list of relevant documents. In contrast, information providers sometimes rely on conceptual graphs to gather and deliver news information [8], and this is also a natural way for information consumers to grasp significant facts in a specific domain. Relying on conceptual graphs to organize and manage news can help us integrate news from different facets, so that we can turn a collection of news into multifaceted information (person, location, time, etc.). Such conceptual graphs differ from traditional graphs that have only a single relation. They also differ from knowledge graphs [18], which normally have a predefined set of relations. We consider conceptual graphs consisting of an open-ended inventory of noun phrase concepts as nodes, and an open-ended inventory of potential relationships to connect the concepts, enabling them to very flexibly encode heterogeneous information. This applies especially for user-specified news topics, as illustrated by an example based on the topic "*presidential election of the US*" in Figure 1.

Achieving the transformation from a raw collection of news articles to this sort of graph is a challenging problem. The main difficulties are three-fold: (1) Newswire text often consists of challenging natural language sentences that are non-trivial to interpret. Traditional relation extraction systems require us to pre-specify the set of relations of interest. This is obviously not suitable for news documents in light of the open-ended nature of the diverse relationships that they may cover. (2) Important concepts and facts appear both within and

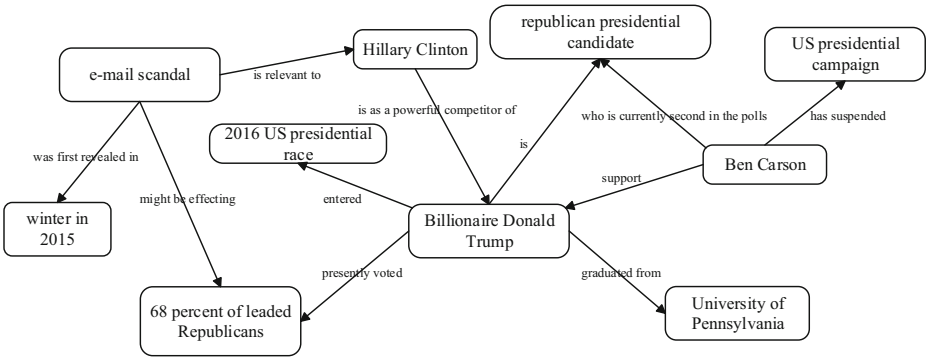



**Figure 1** Example of a conceptual graph on the topic "*presidential election of the US*"

across different unstructured text sources, and how to recognize and normalize them, so as to cope with the high level of redundancy across documents, is not obvious. (3) Even after identifying and normalizing factual information, identifying the most salient pieces of information and constructing a coherent graph that can easily be perused by users is a challenging task that has not been solved in previous work.

In view of the above challenges, we propose a novel multi-document semantic relation extraction system called *MuReX*, to extract salient entities, concepts, and their relationships, discover connections within and across them, such that the resulting information can be represented in a graph-based visualization. We rely on a series of natural language processing techniques, including an open information extraction (Open IE) method with several transformations to automatically extract large amounts of candidate facts (i.e., subject-predicate-object triples) from the pre-processed text documents. However, Open IE does not make any attempt to connect the extracted triples across sentences or even documents. Additionally, many of the triples may be correct yet irrelevant to the user-specified query topic. To overcome these issues, we propose a two-stage candidate triple filtering (TCTF) approach to further filter out large numbers of irrelevant and meaningless triples with regard to the given topic of interest. Moreover, we propose an integer linear programming (ILP) approach for local compatibility to avoid incoherent facts within the filtered results and to integrate them in the form of an initial graph. We further improve this graph based on the heuristic strategy of iteratively removing the weakest concepts with relatively lower importance scores, so as to ensure that the final large conceptual graph only consists of facts that are likely to represent meaningful and salient relationships, which users may explore visually.

The key contributions of our work are as follows:

- We adapt open information extraction (Open IE) such that more correct and meaningful candidate facts are extracted from the input unstructured text documents.
- We propose a two-stage candidate triple filtering approach based on an improved self-learning framework to discern which of the extracted candidate facts are coherent with regard to the pre-specified query topic.
- We model local compatibility of the facts as a constrained optimization problem, and propose an integer linear programming approach to solve it and avoid incoherent facts within the filtered results. Based on the results, we further introduce a heuristic strategy to construct the final conceptual graph consisting only of facts that are likely to represent meaningful and salient relationships that can be explored visually.
- We conduct extensive experiments on real-world English language news articles collected from Web sources, and compare our results against several state-of-the-art Open IE methods. We also investigate the quality of the final generated conceptual graph for different query topics with regard to its coverage rate of topic entities and concepts, the confidence score, and the compatibility of involved facts. The experimental results demonstrate the effectiveness of the proposed approaches.
- Our overall multi-document semantic extraction system, called *MuReX*, consists of a backend that implements the aforementioned algorithms to ensure that only those facts and connections that are deemed most salient and meaningful within and across multiple related documents are maintained. In the frontend, the system provides compelling visual views, covering all steps of the processing pipeline.

The remainder of this paper is organized as follows. We review related research in this area in Section 2 and then provide some preliminaries related to our proposed system in Section 3. Section 4 provides the details of the core algorithms that drive *MuReX*. In Section 5, we conduct an extensive experimental evaluation and provide an analysis of the effectiveness of the different phases of our *MuReX* system. In Section 6, we present the interactive visualization aspects of our system. Finally, the conclusions and future work are described in Section 7.

## 2 Related work

In this section, we review the literature along four distinct lines of research: news representation, topic detection and tracking, open information extraction, and multi-document semantic mining. We highlight the unique aspects of our proposed approaches compared to each line of work.

### 2.1 News representation

The IPTC (International Press Telecommunications Council) has been specifying standards for exchanging news since 1979, including the *NewsML* and *EventsML* standards for conveying news and event information in the news industry [6, 8]. With the growing prominence of Web news media, *rNews* was introduced in 2011 to define the terminology and data model required to embed news-specific meta-data into HTML documents [7]. Google, Microsoft, and Yahoo!'s schema.org effort provides a standard markup vocabulary that, among other things, covers the different items pertaining to an event [16].

In this paper, we study conceptual graphs as another important representation form of news contents. Conceptual graphs resemble mind maps in that they intuitively reflect the way people conceptualize the world in terms of entities, concepts, and their relationships. We hypothesize that such graphs are helpful in discerning the key entities and concepts and their relationships within as well as across documents. Falke and Gurevych [11] explored the idea that when concepts and relations are linked to corresponding locations in the documents they have been extracted from, the graph can be used to navigate in a collection of documents, similar to a table of contents.

### 2.2 Topic detection and tracking

Topic Detection and Tracking (TDT) is a research topic that aims at discovering the topical structure in large streams of news stemming from multiple news outlets [49, 54].

Leskovec *et al.* [23] developed a framework for tracking short, distinctive phrases that travel online while for the most part remaining intact. Li *et al.* [24] designed a flexible topic-driven framework for news exploration. Wang *et al.* [48] proposed a topic model that connects news articles and social media. Hu *et al.* [20] designed a model for travel recommendation from multi-source social media data. Shahaf and Guestrin [37] investigated methods to construct connections between different pieces of information to form a chain across a specific topic. Xu *et al.* [50] proposed a novel media annotation method based on analytics of streaming social interactions using media content instead of the metadata.

Lin *et al.* [25] as well as Mei and Zhai [29] studied the task of Temporal Text Mining (TTM), which seeks to discover and summarize the evolution of patterns of themes in a text stream. Pouliquen *et al.* [34] developed the Europe Media Monitor (EMM) system for cross-lingual story tracking, gathering, grouping and linking of news over time. Shan *et al.* [38] developed the *EventSearch* system for event extraction and retrieval on four types of news-related historical data.

In this paper, we do not consider the detection and tracking of topics or events within the news stream. Rather, our work aims at an in-depth semantic analysis of multiple documents, in particular to capture meaningful and salient *factual* relationships and connections within and across documents. For this, our system allows the user to provide a topic of interest that is used to select a set of relevant input documents.

### 2.3 Open information extraction

Traditional pattern-based information extraction approaches focus on particular relations. For example, the Hearst pattern approach [17] considers the constructions "$NP_Y$ such as $NP_X$" or "$NP_X$ and other $NP_Y$" for hypernym–hyponym relation extraction. Tandon *et al.* [43] investigated scaling such techniques up to hundreds of thousands of patterns that are automatically identified and aggregated. Their study considered 3 different relations.

Open IE [2] instead seeks to extract relational triples from a given text in an open-ended manner, capturing a wide range of possible relationships between items. ReVerb [10] identifies relational phrases via part-of-speech based regular expressions. Carlson and Mitchell *et al.* [5, 32] proposed a never-ending language learning system called NELL based on free-text predicate patterns. There have been attempts to draw on more linguistically informed signals rather than shallow string matching patterns. This can be achieved for instance by invoking dependency parsing or semantic role labeling, so as to better capture long-distance relationships. Along these lines, the ClauseIE [9] system induces short but coherent relationships along dependency paths, also supporting n-ary propositions. Angeli *et al.* [1] adopt a clause splitter using distant supervision and also investigate statistically mapping predicates to a known relation schema. MinIE [14] removes overly specific constituents and captures implicit relations by introducing several statistic measures such as polarity, modality, attribution, and quantities. Other works focused on open-ended extraction of particular kinds of facts, such as object properties [45], comparisons [46], or activities [44].

Compared with previous work, in this paper, we propose a method that leverages the results of open information extraction approaches of the sort described above, but attempts to go beyond Open IE by applying additional transformations and filtering. One reason for this is that open information extraction tools often produce noisy extractions stemming from failed syntactic analyses or from error propagation in their pipeline. We hence rely on rule-based constraints to mitigate the problem.

Ultimately, the task considered in this paper requires going beyond raw open information extraction. While Open IE systems do extract specific facts from text, they do not aim to connect facts across sentences or documents [13, 35], and often neglect whether the extractions are meaningful on their own and indicative of what is genuinely being expressed in the input text. Additionally, our task requires choosing a coherent set of particularly salient facts [51] to present to the user.

### 2.4 Mining multi-document semantics

GoWvis[1] [47] is an interactive web application that generates single-document summarizations for a text provided as input, by producing a Graph-of-Words representation. Edges in such graphs, however, merely represent co-occurrences of words rather than specific relationships expressed in the text. The Networks of Names project[2] [22] adopts a similar strategy, but restricted to named entities, i.e., any two named entities co-occurring in the same sentence are considered related. The system additionally allows the user to assign labels to relationships and can then attempt to find further instances of such relationships, but this requires the user to provide significant amounts of training data for each kind of relation. The Network of the Day project[3] [3] builds on Networks of Names to provide a daily analysis of German news articles. The *news/s/leak* project[4] [52] further extends this line of work by adding access to additional corpora such as the WikiLeaks PlusD collection and by providing additional forms of document analytics. The aim is to aid journalists in discovering and analysing newsworthy connections within such corpora. This version also attaches general document keywords as tags to relationships, but does not aim at sentence-level relation semantics as considered in our work.

Ge *et al.* [15] presented an integrated system that combines crowdsourcing, semi-automatic extraction from text, and visual analytics across large amounts of spatio-temporal data. This system does not, however, display networks of entities, but focuses on insights that cannot be easily observed in individual facts. The present article extends a prior publication [39], which introduced a system that can extract facts from a set of related articles. However, that paper did not provide an adequate evaluation with regard to the quality of the extracted facts.

Ji *et al.* [21] used information extraction-based features to improve multi-document text summarization. Mann [26] investigated logical constraints to aggregate a closed predefined set of relations mined from multiple documents. Our system, in contrast, addresses the task of extracting and connecting arbitrary salient entities and facts from multiple documents, enabling a deeper exploration of meaningful connections.

## 3 Preliminaries

In this section, we first formulate our problem and then proceed to provide an overall overview of our proposed *MuReX* system.

### 3.1 Problem formulation

The objective in our work is to provide means of assisting users in quickly finding meaningful and salient connections and facts from a collection of relevant documents. This can be broken down into five major subtasks:

– **Subtask 1: Data Preprocessing.** Given a collection of documents $D_t = \{d_i \mid i = 1, \ldots, N\}$, the objective in this subtask is to extract a set of sentences $S_k = \{S^i \mid i = 1, \ldots, K\}$ from $D_k$, the top-$k$ documents within $D_t$, where $S^i$ represents a selected collection of top-$k$ sentences in $d_i$.

[1] https://safetyapp.shinyapps.io/GoWvis/
[2] http://maggie.lt.informatik.tu-darmstadt.de/thesis/master/NetworksOfNames
[3] http://tagesnetzwerk.de
[4] http://www.newsleak.io/

- **Subtask 2: Candidate Fact Extraction.** Given a specified query topic $T$, the goal in this subtask is to extract a set of facts $F_c = \{f_i \mid i = 1, \ldots, M\}$ from $S_k$. Each of these facts is a triple of the form $(s, r, o)$, consisting of a *subject* $s$, *relation* $r$, and *object* $o$. Since we still need to assess the saliency and coherence of these facts with regard to the query $T$, we refer to them as *candidate facts*.
- **Subtask 3: Topic Coherence Estimation of Candidate Facts.** The goal of this subtask is to pick a subset $F_t \subseteq F_c$ such that the facts are coherent with the user's query topic $T$.
- **Subtask 4: Compatibility Measure of Facts.** The goal in this subtask is to identify a further subset $F'_t \subseteq F_t$ that additionally maintains a high degree of local compatibility as well.
- **Subtask 5: Conceptual Graph Construction.** The goal of this final subtask is to determine which of the facts from $F'_t$ generated in the previous subtask are more likely to be salient, which of their entities and concepts to merge, and, when merging, which of the available labels to leverage in the final conceptual graph $G$.

### 3.2 Overview of our system

The architecture of *MuReX* is illustrated in Figure 2. It consists of five major phases in order to address the problems associated with the four subtasks introduced above in Section 3.1. Given a specified query topic for the documents, we preprocess all relevant input texts, in order to make the subsequent operations more efficient. This entails applying a series of natural language processing methods, including document ranking, coreference resolution, and sentence ranking.

The subsequent core of our approach consists of four steps as follows:

- First, we invoke three open information extraction systems to extract candidate facts from the preprocessed text documents, resulting in subject–predicate–object triples. During the extraction, we apply several straightforward transformations to correct two types of common errors, particularly wrong boundaries and uninformative extractions, caused by the syntactic analysis in the extraction approaches.

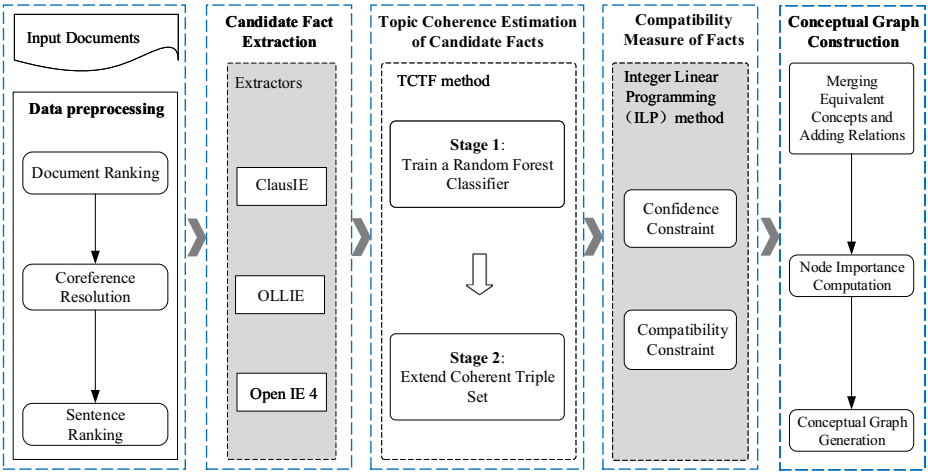



**Figure 2** The overall framework of *MuReX* system

- In the set of candidate facts extracted from the previous phase, unfortunately, many are correct but turn out to be irrelevant to the specified document topic. Hence, we propose a candidate triple filtering (TCTF) method based on an improved self-learning framework to filter out irrelevant triples depending on how coherent they are for the given topic of interest. TCTF consists of two stages, and the procedure operates in an iterative manner until a convergence criterion is met.
- Aggregating all the facts from the filtered results would produce a certain degree of incoherence. In the third step, we avoid this by proposing an integer linear programming (ILP) method for local compatibility to capture the most confident and compatible facts from the filtered results.
- Finally, we create an output graph suitable for presentation to the user. For this, we aggregate the most confident and compatible facts chosen earlier into an initial graph by further merging equivalent entities and concepts, and adding synthetic relations. Then, we adapt this graph using the heuristic of iteratively removing the weakest entities and concepts with relatively lower importance scores, computed using an extended TextRank algorithm. This ensures that the final conceptual graph only consists of facts likely to represent meaningful and salient relationships.

More details of these phases will be given in Section 4.

### 3.2.1 User interface of the system

The system is configured to work with large unstructured document collections. As a first step, the user selects from the overall document collection a specific smaller set of documents that are to be scrutinized. To achieve this, the user can provide a set of query keywords to find relevant documents that can be selected. The overall process of producing a visualization with this system consists of multiple steps, which the interface supports as follows:

- Allowing users to provide query words relating to their search intents or interests, enabling them to browse the document collection while also discerning related documents.
- Given the user query, the system then invokes the processing steps described earlier in Section 3.2 to produce a set of facts that constitute a conceptual graph, consisting of salient and coherent concepts and relationships.
- Users are able to explore the facts via a graph-based visualization that illustrate the connected facts and facilitates exploring the graph. It is also possible to save and share such facts. This final component of the system is described in Section 6 in more detail.

## 4 Multi-document semantic extraction

In this section, the core algorithms for our five main subtasks outlined in Section 3 are described in full detail. The first of these is the preprocessing performed for the input news documents. This is followed by our Open IE approach that exploits the results of multiple extraction systems with additional transformations to automatically extract high-quality candidate facts from the preprocessed text documents. Subsequently, we describe our two-stage candidate triple filtering (TCTF) approach to further filter out the large number of irrelevant and meaningless triples with respect to the topic of interest, as well as our integer linear programming (ILP) approach as a local compatibility measure to eliminate incoherent facts from the filtering results and integrate them in the form of an initial graph. Finally, we describe our heuristic strategy to generate a conceptual graph suitable for presentation to the user, consisting only of facts likely to represent meaningful and salient relationships.

### 4.1 Data preprocessing

Bearing in mind our first subtask is a relation extraction task applied to news documents, it is crucial to preprocess the input texts. In this work, the goal of the data preprocessing is to enhance the subsequent readability and choose sentences with sufficiently high saliency from each document, which are more likely to exhibit a high degree of relevance and contribution towards the core ideas expressed in the document. We rely on the following series of natural language processing methods:

**Step 1: Document Ranking.** Given a specified query topic $T$ and a document collection $D_t = \{d_i \mid i = 1, ..., N\}$ for $T$, the system first selects relevant words or sequences of words (chunks) with sufficiently high frequency appearing in $d_i \in D_t$ as topic words $W^i = \{w_j^i \mid j = 1, ..., M\}$ $(1 \leq i \leq N)$. As candidates, we consider all categories of nouns, adjective – noun combinations (JJ + noun), noun – preposition or subordinating conjunction – noun combinations (noun + IN + noun), other noun phrases (NP), gerund verbs (VBG), prepositional phrases (PP), and named entities (NE). The Stanford CoreNLP system [27] is used for sentence splitting, part-of-speech (POS) labeling, lemmatization, and named entity recognition. Stop words such as "the", "an", "and" with the highest frequency are also removed during this procedure. For each word $w_j^i \in W^i$, we compute its weight via traditional TF-IDF. $D_t$ are ranked according to the document score $score(d_i)$, which is computed as the sum of TF-IDF weights for the entire set of topic words in $d_i$. By default, the top-$k$ documents $D_k \subseteq D_t$ are selected for further processing, denoted by $D_k = \{d_i \mid i = 1, ..., K\}$.

**Step 2: Coreference Resolution.** Pronouns such as "she" are ubiquitous in language and thus entity names often are not explicitly repeated when new facts are expressed in a text. To nevertheless interpret such textual data appropriately, it is hence necessary to resolve pronouns, for which we again rely on the Stanford CoreNLP system [27]. Thus, an occurrence of the pronoun "she" may be replaced by "Angela Merkel", for instance. We perform coreference resolution, which introduces coreference links between mentions that refer to the same entity or concept within $d_i \in D_k$. Note that after individual documents $d_i$ have been processed, cross-document co-reference over $D_k$ is then achieved via simple named entity string matching.

**Step 3: Sentence Ranking.** Different sentences within an article tend to exhibit a high variance with regard to their degree of relevance and contribution towards the core ideas expressed in the article. To address this, our system computes TextRank importance scores [30] for all sentences within $d_i \in D_k$. It then considers top-$k$ sentences with sufficiently high scores, denoted by $S^i = \{s_j^i \mid j = 1, ..., K'\}(1 \leq i \leq K)$.

After the procedure of data preprocessing, we can obtain a collection $S_k = \{S^i \mid i = 1, ..., K\}$, where $S^i$ represents a collection of top-k sentences in $d_i$.

### 4.2 Candidate fact extraction

Similar to knowledge graphs [18] such as Freebase [4], YAGO [42], and WordNet [31], conceptual graphs also consist of a set of triples of the form (*concept phrase*, *relation phrase*, *concept phrase*), where *concept phrase* stands for an entity or concept, and *relation phrase* stands for the relation between the two *concept phrases*. As a crucial step for conceptual graph induction, we therefore rely on accurate information extraction to harvest high-quality

candidate relational triples. Traditional IE systems require the pre-specification of a set of relations of interest, and are thus not effective for coping with the diverse sets of relationships one may encounter in news articles from diverse domains. We thus consider an *open information extraction* (Open IE) approach [2] to extract relational triples of the form (*noun phrase*, *relation phrase*, *noun phrase*) from the given collection of news texts. Note that our approach also needs to support an unbounded range of noun phrase concepts (e.g., *"the snow storm on the East Coast"*) in addition to the open-ended inventory of potential relationships with explicit relation labels (e.g., *"became mayor of"*). The latter are extracted from verb phrases as well as from other syntactic constructions. While existing Open IE systems achieve promising degrees of accuracy on benchmarks, in practice, they often produce noisy extractions. We thus invoke a combined extractor to simultaneously leverage three popular state-of-the-art Open IE systems. Specifically, we rely on ClausIE [9], OLLIE [36], and Open IE 4 [28], to process all of the selected sentences from $S_k$ from the last phase. As our output we consider only the intersection of triples emitted simultaneously by all considered systems for incorporation into the collection of candidate facts.[5]

Because extraction systems mainly rely on dependency parsing to generate syntactic analyses that guide the relational triple extraction, we can easily encounter two main kinds of errors: (1) **Incorrect boundaries**, especially when triples with conjunctions in the sentence were not properly segmented, which the Open IE approach sometimes fails to do; (2) **Incoherent extractions**, which often occur in the *relation phrases*. The *relation phrases* may be expressed by a combination of a verb with a noun, e.g., light verb constructions. However, adjacent words in the *relation phrase* may, in fact, be distant in the original sentence, e.g., we may obtain a triple such as (*"the people of Tamale"*,*"could contribute mobile application"*, *"on an interactive online platform"*) from the sentence *"the people of Tamale could access, track and monitor the Tamale Metropolitan Assembly projects and plans online at www.opengov.org.gh, and could also contribute on an interactive online platform integrated with mobile application"*.

To mitigate the above issues, we invoke several transformation operations to correct them: (1) **Chunking of triples with conjunctions**. We break down triples with conjunctions in either *noun phrase* into separate triples; (2) **Word order constraints**. We define certain word order constraints. For instance, the noun phrases in a triple of the form (*noun phrase 1*, *relation phrase*, *noun phrase 2*) are considered ordered. All words in *noun phrase 1* must appear before all words in the *relation phrase*, and all the words in *noun phrase 2* must appear after the *relation phrase*. The order of words appearing in the *relation phrase* must be consistent with how they appear in the original sentence. In addition, each *relation phrase* must have appeared in the original sentence, without word modifications or added words. While these operations may remove certain acceptable triples, overall they lead to greatly improved sets of extractions.

### 4.3 Topic coherence estimation of candidate facts

Once the above extraction process has been completed, numerous candidate triples $F_c = \{f_i \mid i = 1, 2, ..., M\}$ are obtained. However, we remain oblivious of which candidate triples are pertinent for the user-specified topic of interest. For example, we may know that the relational triple *("Trump", "will visit", "Apple Inc.")* is a correct one, but it is irrelevant for

[5]Given a relational triple extracted by ClausIE, OLLIE, or Open IE 4, only when its confidence is greater than 0.85 is it judged as being a suitable extraction.

the news topic *US presidential election*. Hence, the relevance of triples needs to be assessed from the data with minimal human guidance. To achieve this, we propose a two-stage candidate triple filtering (TCTF) approach based on an improved self-learning framework for gradually discovering more and more coherent triples and correct information from the candidates for the target topic.

The underlying idea of the TCTF approach is that coherent triples for a specified topic should reflect cohesive semantics and characterize the topic with regard to different aspects, e.g., for a news topic such as *US presidential election*, a candidate triple *("Hillary", "as a powerful competitor of", "Trump")* can be viewed as a highly coherent one due to both its *subject* and *object* appearing in the topic word list. Recall that the identification of topic words is based on a sufficiently high TF-IDF score, as described earlier for the document ranking step in Section 4.1. In order to measure the topic coherence for each $f_i \in F_c$ in different aspects, we define several features, divided into three groups, i.e., *topic features*, *text features*, as well as *source features*.

However, without supervision, it is hard to assess whether the triple is coherent enough with the topic. To overcome the lack of supervision, TCTF exploits an improved self-learning framework that can automatically generate coherent triples from the candidates and the design runs with little human guidance.

An overview of the features used in the system is given in Table 1. In the following, we describe some of the less obvious ones in more detail. Here, $f$ refers to the candidate fact under consideration, while $l$ refers to the overall candidate list.

- *Redundancy* is defined as the ratio of the number of candidates that have the same *subject*, *relation*, and *object* as the candidate triple $f$ to the total number of candidates. These extracted redundant facts can be from different sentences within and across the documents.
- *Similarity*: This is defined as the ratio of the number of candidates that are similar to the candidate triple $f$ over the total number of candidates, i.e.,

$$\mathrm{sim}(f, l) = \frac{n_{\mathrm{sim}}(f, l)}{n(l)} \tag{1}$$

  where $n_{\mathrm{sim}}(f, l)$ denotes the number of candidates that are similar to $f$ among all candidates, and $n(l)$ is the total number of candidates.
- *Relation_Context* is the ratio of the size of context of the relation $r$ in the candidate triple $f$ over the total number of candidates, i.e.,

$$\mathrm{RelCxt}(r_f, l) = \frac{n_{\mathrm{context}(r_f)}}{n(l)} \tag{2}$$

  where $n_{\mathrm{context}(r_f)}$ denotes the size of the *relation context* of $r$ in $f$, which consists of the candidates that have the same *relation type* as $f$, while $n(l)$ refers to the total number of candidates.
- *Compatibility*: The compatibility between the *relation context* of the relation $r$ in a candidate triple $f$ and the semantic information of $f$ itself, i.e.,

$$\mathrm{Cmp}(r_f, f_{ht}) = (1 - \epsilon)\,\mathrm{RelCxt}(r_f, l) + \epsilon\,\mathrm{Sem}(f_{ht}, \mathrm{context}(r_f)) \tag{3}$$

  Here, the first term $\mathrm{RelCxt}(r_f, l)$ denotes the *relation context* of $r$ in $f$ as computed in Equation 2; the second term denotes the ratio of the number of candidates that have the same *subject* or *object* with $f$ from $\mathrm{RelCxt}(r_f, l)$, which is computed as $\mathrm{Sem}(f_{ht}, \mathrm{context}(r_f)) = \frac{n_{f_{ht}}}{n_{\mathrm{context}(r_f)}}$. Parameter $\epsilon$ is used for smoothing as well as to control the influence of the *relation context*, and is fixed to 0.5 in our implementation.

**Algorithm 1** A two-stage candidate triple filtering approach based on an improved self-learning framework

**Input:** $F_c$: Unlabeled triple set (i.e., Candidate triple set);
$F_{\text{train}}$: a small fraction of labeled training triple set for $F_c$;
$F_{\text{positive train}}$: positive triples in training set;
$F_{\text{predicted}}$: predicted positive triple set;
$F_{\text{last}}$: positive triple set in the last iteration;
$k_{\text{F}}$: flag for the first update for the model;
$\alpha$: a fixed threshold;
$\eta$: learning rate;
$\varepsilon$: a parameter;
$k_{\max}$: the maximum number of iterations.
**Output**: $F_t$: Combined coherent triple set;

1. $F_{\text{last}} \leftarrow \emptyset$, $F_{\text{predicted}} \leftarrow \emptyset$, $F_{\text{combined}} \leftarrow \emptyset$, $F_{\text{last}} \leftarrow \emptyset$; ▷ **Initialization**
2. $\theta_+ \leftarrow 0$, $k_{\text{F}} \leftarrow \texttt{true}$, $k \leftarrow 0$;
3. $M \leftarrow$ Learn random forest classifier from $F_{\text{train}}$; ▷ **First stage: Train a Random Forest Classifier**
4. $\theta \leftarrow$ $\text{F}_1$-score from $M$;
5. **repeat** ▷ **Second stage: Extend Coherent Triple Set**
6. $k \leftarrow k + 1$; ▷ iteration counter
7. **if**$\neg k_{\text{F}}$ **then** ▷ update if not first iteration
8. $\theta \leftarrow \theta_+$;
9. $M \leftarrow M_+$;
10. **end if**
11. **for** each candidate triple $f \in F_c$ **do** ▷ get high-quality fact candidates
12. $c \leftarrow$ compute confidence score for $f$ using $M$;
13. **f** $c > \alpha$ **then**
14. $F_{\text{predicted}} \leftarrow F_{\text{predicted}} \cup \{f\}$;
15. **end if**
16. **end for**
17. **if** $F_{\text{last}} \neq \emptyset \wedge |F_{\text{predicted}}| \leq |F_{\text{last}}|$
18. $\alpha \leftarrow \alpha - \eta$; ▷ lower threshold
19. **else**
20. $F_{\text{last}} \leftarrow F_{\text{predicted}}$;
21. **end if**
22. $M_+ \leftarrow$ retrain on $F_{\text{train}} \cup F_{\text{predicted}}$;
23. $\theta_+ \leftarrow$ new $\text{F}_1$ score of $M_+$;
24. $k_{\text{F}} \leftarrow \texttt{false}$;
25. **until** $\theta_+ - \theta < \epsilon$ or $k > k_{\max}$
26. $F_t \leftarrow F_{\text{predicted}} \bigcup F_{\text{positive\_train}}$;
27. **return** $F_t$

The flow of the TCTF approach is formalized as Algorithm 1. In the first stage, we randomly select a small fraction of triples $F_{\text{train}} \subseteq F_c$ that serves as a seed example set for the specified topic. We divide this into a training set, validation set, and test set with a 8:1:1 ratio. We then train a random forest classifier $M$ over $F_{\text{train}}$ and obtain an $\text{F}_1$-score $\theta$ (Lines 3–4). Motivated by previous work [41], which leverages topic words to induce document

**Table 1** The features for candidate triples classification

| # | Advanced Features | Comment | Value Range |
|---|---|---|---|
| Topic Features | | | |
| 1 | Is_Topic_Word | whether both subject and object in a candidate fact occur in the topic words list | 0 or 1 |
| 2 | Is_Subject_tw | whether subject in a candidate fact occurs in the topic words list | 0 or 0.5 |
| 3 | Is_Object_tw | whether object in a candidate fact occurs in the topic words list | 0 or 0.5 |
| 4 | Redundancy | the ratio of candidates redundant with the candidate fact | [0, 1] |
| 5 | Similarity | the ratio of candidates similar to the candidate fact | [0, 1] |
| 6 | Relation_Context | the ratio of candidates involving the same type of relation as the candidate fact | [0, 1] |
| 7 | Compatibility | the compatibility between the relation context of the candidate triple and the semantic information itself | [0, 1] |
| Text Features | | | |
| 8 | Is_In_Title | whether a candidate triple appears in the document title | 0 or 1 |
| 9 | Is_In_Abstract | whether a candidate triple appears in an automatic summarization of the document | 0 or 1 |
| 10 | Is_In_MaxSent | whether a candidate triple appears in the sentence with maximum TextRank importance score in the document | 0 or 1 |
| 11 | Sum_tfidf | sum of TF-IDF of subject and object in the candidate triple in the relevant documents | [0, 1] |
| 12 | Avg_tfidf | average of TF-IDF of subject and object in the candidate triple in the relevant documents | [0, 1] |
| Source Features | | | |
| 13 | Source_Num | the number of sources the candidate triple is extracted from | 1 or 2 |
| 14 | Sentence_Num | the number of sentences the candidate triple is extracted from | 1, 2, ... 50 |
| 15 | Relevant_Docs | the ratio of documents that contain the candidate triple | [0, 1] |

representations, in our setting, we also use topic words to label the training set $F_{\text{train}}$. The triples for which both subject and object occur in the topic words list are labeled as positive examples $F_{\text{positive\_train}}$ (i.e., the initial coherent triples set), while those where neither appear in that list are labeled as negative examples. To avoid over-fitting, negative examples that are close to any positive examples are removed from the training set.

In the second stage, based on the trained classifier $M$, the algorithm computes a confidence score for each unlabeled triple $f \in F_c$ using the classifier's confidence (Line 12) and regards those triples that are assigned a score greater than a threshold $\alpha$ as the predicted coherent triples. Different from existing works, which directly add such predicted triples to the coherent triple set $F_{\text{positive\_train}}$ as actual ones for the next iteration, we add it into the predicted positive triple set $F_{\text{predicted}}$ (Line 14). This is because if the predictions contain errors, such errors could accumulate by cascading down the pipeline, potentially entailing more substantial errors that result in a poor overall performance of the model (i.e., the repeatedly trained random forest classifier). After this step, the algorithm compares $F_{\text{predicted}}$ with the previously found positive triples $F_{\text{last}}$ and sets $\alpha = \alpha - \eta$ if $F_{\text{predicted}} \neq \emptyset$ and $F_{\text{predicted}} \leq F_{\text{last}}$. Otherwise, it sets $F_{\text{last}} = F_{\text{predicted}}$ (Lines 17–21). As we are relying on self-learning, we initially keep a high threshold $\alpha$ so that only the highest-confidence instances serve as training data, but in subsequent rounds, this threshold is lowered to gradually incorporate more diverse examples.

Finally, our algorithm retrains a better classifier $M_+$ on $F_{\text{train}} \bigcup F_{\text{predicted}}$. Correspondingly, the parameters of $M_+$ are tuned according to the precision metric, and we obtain a new $F_1$-score $\theta_+$ (Line 23). We compute $\theta_+ - \theta$ and compare it to a threshold $\epsilon$. If the result is greater than $\epsilon$, it will enter the next iteration by repeating the previous training–predicting–extraction steps until the stopping criterion is met (i.e., $\theta_+ - \theta < \epsilon$ or $k > k_{\max}$). At that point, the final extended coherent triple set $F_t$ is generated and used for further processing in the next phase of our approach.

### 4.4 Compatibility measure of facts

In practice, aggregating all the facts from the filtered results $F_t$ from the last phase, we observe a certain degree of incoherence. For instance, for a news topic such as *US presidential election*, we obtain relational triples from the results after the filtering procedure *("Trump", "entered", "2016 US presidential race')*, *("Hillary", "is a powerful competitor of", "Trump")*, *("e-mail scandal", "might be affecting", "68 percent of Republicans")*. Intuitively, we notice that the third triple is not closely connected to the first two triples. Our assumption is that more coherent sets of facts are more likely to express highly relevant information (with regard to the query) that frequently appears in the documents. We thus propose a local compatibility measure, which aims at overcoming incoherent facts, seeking to retain only the most confident and coherent facts in the context of the specified topic so as to further enhance the data quality. This is achieved by optimizing for a high degree of compatibility between facts with high confidence. We formalize the joint optimization problem as an integer linear program (ILP) as follows:

$$\max_{\boldsymbol{x},\boldsymbol{y}} \quad \boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{x} + \boldsymbol{\beta}^{\mathsf{T}}\boldsymbol{y} \tag{4}$$

$$\text{s.t.} \quad \mathbf{1}^{\mathsf{T}}\boldsymbol{y} \leq n_{\max} \tag{5}$$

$$x_k \leq \min\{y_i, y_j\} \tag{6}$$

$$\forall\ i < j, i, j \in \{1, \ldots, N\},$$

$$k = \frac{1}{2}(2N - i)(i - 1) + j - i$$

$$x_k, y_i \in \{0, 1\}\, \forall i \in \{1, \ldots, N\}, k \tag{7}$$

Here, $\boldsymbol{x} \in \mathbb{R}^L$, $\boldsymbol{y} \in \mathbb{R}^N$ with $L = \frac{1}{2}(N + 1)(N - 2) + 1$. The $y_i$ are indicator variables for the fact $f_i$: If $y_i$ is true, $f_i$ is selected to be retained. $x_k$ represents the compatibility between

two facts $f_i, f_j \in F_t$ $(i, j \leq N, i \neq j)$, where $F_t = \{f_i \mid i = 1, 2, \ldots, N\}$ is the coherent triple set consisting of $N$ elements generated in the last phase of the system. $\beta_i$ denotes the confidence of $f_i$ [15], and $n_{\max}$ is the number of compatible facts desired by the user. In our experiment, $n_{\max}$ is set to 100. $\alpha_k$ is weighted by similarity scores $\text{sim}(f_i, f_j)$ between two facts $f_i, f_j$. Specifically, $\alpha_k = \text{sim}(f_i, f_j) = \gamma s_k + (1 - \gamma) l_k$, where $s_k, l_k$ denote the semantic similarity and literal similarity scores between the facts, respectively. We obtain $s_k$ from *Align, Disambiguate and Walk* [33], a WordNet-based state-of-the-art approach for estimating the semantic similarity of arbitrary pairs of lexical items. $l_k$ is calculated using the Jaccard index. The weighting factor $\gamma = 0.8$ denotes the relative degree to which the semantic similarity contributes to the overall similarity score, as opposed to the literal similarity. The constraints guarantee that the number of results is not larger than $n_{\max}$. If $x_k$ is true, the two connected facts $f_i, f_j$ should be selected, which entails $y_i = 1, y_j = 1$.

After that, the compatible facts can be further aggregated to form a collection $F'_{\text{t}} \subseteq F_{\text{t}}$, and used for the conceptual graph construction in the next phase of our system.

## 4.5 Conceptual graph construction

### 4.5.1 Merging equivalent concepts and adding relations

In order to establish a single connected graph that is more consistent, we further merge potential entities and concepts in $F'_{\text{t}}$ stemming from the preceding process. This proceeds in two steps: (1) We make use of the Stanford CoreNLP entity linker [40] to identify entity or concept mentions and link them to a knowledge base such as Wikipedia or Freebase. Roughly, in about 30% of cases, we are able to obtain this information for the entities. If two entities and concepts are linked to the same knowledge base entity, we assume them to be equivalent as per this information. For example, *US* and *America* may be linked to the same Wikipedia entity *United_States*; (2) We observed that approximately 50% of all pairs of entities or concepts have labels with a slight similarity in terms of their literal form, suggesting a possible connection. When we cannot obtain sufficient context information, it is difficult to decide whether we should merge them to form a single node with a single label that is appropriate for both of them. For instance, we may have pairs such as *all the Democratic candidates* and *US Democratic Parties*. To decide whether to identify them requires more human-crafted knowledge. For this, our system allows the user to connect entities and concepts. To support the annotators, once again the *Align, Disambiguate and Walk tool*[6] is used for semantic similarity computation between concepts for coreference.

Typically, after these steps there are only very few subgraphs that remain. For each topic, annotators can thus additionally add a small number of synthetic relations with freely defined labels to connect these subgraphs into a fully connected graph $G'$.

### 4.5.2 Node importance computation

A relational triple is more likely to be salient if it involves important entities and concepts of the sentence. Motivated in part by the considerations given by Yu *et al.* [53], we illustrate the node importance computation, seeking to retain only the most salient facts to include in the final concept graph for different document topics. Formally, let $G' = (\mathcal{V}, \xi)$ denote

[6] https://github.com/pilehvar/ADW

a weighted directed graph generated in the preceding step, where $\mathcal{V} = \{v_1, v_2, ..., v_R\}$ represents a set of preferred nodes that correspond to entities and concepts in $G'$, and $\xi$ is a directed edge set, associated with each directed edge $v_i \rightarrow v_j$ representing a directed relationship originating from $v_i$ to $v_j$. We assign a weight $w_{ij} = 1$ to $v_i \rightarrow v_j$ and its reverse edge $v_j \rightarrow v_i$ is given $w_{ji} = 0.5$. By adding lower-weighted reverse edges, we can analyze the relationship between two nodes that are not connected by directed links while maintaining a preference toward the original directions.

TextRank [30] is a ranking algorithm that can be used to compute the importance of each node within $G'$ based on graph random walks. Similarly, suppose a random walker keeps visiting adjacent nodes in $G'$ at random. The expected percentage of walkers visiting each node converges to the TextRank score. We assign a higher preference toward these nodes when computing the importance scores, since such entities and concepts are more informative for $G'$. We extend TextRank by introducing a new measure called "back probability" $d \in [0, 1]$ to determine how often walkers jump back to the nodes in $\mathcal{V}$ so that the converged score can be used to estimate the relative probability of visiting these preferred nodes. We defined a preference vector $\mathbf{p}_R = \{p_1, p_2, ..., p_{|\mathcal{V}|}\}$ such that the probabilities sum to 1, and $p_k$ denotes the relative importance attached to $v_k$. $p_k$ is set to $\frac{1}{|\mathcal{V}|}$ for $v_k \in \mathcal{V}$, otherwise 0. We finally define $I$ as $1 \times |\mathcal{V}|$ importance vector to be computed over all nodes in $G'$ as follows.

$$I_i = (1 - d) \sum_{j \in \mathcal{N}(i)} \frac{w_{ji}}{\sum_{k \in \mathcal{N}(j)} w_{jk}} I(j) + d\, p_i, \tag{8}$$

Here, $\mathcal{N}(i)$ stands for the set of the node $v_i$'s neighbors.

#### 4.5.3 Conceptual graph generation

We assume there is a constraint $n_{\max}$ on the maximal number of concepts in the conceptual graph (configured to 200 in our system). We rely on a heuristic to find a graph that is connected and satisfies the size limit of $n_{\max}$ concepts: We iteratively remove the weakest concepts with relatively lower importance score computed using Equation 8 until only one connected component of at most $n_{\max}$ entities and concepts remains, which is used as the final conceptual graph $G$. This approach guarantees that the graph is connected with salient concepts (though it might not find the subset of concepts that has the highest total importance score).

## 5 Experiment

In this section, we first introduce the datasets and experimental setup, including the annotation of ground truth data, compared Open IE baseline methods, and evaluation metrics. Then, we conduct extensive experiments to evaluate the effectiveness of our proposed major algorithms in the system. Finally, the experimental results are discussed, including: (1) analysis of candidate fact extraction, (2) analysis of candidate triple classification, and (3) investigation of the quality of the final generated conceptual graph towards different document topics on its coverage rate of topic entities and concepts, confidence score, and the compatibility of involved facts.

### 5.1 Datasets

The dataset used in our system includes 5 categories, and for each category we have 3 popular events, each of which represents a query topic. Every topic cluster comprises approximately 30 documents with on average 1,316 tokens, which leads to an average topic cluster size of 2,632 tokens. This is 3 times larger than typical Document Understanding Conference (DUC)[7] clusters of 10 documents. With these properties, our dataset presents an interesting challenge towards real-world application scenarios, in which users typically have to deal with much more than 10 documents. The documents in our dataset stem from a larger news document collection[8] released by Signal Media as well as from Web news articles that we crawled. We rely on event keywords to filter them so as to retain related ones for different topics. The overall statistics of the resulting dataset are shown in Table 2.

### 5.2 Experimental setup

#### 5.2.1 Annotation of ground-truth data

For the sentence-level extraction task (i.e., candidate fact extraction), we first randomly sampled 10 documents for every query topic (150 documents in total) and performed coreference resolution. Then, once again a random sample of 10 sentences from each extracted document (1,500 sentences in total) is selected for further analysis. Each sentence is examined by three expert annotators with NLP background independently to annotate all of the correct triples. A triple is annotated as correct if the following conditions are met: (1) it is entailed by its corresponding clause; (2) it is reasonable or meaningful without depending on any context, for example, *("e-mail scandal", "is relevant to", "Hillary Clinton")* will be regarded as correct as long as it matches the statements expressed in the sentence. In contrast, triples that cannot be read smoothly are treated as incorrect extractions, for example, *("Hillary Clinton", "leads", "")*, *("Trump", "is", "is the candidate of 2016 US presidential race')*, or *("Republican", "received", "3 percent and')*, will not be counted, since they have mistakes at the word segmentation level or may in fact be meaningless. (3) All three annotators must label it as correct simultaneously (The inter-annotator agreement was 82%, Randolph's free-marginal multirater $\kappa = 0.60$). All triples with conflicting labels results were disregarded.

#### 5.2.2 Compared open IE baseline methods

In order to evaluate our extraction approach more comprehensively, a number of competitive Open IE baseline systems are compared, including the following:

- **ClausIE**[9] [9]: This system depends on a predefined set of rules on how to extract relations instead of learning extraction patterns. It identifies and classifies clauses into clause types, and then generates extractions based on these clause patterns.

[7] https://duc.nist.gov/

[8] http://research.signalmedia.co/newsir16/signal-dataset.html

[9] https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/ambiverse-nlu/clausie/

**Table 2** Dataset description

| Category | Topic ID | Document topic | Time period | Docs | Doc. Size | Source |
|---|---|---|---|---|---|---|
| Armed conflicts and attacks | 1 | Syria refugee crisis | 2015-09-01 - 2015-09-30 | 30 | 2179 ± 506 | News, Web |
| | 2 | North Korea nuclear test | 2017-08-09 - 2017-11-20 | 30 | 1713 ± 122 | News |
| | 3 | THAAD defence deployment | 2017-05-11 - 2017-07-08 | 30 | 980 ± 154 | News, Web |
| Business and economy | 4 | Chinese cooperation with Sudan | 2015-09-01 - 2015-09-30 | 30 | 768 ± 132 | News, Web |
| | 5 | OPEC oil | 2016-11-10 - 2017-01-10 | 30 | 1364 ± 461 | News, Web |
| | 6 | Trump TPP | 2016-12-23 - 2017-02-23 | 30 | 879 ± 306 | News |
| Politics and elections | 7 | US presidential election | 2016-06-14 - 2016-08-14 | 30 | 1175 ± 207 | News, Web |
| | 8 | US-China trade war | 2018-03-23 - 2018-06-15 | 30 | 2412 ± 542 | News, Web |
| | 9 | Trump tax | 2017-03-26 - 2017-05-30 | 30 | 1729 ± 480 | News |
| Arts and culture | 10 | Nobel prize | 2016-09-08 - 2016-11-08 | 30 | 568 ± 126 | News, Web |
| | 11 | Muslim culture | 2013-02-01 - 2013-05-01 | 30 | 972 ± 161 | News, Web |
| | 12 | Turing Award winner | 2019-03-15 - 2019-04-01 | 30 | 1563 ± 464 | News, Web |
| Information technology and application software | 13 | Next-generation search engine | 2016-11-07 - 2017-01-03 | 30 | 729 ± 280 | News, Web |
| | 14 | Program repair for Android system | 2018-02-01 - 2018-05-10 | 30 | 772 ± 453 | Web |
| | 15 | Software repository management | 2018-05-16 - 2018-07-16 | 30 | 1412 ± 376 | Web |

- **OLLIE**[10] [36]: This system learns syntactic and lexical dependency parse tree patterns for relation extraction. Additionally, it is extended optionally to capture contextual information about conditional truths and attributions for extractions.
- **Stanford OpenIE**[11] [1]: This system implements relation extraction by breaking long sentences into short, coherent clauses, and then finding the maximally simple relational triples that are warranted given each of these clauses.
- **Open IE 4**[12] [28]: This system uses bootstrapped dependency parse paths to extract relational triples from a Semantic Role Labeling (SRL) structure. It is fairly efficient and obtains a good balance between precision and recall.
- **MineIE**[13] [14]: This system is built on top of ClausIE. Its purpose is to achieve useful, compact extractions with high precision and recall. It thus minimizes overly specific constituents, and generates additional extractions to capture implicit relations.
- **Ours**$_{\text{part}}$: This method adopts a more lenient triple acceptance principle than the full method. If *any two* Open IE extractors emit a fact with sufficient confidence (as defined for our regular method), a relational triple is accepted. We use the sub-script *part* to denote this setting.
- **Ours**$_{\text{without\_coref}}$: This is a variant of our method, for which we conduct our extraction based on the original texts without having performed coreference resolution. We use the sub-script *without_coref* to denote this setting.
- **Ours**$_{\text{without\_trans}}$: This is a variant of our method, for which we conduct our extraction without any transformations such as chunking of triples with conjunctions or word order constraints. We use the subscript *without_trans* to denote this setting.
- **Ours**: This denotes our full default fact extraction method as introduced earlier in Section 4.2.

Note that systems such as ClausIE [9], OLLIE [36], Stanford Open IE [1], and Open IE 4 [28] typically associate a confidence score with each extraction, which allows downstream applications to trade off precision and recall.

### 5.2.3 Evaluation metrics

As the main evaluation measure, standard information retrieval/IE metrics, i.e., Precision (P), Recall (R), and F-score ($F_1$), are applied. In the case of multiple human judges, Randolph's free-marginal multirater $\kappa$ for measuring the inter-annotator agreement is computed. As to the analysis of the quality of the final generated conceptual graph, we consider the coverage rate of topic entities and concepts, confidence score, and compatibility of involved facts in the graph. The measures are computed as follows:

- Precision (P), Recall (R), and F-score ($F_1$) are standard IE metrics:

$$P = \frac{\#\text{ correct}}{\#\text{ extractions}}, R = \frac{\#\text{ correct}}{\#\text{ relations}}, F_1 = \frac{2PR}{P+R} \tag{9}$$

  Here, "# correct" denotes the number of extractions deemed correct, "# extractions" denotes the total number of extractions, and "# relations" denotes the number of triples annotated as correct extractions[5.2].

[10] http://knowitall.github.io/ollie/
[11] https://nlp.stanford.edu/software/openie.shtml
[12] https://github.com/knowitall/openie
[13] https://github.com/uma-pi1/minie

– The ratio of topic-related entities and concepts, i.e.,

$$\text{TopicCon\_Rate} = \frac{\#\ \text{topic\_concepts}}{\#\ \text{concepts}}, \tag{10}$$

where "# topic_concepts" denotes the number of entities or noun phrase concepts annotated as topic concepts[14], and "# concepts" denotes the total number of all entities and concepts in the conceptual graph.

– Confidence score, i.e.,

$$\text{Avg\_Confidence}(f_i, n) = \frac{\sum_{i=1}^{n} c(f_i)}{n}, \tag{11}$$

where $c(f_i)$ denotes the confidence score[15] of each fact $f_i$, and $n$ is the number of facts involved in the final conceptual graph.

– Compatibility of involved facts in the graph, i.e.,

$$\text{Avg\_Compatibility}(f_i, f_j, n) = \frac{\sum_{i=1}^{n} \sum_{j>i} \sigma(f_i, f_j)}{c_n^2}, \tag{12}$$

where $f_i$ and $f_j$ are any facts in the final conceptual graph, which contains $n$ facts, and $\sigma(f_i, f_j)$ denotes the compatibility between $f_i$ and $f_j$. The latter is similar to Equation 3, and is computed as $\sigma(f_i, f_j) = (1-\epsilon)\,(\text{RelCxt}(r_{f_i}, n) + \text{RelCxt}(r_{f_j}, n)) + \epsilon\,\text{sim}(f_i, f_j)$, where $\text{sim}(f_i, f_j)$ denotes the similarity scores (see Section 4.4) between fact $f_i$ and fact $f_j$. Parameter $\epsilon$ is used for smoothing as well as to control the influence of the relation context, and is fixed to 0.5 in our implementation.

## 5.3 Evaluation and results analysis

### 5.3.1 Analysis of candidate fact extraction

We present the evaluation results of our extraction method and Open IE baseline methods on fifteen document topics in Tables 3, 4, and 5. The major observations from the results can be summarized as follows:

1. **Ours**$_{\text{without\_coref}}$ obtains the worst results among all the methods across all topics. This is because of the many cases in which the selected sentences from these topics are not readable without the context when coreference resolution is omitted. This shows the importance of coreference resolution for more interpretable extractions from the sentences in the document. In addition, all methods achieve F-scores of lower than 60% on Topics 3, 9, and 14, and are not effective for our extraction task. Through our observation of the corresponding data pertaining to these topics, the reason is that the original document subsets show substantial use of non-standard language and grammatical errors in the sentences. This hurts the performance of the fact extraction step.

[14] An entity or concept is regarded as a topic concept if it occurs in the topic words list as described in Section 4.1.

[15] For popular OpenIE systems such as ClausIE, OLLIE, and Open IE 4, we rely on the confidence value computed by each system itself as the confidence score of each of facts.

**Table 3** Average precision (%), recall (%), and F-score (%) of different methods on five independent document topics (Topic 1 to Topic 5). The best and second best results in each column is boldfaced and underlined, respectively. Improvements over the best baseline are shown in the last row.

| News topic | #Topic 1 | | | #Topic 2 | | | #Topic 3 | | | #Topic 4 | | | #Topic 5 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Measure / Open IE method | P | R | $F_1$ | P | R | $F_1$ | P | R | $F_1$ | P | R | $F_1$ | P | R | $F_1$ |
| ClausIE | 0.65 | 0.54 | 0.59 | 0.46 | 0.39 | 0.42 | 0.66 | 0.33 | 0.44 | 0.67 | 0.51 | 0.58 | 0.64 | 0.46 | 0.54 |
| OLLIE | 0.79 | 0.46 | 0.58 | 0.63 | 0.30 | 0.41 | 0.73 | 0.28 | 0.40 | 0.74 | 0.42 | 0.54 | 0.69 | 0.31 | 0.43 |
| Stanford OpenIE | 0.55 | 0.47 | 0.51 | 0.38 | 0.26 | 0.31 | 0.53 | **0.51** | 0.52 | 0.44 | 0.40 | 0.42 | 0.43 | 0.41 | 0.42 |
| Open IE 4 | 0.60 | 0.52 | 0.58 | 0.45 | 0.37 | 0.41 | 0.49 | 0.32 | 0.39 | 0.66 | 0.50 | 0.57 | 0.60 | 0.42 | 0.49 |
| MinIE | 0.82 | 0.58 | 0.68 | 0.60 | 0.43 | 0.50 | 0.75 | 0.42 | 0.54 | 0.79 | 0.58 | 0.67 | 0.75 | 0.44 | 0.55 |
| **Ours**$_{\text{part}}$ | 0.79 | 0.59 | 0.68 | 0.71 | 0.57 | 0.63 | 0.76 | 0.48 | 0.58 | 0.81 | **0.67** | 0.73 | 0.74 | 0.57 | 0.64 |
| **Ours**$_{\text{without\_coref}}$ | 0.35 | 0.27 | 0.30 | 0.37 | 0.19 | 0.25 | 0.38 | 0.17 | 0.23 | 0.41 | 0.24 | 0.30 | 0.33 | 0.18 | 0.23 |
| **Ours**$_{\text{without\_trans}}$ | 0.80 | 0.51 | 0.62 | 0.77 | 0.60 | 0.67 | 0.72 | 0.46 | 0.56 | 0.78 | 0.64 | 0.71 | 0.72 | 0.58 | 0.64 |
| **Ours** | **0.83** | **0.67** | **0.74** | **0.79** | 0.66 | **0.72** | **0.79** | 0.47 | **0.59** | **0.88** | 0.65 | **0.74** | **0.76** | **0.62** | **0.68** |
| Improvement (%) | 1.22 | 13.56 | 9.13 | 2.60 | 1.00 | 6.63 | 3.95 | - | 0.17 | 8.64 | -2.99 | 1.95 | 1.33 | 6.90 | 6.05 |

**Table 4** Average precision (%), recall (%), and F-score (%) of different methods on five independent document topics (Topic 6 to Topic 10). The best and second best results in each column is boldfaced and underlined, respectively. Improvements over the best baseline are shown in the last row.

| News topic | #Topic 6 | | | #Topic 7 | | | #Topic 8 | | | #Topic 9 | | | #Topic 10 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Measure / Open IE method | P | R | $F_1$ | P | R | $F_1$ | P | R | $F_1$ | P | R | $F_1$ | P | R | $F_1$ |
| ClausIE | 0.67 | 0.57 | 0.61 | 0.69 | 0.60 | 0.64 | 0.69 | 0.41 | 0.51 | 0.51 | 0.34 | 0.41 | 0.65 | 0.48 | 0.55 |
| OLLIE | 0.71 | 0.38 | 0.50 | 0.67 | 0.55 | 0.60 | 0.81 | 0.39 | 0.53 | 0.57 | 0.35 | 0.43 | 0.73 | 0.34 | 0.46 |
| Stanford OpenIE | 0.58 | 0.48 | 0.53 | 0.62 | 0.58 | 0.60 | 0.58 | 0.54 | 0.56 | 0.46 | 0.39 | 0.42 | 0.56 | 0.49 | 0.52 |
| Open IE 4 | 0.66 | 0.41 | 0.51 | 0.64 | 0.58 | 0.61 | 0.60 | 0.40 | 0.48 | 0.55 | 0.32 | 0.40 | 0.62 | 0.35 | 0.45 |
| MinIE | 0.78 | 0.68 | 0.73 | 0.80 | 0.64 | 0.71 | 0.78 | 0.49 | 0.60 | 0.67 | 0.40 | 0.50 | 0.79 | 0.65 | 0.71 |
| **Ours**$_{\text{part}}$ | 0.84 | **0.72** | 0.77 | 0.81 | 0.60 | 0.69 | 0.79 | **0.62** | 0.68 | 0.66 | 0.41 | 0.51 | 0.82 | 0.62 | 0.71 |
| **Ours**$_{\text{without\_coref}}$ | 0.39 | 0.13 | 0.20 | 0.42 | 0.21 | 0.28 | 0.35 | 0.18 | 0.24 | 0.28 | 0.14 | 0.19 | 0.48 | 0.16 | 0.24 |
| **Ours**$_{\text{without\_trans}}$ | 0.79 | 0.65 | 0.71 | 0.80 | 0.52 | 0.63 | 0.73 | 0.57 | 0.64 | 0.66 | 0.39 | 0.49 | 0.80 | 0.71 | 0.75 |
| **Ours** | **0.87** | 0.70 | **0.78** | **0.84** | **0.69** | **0.76** | **0.82** | 0.60 | **0.69** | **0.70** | **0.42** | **0.53** | **0.82** | **0.75** | **0.78** |
| Improvement (%) | 3.57 | -2.78 | 0.05 | 3.70 | 7.81 | 6.54 | 2.47 | -3.23 | 0.25 | 4.48 | 2.44 | 3.80 | 2.50 | 5.63 | 4.14 |

**Table 5** Average precision (%), recall (%), and F-score (%) of different methods on five independent document topics (Topic 11 to Topic 15). The best and second best results in each column is boldfaced and underlined, respectively. Improvements over the best baseline are shown in the last row.

| News topic | #Topic 11 | | | #Topic 12 | | | #Topic 13 | | | #Topic 14 | | | #Topic 15 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Measure / Open IE method | P | R | $F_1$ | P | R | $F_1$ | P | R | $F_1$ | P | R | $F_1$ | P | R | $F_1$ |
| ClausIE | 0.59 | 0.48 | 0.53 | 0.76 | 0.59 | 0.66 | 0.63 | 0.38 | 0.47 | 0.48 | 0.37 | 0.42 | 0.69 | 0.50 | 0.58 |
| OLLIE | 0.76 | 0.42 | 0.54 | 0.74 | 0.53 | 0.62 | 0.68 | 0.36 | 0.47 | 0.56 | 0.29 | 0.38 | 0.75 | 0.30 | 0.43 |
| Stanford OpenIE | 0.55 | 0.52 | 0.53 | 0.60 | 0.57 | 0.58 | 0.51 | 0.46 | 0.48 | 0.40 | 0.36 | 0.38 | 0.55 | 0.48 | 0.51 |
| Open IE 4 | 0.70 | 0.47 | 0.56 | 0.62 | 0.56 | 0.59 | 0.57 | 0.36 | 0.44 | 0.44 | 0.35 | 0.39 | 0.66 | 0.45 | 0.54 |
| MinIE | 0.61 | 0.53 | 0.57 | 0.76 | **0.61** | 0.68 | 0.71 | 0.48 | 0.57 | 0.62 | 0.41 | 0.49 | 0.78 | 0.54 | 0.64 |
| **Ours**$_{\text{part}}$ | 0.73 | 0.57 | 0.64 | 0.79 | 0.48 | 0.60 | 0.70 | 0.51 | 0.59 | 0.61 | 0.42 | 0.50 | 0.77 | 0.61 | 0.68 |
| **Ours**$_{\text{without\_coref}}$ | 0.29 | 0.14 | 0.19 | 0.38 | 0.16 | 0.23 | 0.55 | 0.27 | 0.36 | 0.22 | 0.13 | 0.16 | 0.34 | 0.18 | 0.24 |
| **Ours**$_{\text{without\_trans}}$ | 0.77 | 0.54 | 0.63 | 0.78 | 0.47 | 0.59 | 0.70 | 0.54 | 0.61 | 0.63 | 0.46 | 0.53 | 0.76 | 0.60 | 0.67 |
| **Ours** | **0.79** | **0.58** | **0.67** | **0.83** | 0.59 | **0.69** | **0.72** | **0.58** | **0.64** | **0.66** | **0.47** | **0.55** | **0.79** | **0.64** | **0.71** |
| Improvement (%) | 2.60 | 1.75 | 4.49 | 5.06 | -3.28 | 1.91 | 1.41 | 7.41 | 5.38 | 4.76 | 2.17 | 3.25 | 1.28 | 4.92 | 3.29 |

2. We observe that Stanford OpenIE achieves the lowest precision and recall for non-redundant extraction in the subsets from Topics 2, 4, and 5 (apart from the more noisy subsets from Topics 9 and 14), due to its aggressive generation of incorrect and redundant extractions with short relation phrases. We conjecture that it has trouble coping with sentences that contain long relation phrases. In this case, the relation phrase is often split apart, leaving only a short verb, while the details are often lost. As a result, numerous redundant triples from similar sentences are generated. Compared with ClausIE, MinIE achieves better results results on all the topics. Specifically, ClausIE adopts clause patterns to handle long-distance relations, while MinIE refines extractions emitted by ClausIE, avoiding the parts that are considered overly specific (e.g., the extracted relation "is offering only during the meeting" from the sentence "Trump is offering only minor concessions during the meeting."). OLLIE achieves a higher precision compared to Open IE 4 across all topics. This is probably because OLLIE is able to further eliminate less reliable extractions from constructions such as noun-mediated relations with long-distance dependencies. In contrast, Open IE 4 obtains substantially shorter extractions, mainly noun-mediated or verb-mediated relations based on SRL. Often, these are incomplete or not sufficiently informative, especially when extracted from long noun-mediated relation phrases. Compared to both OLLIE and Open IE 4, MinIE is more useful for our extraction task, based on its ability to both minimize overly specific constituents and capture explicit relations. In general, **Ours**$_{\text{without\_trans}}$ brings better performance compared to other baselines, especially for Topics 2, 5, 10, 13, and 14, since it is close to **Ours**, which aggregates the results of three popular extractors.
3. We observe that **Ours** consistently outperforms all baseline methods in terms of F-scores. Specifically, **Ours** obtains F-scores with an average improvement of 9.13%, 6.63%, 6.05%, 6.54%, and 5.38% on Topics 1, 2, 5, 7, and 13, respectively. This can be credited to the following factors:

   (i) **Ours** carefully aggregates the results of three top-performing extraction systems, including ClausIE, OLLIE, and Open IE 4, which is beneficial for achieving better results compared to trusting just a single extractor;
   (ii) Exploiting a few straightforward transformations, we observe that **Ours** is better able to identify the boundary of triples for long sentences with conjunction structure, while other methods, including OLLIE and Open IE 4, often fail at this. Moreover, we also observe the number of incoherent extractions, especially in the relation phrases, can significantly be reduced by using a relaxed constraint for the word order, decreasing the frequency of this type of error from 36.8% to 17.1%. This illustrates the effectiveness of our transformations. It also indicates that the two types of extraction errors as above in **Ours** caused by depending on intermediate structures such as dependency parses can also be eliminated fairly well by such transformations.

4. One interesting observation is that **Ours** does not outperform **Ours**$_{\text{part}}$ that much on all of the topics. Indeed, **Ours**$_{\text{part}}$ achieves relatively higher recall compared to **Ours** on Topics 3, 4, 6, and 8. This is because **Ours**$_{\text{part}}$ adopts a more lenient acceptance criteria for facts, where just two extractors suffice. This leads to the acceptance of a greater number of extractions, although it may also bring in some extra noise. Thus recall increases at the expense of precision.

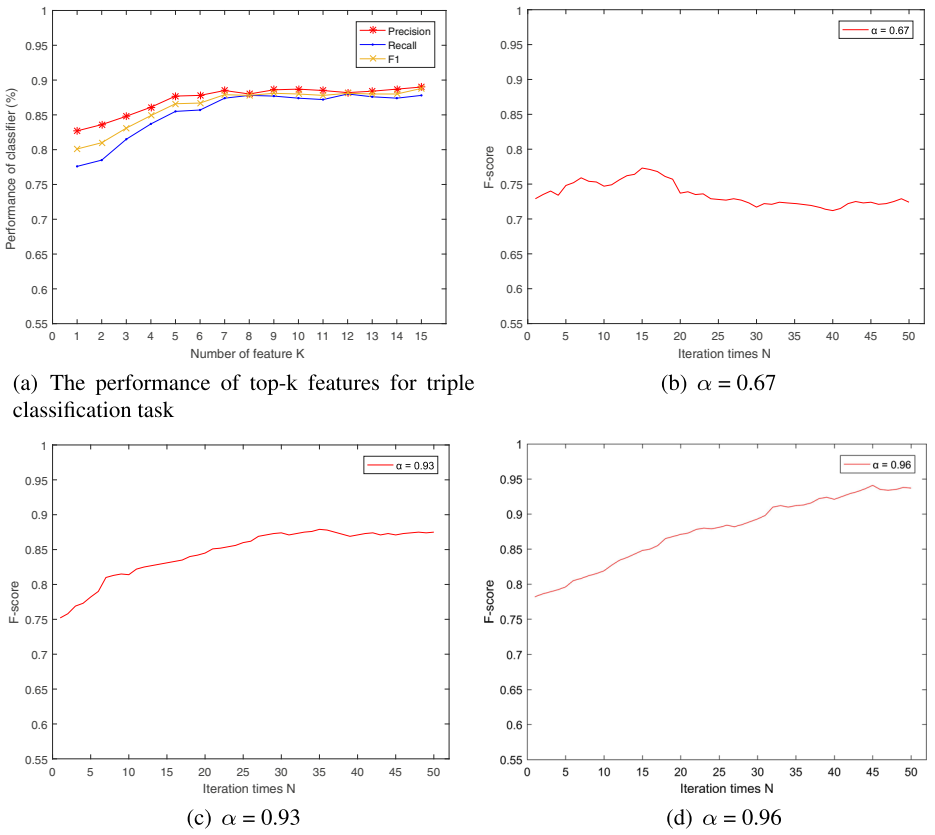
### 5.3.2 Analysis of candidate triple classification

In the process of estimating the topic coherence of candidate facts, although there are sixteen designed features for measuring the topic coherence for each fact from different perspectives, simply combining all of them may not lead to the optimal triple classification quality. Taking the news topic *US presidential election* as an example, we study the effectiveness of each feature for the trained random forest classifier. We selected $\chi^2$ and information gain (IG) [12] as the classification criteria. In Table 6, we rank the features in terms of their $\chi^2$ scores. The results show that Is_Topic_Word is the top-1 feature as ranked by both two measures. Moreover, we further evaluate the contribution of each feature for the classifier when top-$k$ features are used sorted by $\chi^2$ (where the degrees of freedom $v$ is 10, and the significance level with $p$-value $< 0.05$ is used in the $\chi^2$ test). All results, including accuracy, recall, and averaged F-score are reported in Figure 3 (a). We observe that the top-7 features dominate the effectiveness of the classifier, i.e., all measures converge to an upper limit after the top-7 features are leveraged. The results suggest that a few particularly effective features need to be included for strong classification results.

Subsequently, we compared three different thresholds $\alpha \in \{0.67, 0.93, 0.96\}$ for controlling noise and selecting adequate triples at the second stage of the proposed TCTF approach. We observe that, based on the self-learning framework, improper thresholds not only fail to improve the model (i.e., the random forest classifier), but also render the model unable to detect improper triples as incoherent ones for the specified document topic. The reasons for this phenomenon are likely as follows: (1) If the threshold is set to be too small (e.g., $\alpha = 0.67$), the training of the model might easily be affected by noisy data (e.g., added triples), leading to less improvement in the triple classification. The results are shown in Figure 3 (b); (2) If the threshold is set to be too large, on the other hand, the training of the model could take longer. For example, if we set $\alpha = 0.96$, the model converges in a total of approximately 35 epochs. The results are shown in Figure 3 (d). Hence, by considering both training speed and the performance of the self-training process in the TCTF approach, we set the confidence threshold $\alpha$ as 0.93, which achieves a strong performance within fewer training epochs. The results are shown in Figure 3 (c).

**Table 6** Effectiveness of features (sorted by $\chi^2$)

| # | Feature | $\chi^2$ | IG% | # | Feature | $\chi^2$ | IG% |
|---|---|---|---|---|---|---|---|
| 1 | Is_Topic_Word | 53.94 | 2.90 | 11 | Redundance | 16.93 | 0.68 |
| 2 | Is_Subject_tw | 41.32 | 1.72 | 12 | Is_In_Title | 0.82 | 0.51 |
| 3 | Is_Object_tw | 40.02 | 1.73 | 13 | Sentence_Num | 0.74 | 0.06 |
| 4 | Relation_Context | 38.20 | 1.32 | 14 | Relevant_Docs | 0.45 | 1.62 |
| 5 | Similarity | 37.07 | 2.03 | 15 | Source_Num | 0.44 | 0.07 |
| 6 | Compatibility | 23.09 | 2.24 | | | | |
| 7 | Is_In_Abstract | 24.41 | 1.48 | | | | |
| 8 | Is_In_MaxSent | 23.20 | 1.70 | | | | |
| 9 | Sum_tfidf | 20.07 | 0.52 | | | | |
| 10 | Avg_tfidf | 19.58 | 0.48 | | | | |




(a) The performance of top-k features for triple classification task

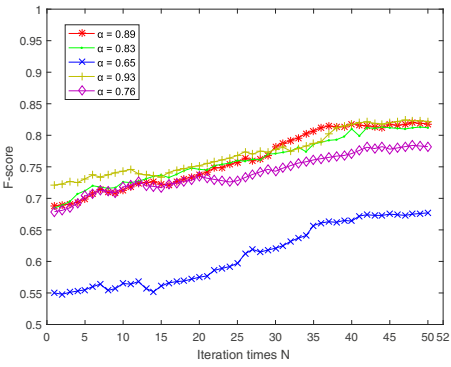
(b) $\alpha = 0.67$

(c) $\alpha = 0.93$

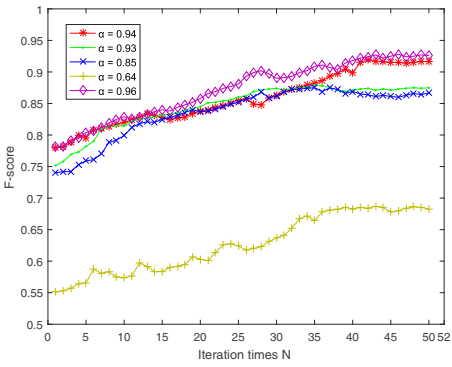
(d) $\alpha = 0.96$

**Figure 3** Evaluation of triple classification and comparison of F-scores of the TCTF approach for different confidence thresholds $\alpha$ across iterations $N$, on the topic "*US presidential election*".

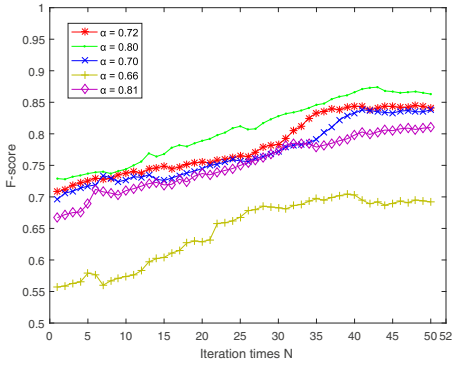
Figure 4 reports the development of the F-scores of the TCTF approach separately for 15 different document topics, considering their respective confidence threshold $\alpha$. We observe that the initial process in the TCTF approach is somewhat stable for different topics, though there are notable differences as well. Subsequently, the scores improve significantly across iterations $N$ until $N$ approximately falls in the range of [35, 50]. After that, the improvement becomes minor and tends to stabilize at a certain level. This indicates that in the initial iterative self-learning loops, noisy initial classifications can easily lead to noisy training data that misleads subsequent training iterations. The more such noisy training data is incorporated, the larger the influence on the model's performance. For example, for Topics 3, 9, and 14, we observe that more epochs are required until stabilizing compared to other topics because the extraction results from the candidate fact extraction phase for these topics may contain more noisy data (Section 5.3.1 considers a similar problem regarding noise in the original data). After several rounds of iterations, the TCTF approach carefully chooses which triple classifications to accept and adjusts the thresholds gradually, and finally guarantees that the models converge for different topics, which proves the effectiveness of the proposed TCTF approach. That is, the approach is capable of retaining only the most coherent triples from the candidates for different topics.




(a) Topic 1 – Topic 5




(b) Topic 6 – Topic 10




(c) Topic 11 – Topic 15

**Figure 4** F-scores of the TCTF approach for different topics across iterations $N$, where each plot shows five topics, each with a different corresponding $\alpha$ value (as given in the legends).

### 5.3.3 Quality analysis of the generated conceptual graph

Finally, after evaluating the extracted relationships, we can obtain the final generated conceptual graphs for different topics. Figure 5 (a)–(e) presents examples of such conceptual graphs, showing the core parts of the graphs obtained for Topics 1, 4, 7, 13, and 15. For Topic 1 ("*Syria refugee crisis*"), for example, we can clearly observe important causal relationships as described by relational triples such as *("a burgeoning refugee crisis in Europe", "is sparked by", "global extremism")*, *("current refugee flow", "is relevant to", "a burgeoning refugee crisis in Europe")*, *("the crisis in Syria", "has extended", "current refugee flow")*.

Considering another example, for Topic 13 ("*Next-generation search engine*"), we can see that the graph provides an overview of challenges and opportunities related to next generation search engines. It also points out that AI techniques play an important role.

Similarly, for Topic 15 ("*Software repository management*"), we can easily observe the importance of software warehouses and database systems, with properties and relations introduced in relational triples such as *("software warehouse", "performs", "source code controlling")*, and *("database system", "runs on", "Unix OS")*.

We can further analyze the quality of the conceptual graphs quantitatively. To this end, we report several metrics in Figure 6. Averaging over the fifteen document topics, the

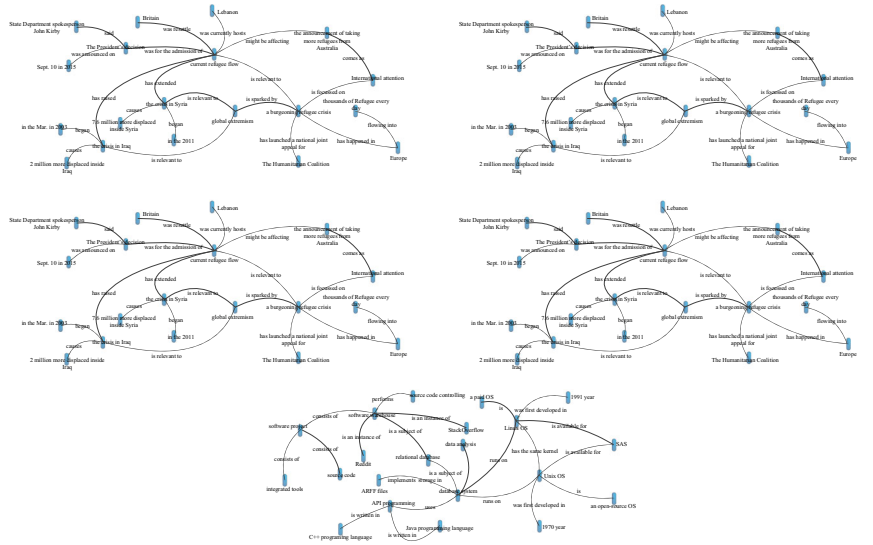


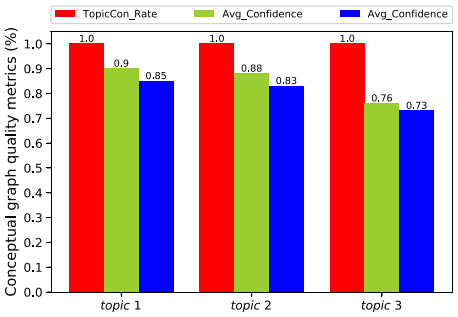
**Figure 5** Example of core parts of the final generated conceptual graphs for five topics.

coverage rate of topic entities and concepts (TopicCon_Rate) is 100%, the confidence score (Avg_Confidence) is 85.9%, and the fact compatibility (Avg_Compatibility) is 79.5%.

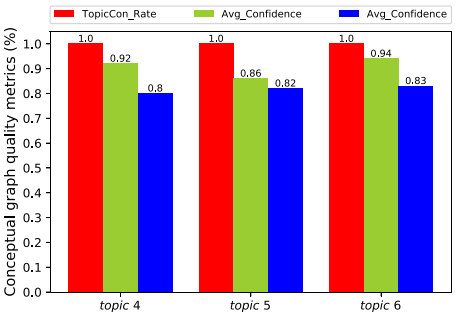
We make the following observations: (1) TopicCon_Rate assesses to what extent the nodes in the graph consist only of ones that pertain to the query topic. A typical news article will lead to a large number of extractions, only some of which are relevant to the query at hand. Our extracted graphs in contrast obtain perfect results, which means that all nodes in the graph are relevant to the query. This suggests that the graphs describe a coherent network of salient relationships. Of course, this result is one that is obtained under the default configuration, in particular, with the maximal number of concepts in the conceptual graph configured to 200. With larger graphs, more irrelevant nodes will be included, which may also be desirable in some circumstances. (2) Across a diverse set of query topics, the proposed TCTF approach is capable of retaining only the most coherent triples from the candidates. This is non-trivial, because different subsets of documents corresponding to the topics may require very different thresholds. In an iterative self-learning loop, noisy initial classifications could easily lead to a noisy training set that misleads subsequent training iterations. Our TCTF approach carefully chooses which classifications to accept and adjusts the thresholds gradually. (3) The facts in the final conceptual graph for different topics have a higher confidence and better compatibility, which demonstrates the importance of our approaches, especially the local compatibility measure, for capturing the most confident and compatible facts. (4) Our method quite robustly obtains appealing conceptual graphs across different topics. In general, it obtains competitive results even if some documents are erroneously classified as pertaining to a topic.

## 6 Interactive visualization

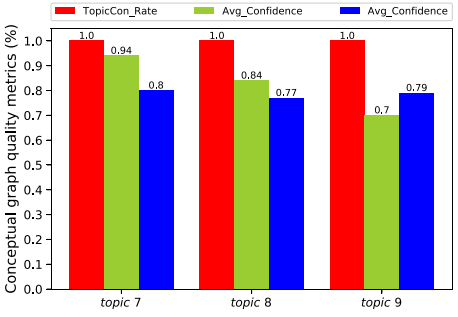
All steps of the processing pipeline within our system, including the processes of candidate fact extraction, filtering, and displaying salient connections from multiple documents,



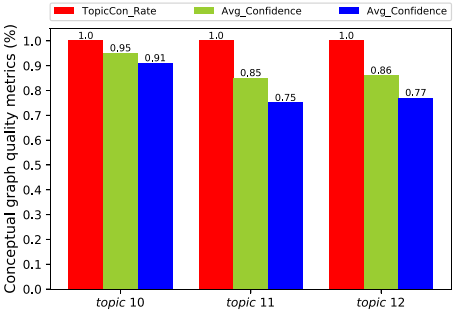

(a) Topic 1 – Topic 3



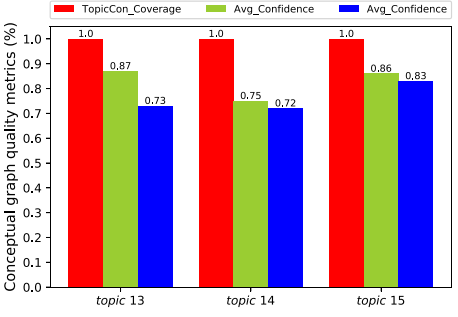

(b) Topic 4 – Topic 6




(c) Topic 7 – Topic 9




(d) Topic 10 – Topic 12




(e) Topic 13 – Topic 15

**Figure 6** Quality analysis of the final generated conceptual graphs for fifteen document topics with regard to three metrics (TopicCon_Rate, Avg_Confidence, and Avg_Compatibility).

are visualized in our system. The user is able to explore the documents with several different views: (1) Topic keywords selection, (2) Document view, (3) Candidate fact extraction view, (4) Fact filtering and aggregation, and (5) Conceptual graph view. A recorded video presenting the system is provided at https://shengyp.github.io/vmse/.

The views are all interactively linked so that the user can start exploring concepts and their relationships by searching for keywords relevant to the topic. In the Document view, the user can assess documents and drill down by document frequency, and any choices made there can be regarded as filters that constrain the current document set. They thus affect the

subsequent views for fact extraction, fact filtering, fact aggregating, and so on, which in turn affect the information displayed in the final visualizations. We now present these views in more detail.

**Topic Keywords Selection.** This view allows the user to pick topics based on keywords with high frequency in the document corpus as queries. By default, the system lists fifteen predefined trending news topics, e.g., the Syria refugee crisis, Chinese cooperation with Sudan, US presidential election, Trump tax, Nobel prize, Program repair for Android system, and so forth. For each topic, the user can select relevant high-frequency keywords, e.g., for the US presidential election, these include "candidate", "Hillary Clinton", "Donald Trump", "Democratic", "Republican", "GOP" as keywords. The topics are completely customizable and the user can provide one or more keywords as input to define new topics, as shown in Figure 7, Part 1.

**Document View.** The document view provides a list of documents ordered by title or ranked by their weight as selected by the currently active filters. For large text collections, the documents are loaded on demand. Users can browse the list and identify documents for closer reading (bold, open folder icon), as in Figure 7, Part 2. The document text view shows the full text of the document, where pronouns and other forms of coreference have been resolved and replaced, and meaningful phrases are automatically highlighted in red, as in Figure 7, Part 4.

Figure 7, Part 3 shows the distribution of documents over a year. It is displayed as a bar chart with logarithmic scale, as this better complies with the exponential characteristics of the document distribution. Users can drill down temporally to consider the document distribution for specific months or days. It is also possible to select a time interval for which the corresponding documents are shown in the document view.

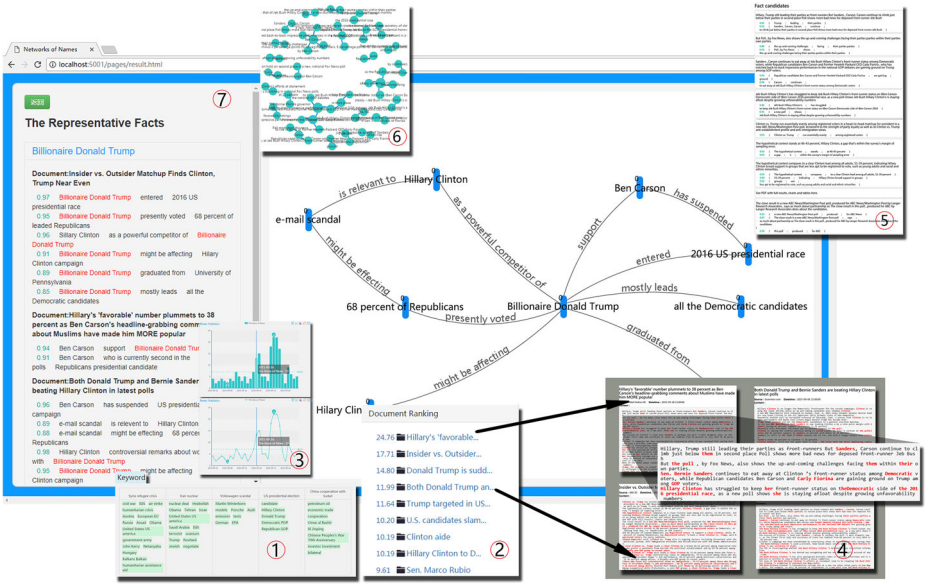



**Figure 7** Screenshots from the user interface of our system

**Candidate Fact Extraction View.** Our system is able to automatically extract all candidate facts from multiple documents by invoking three Open IE systems, along with additional transformations. During this procedure, several specific parameters are customizable, such as the size and step size of a sliding window (used to reduce the computational load), coefficients for semantic similarity, literal similarity, and the number of representative facts. Figure 7, Part 5 shows the result of the fact extraction.

**Fact Filtering and Aggregating View.** In contrast, Figure 7, Part 6 shows a set of entities and concepts as nodes and their connections as links to mark those facts assessed as confident, coherent, and compatible after applying the TCTF approach and local compatibility measure.

**Conceptual Graph View.** The resulting conceptual graph can be filtered such that only the most meaningful and salient connections are maintained. Users can thus more easily explore such strong connections, and user-selected entities or concepts are highlighted. For example, in the left panel, when the user selects the first entity "Billionaire Donald Trump" within the set of representation facts that reflect the current documents' topic, the system presents the pertinent entities, concepts, and relations associated with this concept via a graph-based visualization in the right panel, including "Hillary Clinton" as a prominent figure, as shown in Figure 7, Part 7.

**Fronted and Backend Implementation Details.** Our system is implemented in Java, with Apache Tomcat[16] as the Web server. All data in the backend is stored in a MySQL database[17]. For our frontend, the graph-based visualization of the system is based on Avalon[18], as well as the jQuery[19] framework.

## 7 Conclusion

This paper presents a novel multi-document semantic extraction system, called *MuReX*, to aid users in quickly discerning salient and meaningful facts and connections in a collection of relevant documents, via graph-based visualizations of relationships between concepts even across documents. We collect candidate facts using an Open IE approach, which is capable of extracting high-quality facts as candidates based on multiple existing extraction engines and automatic transformations. We propose a two-stage candidate triple filtering method based on an improved self-learning framework to discern confident and coherent triples from the overall set of candidates. We further select compact and compatible triples from the filtered results by modeling local compatibility, and connecting them in the form of an initial graph. We construct the final conceptual graph using a heuristic that ensures that it only consists of facts and connections likely to represent meaningful and salient relationships. In our experiments, we illustrate that our extraction method achieves a higher F-score on average over several competitive Open IE baseline methods on two real-world news datasets. Besides, our approach can also guarantee a high-quality conceptual graph for

[16] http://tomcat.apache.org/
[17] http://www.mysql.com/
[18] http://avalonjs.coding.me/
[19] http://jquery.com/

different topics in terms of its proportion of topic-related entities and concepts, confidence score, and fact compatibility.

Semantic information extraction from news data is an enduring and interesting problem. Thus our work can be extended in a number of potential future directions. First, in the paper we only considered the salient entities and concepts in the conceptual graph. A straightforward extension is to enrich them with links to information on their external origins, e.g., knowledge bases and social content, based on knowledge linking methods and a concept-based social content alignment method. A second extension concerns the news representation. Inspired by Hou *et al.* [19], who present a link-centric news representation, organizing news at three semantic levels, namely events, topics and entities, we can attempt to induce a three-tiered representation for the news network by integrating different types of news, as well as linking named entities and concepts to public knowledge graphs providing background knowledge, or aligning news articles establishing links between news and social content. A third direction is to improve the robustness of the current system implementation. On the one hand, we may give greater consideration to fact fusion via an automatic procedure in the process of conceptual graph construction. On the other hand, we can further lower the threshold to transfer the approaches from newswire text to arbitrary other domains, including even some low-resource scenarios. Our eventual goal is to develop a unified system framework that can achieve fully automated conceptual graph construction for a wide range of different domains.

**Acknowledgments** This paper was partially supported by National Natural Science Foundation of China (Nos. 61572111 and 61876034). Yafang Wang's research was supported by the National Natural Science Foundation of China (No. 61503217).

## References

1. Angeli, G., Premkumar, M.J.J., Manning, C.D.: Leveraging linguistic structure for open domain information extraction. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pp. 344–354 (2015)
2. Banko, M., Cafarella, M.J., Soderland, S., Broadhead, M., Etzioni, O.: Open information extraction from the web. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence, vol. 7, pp. 2670–2676 (2007)
3. Benikova, D., Fahrer, U., Gabriel, A., Kaufmann, M., Yimam, S.M., von Landesberger, T., Biemann, C.: Network of the day: Aggregating and visualizing entity networks from online sources
4. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, pp. 1247–1250. ACM (2008)
5. Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka, E.R., Mitchell, T.M.: Toward an architecture for never-ending language learning. In: Proceedings of the 24th AAAI Conference on Artificial Intelligence (2010)
6. Council, I.: EventsML-G2: A data model and format for collecting and distributing event information (2014). http://www.iptc.org/site/News_Exchange_Formats/EventsML-G2
7. Council, I.P.T.: rnews (2014). http://dev.iptc.org/rNews
8. Council, I.P.T.: NewsML-G2 2.28 specification (2019). https://iptc.org/std/NewsML-G2/2.28/specification/NewsML-G2-2.28-specification.html
9. Del Corro, L., Gemulla, R.: ClausIE: clause-based open information extraction. In: Proceedings of the 22nd International Conference on World Wide Web, pp. 355–366. ACM (2013)

10. Fader, A., Soderland, S., Etzioni, O.: Identifying relations for open information extraction. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 1535–1545. Association for Computational Linguistics (2011)
11. Falke, T., Gurevych, I.: GraphDocExplore: A framework for the experimental comparison of graph-based document exploration techniques. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 19–24 (2017)
12. Fuchs, C.A., Peres, A.: Quantum-state disturbance versus information gain: Uncertainty relations for quantum information. Phys. Rev. A **53**(4), 2038 (1996)
13. Galárraga, L., Heitz, G., Murphy, K., Suchanek, F.M.: Canonicalizing open knowledge bases. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM '14, pp. 1679-1688. ACM, New York, NY, USA (2014). 10.1145/2661829.2662073
14. Gashteovski, K., Gemulla, R., Del Corro, L.: MinIE: minimizing facts in open information extraction. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 2630–2640 (2017)
15. Ge, T., Wang, Y., de Melo, G., Li, H., Chen, B.: Visualizing and curating knowledge graphs over time and space. pp. 25–30 (2016). https://www.aclweb.org/anthology/P16-4005.pdf
16. Google Microsoft, Y.: Schemas – schema.org. (2012). http://www.schema.org/docs/schemas.html
17. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. In: Proceedings of the 14th Annual Meeting of the Association for Computational Linguistics, pp. 539–545. Association for Computational Linguistics (1992)
18. Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., de Melo, G., Gutierrez, C., Labra Gayo, J.E., Kirrane, S., Neumaier, S., Polleres, A., Navigli, R., Ngonga Ngomo, A.C., Rashid, S.M., Rula, A., Schmelzeisen, L., Sequeda, J., Staab, S., Zimmermann, A.: Knowledge graphs. arXiv:2003.02320 (2020)
19. Hou, L., Li, J., Wang, Z., Tang, J., Zhang, P., Yang, R., Zheng, Q.: Newsminer: Multifaceted news analysis for event search. Knowl.-Based Syst. **76**, 17–29 (2015)
20. Hu, G., Qin, Y., Shao, J.: Personalized travel route recommendation from multi-source social media data Multimedia Tools and Applications (2018)
21. Ji, H., Favre, B., Lin, W.P., Gillick, D., Hakkani-Tur, D., Grishman, R.: Open-Domain Multi-Document Summarization via Information Extraction: Challenges and Prospects Multi-Source, Multilingual Information Extraction and Summarization, Pp. 177–201. Springer (2013)
22. Kochtchi, A., Landesberger, T.v., Biemann, C.: Networks of Names: Visual Exploration and Semi-Automatic Tagging of Social Networks from Newspaper Articles. In: Computer Graphics Forum, Vol. 33, pp. 211–220. Wiley Online Library (2014)
23. Leskovec, J., Backstrom, L., Kleinberg, J.: Meme-tracking and the dynamics of the news cycle. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 497–506. ACM (2009)
24. Li, J., Li, J., Tang, J.: A flexible topic-driven framework for news exploration. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, vol. 2007 (2007)
25. Lin, C.X., Zhao, B., Mei, Q., Han, J.: PET: A statistical model for popular events tracking in social communities. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 929–938. ACM (2010)
26. Mann, G.: Multi-document relationship fusion via constraints on probabilistic databases. In: Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference, pp. 332–339 (2007)
27. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J., McClosky, D.: The Stanford CoreNLP Natural Language Processing Toolkit. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pp. 55–60 (2014)
28. Mausam, M.: Open information extraction systems and downstream applications. In: Proceedings of the 25th International Joint Conference on Artificial Intelligence, pp. 4074–4077. AAAI Press (2016)
29. Mei, Q., Zhai, C.: Discovering evolutionary theme patterns from text: an exploration of temporal text mining. In: Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 198–207. ACM (2005)
30. Mihalcea, R., Tarau, P.: TextRank: Bringing order into text. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (2004)
31. Miller, G.A.: WordNet: a lexical database for english. Commun. ACM **38**(11), 39–41 (1995)

32. Mitchell, T., Cohen, W., Hruschka, E., Talukdar, P., Yang, B., Betteridge, J., Carlson, A., Dalvi, B., Gardner, M., Kisiel, B., et al.: Never-ending learning. Communications of the ACM **61**(5), 103–115 (2018)
33. Pilehvar, M.T., Jurgens, D., Navigli, R.: Align, disambiguate and walk: a unified approach for measuring semantic similarity. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, vol. 1, pp. 1341–1351 (2013)
34. Pouliquen, B., Steinberger, R., Deguernel, O.: Story tracking: linking similar news over time and across languages. In: Proceedings of the Workshop on Multi-source Multilingual Information Extraction and Summarization, pp. 49–56. Association for Computational Linguistics (2008)
35. Rouces, J., de Melo, G., Hose, K.: Heuristics for connecting heterogeneous knowledge via FrameBase. In: Proceedings of ESWC 2016, Lecture Notes in Computer Science, pp. 20–35. Springer (2016). https://link.springer.com/chapter/10.1007/978-3-319-34129-3_2
36. Schmitz, M., Bart, R., Soderland, S., Etzioni, O., et al.: Open language learning for information extraction. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 523–534. ACL (2012)
37. Shahaf, D., Guestrin, C.: Connecting the dots between news articles. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 623–632. ACM (2010)
38. Shan, D., Zhao, W.X., Chen, R., Shu, B., Wang, Z., Yao, J., Yan, H., Li, X.: EventSearch: a system for event discovery and retrieval on multi-type historical data. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1564–1567. ACM (2012)
39. Sheng, Y., Xu, Z., Wang, Y., Zhang, X., Jia, J., You, Z., de Melo, G.: Visualizing multi-document semantics via open domain information extraction. In: Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, pp. 695–699. Springer (2018)
40. Spitkovsky, V.I., Chang, A.X.: A cross-lingual dictionary for English Wikipedia concepts. In: Proceedings of the 8th International Conference on Language Resources and Evaluation, pp. 3168–3175 (2012)
41. Sridhar, V.K.R.: Unsupervised topic modeling for short texts using distributed representations of words. In: Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, pp. 192–200 (2015)
42. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: Proceedings of the 16th International Conference on World Wide Web, pp. 697–706. ACM (2007)
43. Tandon, N., de Melo, G.: Information extraction from web-scale n-gram data. In: Zhai, C., Yarowsky, D., Viegas, E., Wang, K., Vogel, S. (eds.) Web N-gram Workshop. Workshop of the 33rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, vol. 5803, pp. 8–15. ACM (2010). http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.365.2318
44. Tandon, N., de Melo, G., De, A., Weikum, G.: Knowlywood: Mining activity knowledge from Hollywood narratives. In: Proceedings of CIKM 2015, pp. 223–232. ACM. (2015). https://dl.acm.org/doi/10.1145/2806416.2806583
45. Tandon, N., de Melo, G., Suchanek, F.M., Weikum, G.: WebChild: Harvesting and organizing commonsense knowledge from the web. In: Carterettem, B., Diaz, F., Castillo, C., Metzler, D. (eds.) Proceedings of ACM WSDM 2014, pp. 523–532. ACM (2014)
46. Tandon, N., de Melo, G., Weikum, G.: Acquiring comparative commonsense knowledge from the web. In: Proceedings of AAAI 2014, pp. 166–172. AAAI. (2014). https://dl.acm.org/doi/10.5555/2893873.2893902
47. Tixier, A., Skianis, K., Vazirgiannis, M.: GoWvis: a web application for graph-of-words-based text visualization and summarization (2016)
48. Wang, L., Guo, Z., Wang, Y., Cui, Z., Liu, S., de Melo, G.: Social media vs. news media: Analyzing real-world events from different perspectives. In: Proceedings of DEXA 2018, LNCS, vol. 11030, pp. 471–479. Springer Verlag (2018). https://doi.org/10.1007/978-3-319-98812-243. https://link.springer.com/chapter/10.1007/978-3-319-98812-243
49. Xu, T., Liu, D., Chen, E., Cao, H., Tian, J.: Towards Annotating Media Contents through Social Diffusion Analysis. In: 2012 IEEE 12Th International Conference on Data Mining, pp. 1158–1163. IEEE (2012)
50. Xu, T., Zhu, H., Chen, E., Huai, B., Xiong, H., Tian, J.: Learning to annotate via social interaction analytics. Knowledge and information systems **41**(2), 251–276 (2014)
51. Yang, Q., Cheng, Y., Wang, S., de Melo, G.: HiText: Text reading with dynamic salience marking. In: Proceedings of WWW 2017, pp. 311–319. ACM (2017). https://dl.acm.org/citation.cfm?id=3041021.3054168

52. Yimam, S.M., Ulrich, H., von Landesberger, T., Rosenbach, M., Regneri, M., Panchenko, A., Lehmann, F., Fahrer, U., Biemann, C., Ballweg, K.: new/s/leak–information extraction and visualization for investigative data journalists. In: Proceedings of ACL 2016 (System Demonstrations), pp. 163–168. Association for Computational Linguistics (2016). https://doi.org/10.18653/v1/P16-4028, https://www.aclweb.org/anthology/P16-4028/
53. Yu, D., Huang, L., Ji, H.: Open relation extraction and grounding. In: Proceedings of the 8th International Joint Conference on Natural Language Processing, pp. 854–864 (2017)
54. Zhu, C., Zhu, H., Ge, Y., Chen, E., Liu, Q., Xu, T., Xiong, H.: Tracking the evolution of social emotions with topic models. Knowl. Inf. Syst. **47**(3), 517–544 (2016)