Coherence and Salience-Based Multi-Document Relationship Mining

Yongpan Sheng and Zenglin Xu(✉)

University of Electronic Science and Technology of China, Chengdu, China
shengyp2011@gmail.com, zlxu@uestc.edu.cn

Abstract. In today's interconnected world, there is an endless 24/7 stream of new articles appearing online. Faced with these overwhelming amounts of data, it is often helpful to consider only the key entities and concepts and their relationships. This is challenging, as relevant connections may be spread across a number of disparate articles and sources. In this paper, we propose a unified framework to aid users in quickly discerning salient connections and facts from a set of related documents, and presents the resulting information in a graph-based visualization. Specifically, given a set of relevant documents as input, we firstly extract candidate facts from above sources by exploiting Open Information Extraction (Open IE) approaches. Then, we design a Two-Stage Candidate Triple Filtering (TCTF) approach based on a self-training framework to maintain only coherent facts associated with the specified document topic from the candidates and connect them in the form of an initial graph. We further construct this graph by a heuristic to ensure the final conceptual graph only consist of facts likely to represent meaningful and salient relationships, which users may explore graphically. The experiments on two real-world datasets illustrate that our extraction approach achieves 2.4% higher on the average of F-score over several OpenIE baselines. We also further present an empirical evaluation of the quality of the final generated conceptual graph towards different topics on its coverage rate of topic entities and concepts, confidence score, and the compatibility of involved facts. Experimental results show the effectiveness of our proposed approach.

Keywords: Multi-Document relationship mining · 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. Faced with these overwhelming amounts of information, it is helpful to consider only the key entities and concepts and their relationships. Often, these are spread across a number of disparate articles and sources. Not only do different outlets often cover different aspects of a story.

© Springer Nature Switzerland AG 2019
J. Shao et al. (Eds.): APWeb-WAIM 2019, LNCS 11641, pp. 414–430, 2019.
https://doi.org/10.1007/978-3-030-26072-9_30

Typically, 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. While there is ample work on news topic detection and tracking, previous work has not explored how to connect and present important facts and connections across articles.

In this paper, we propose a unified framework 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 approaches, such as Open Information Extraction (Open IE) methods [3] readily extract large amounts of subject-predicate-object triples from the unstructured texts. However, it does not make any attempt to connect the extracted facts across sentences or even documents. Additionally, OpenIE methods tend to yield countless non-informative and redundant extractions that are not indicative of what is genuinely being expressed in a given text. E.g, we may obtain triples such as (*"they"*, *"spoke with"*, *"multiple sources"*). From this, we proposed a Two-Stage Candidate Triple Filtering (TCTF) approach to discern which of the mined candidate facts are coherent with the specified document topic, and connected them to form an initial graph. We further construct this graph by a heuristic strategy that iteratively remove the weakest concepts with relatively lower importance scores are computed by the extended TextRank algorithm, so that it ensures the final large conceptual graph only consists of facts are likely to represent meaningful and salient relationships, which users may explore graphically.

2 Related Work

GoWvis[1] 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 [4] 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 Network of the Day project[2] builds on Networks of Names to provide a daily analysis of German news articles. The *news/s/leak* project[3] further extends this line of work by adding access to further corpora and helps journalists to analyse and discover newsworthy stories from large textual datasets. This version also attaches general document keywords as tags to relationships, but does not aim at sentence-level relation semantics as our system. Sheng et al. [15] introduce a system that can extract facts from a set of related articles. However, for the quality of extracted facts, not sufficient evaluation is performed.

OpenIE systems [3] do extract specific facts from text, but they do not aim at connecting facts across sentences or documents, and often neglect whether

[1] https://safetyapp.shinyapps.io/GoWvis/.
[2] http://tagesnetzwerk.de.
[3] http://www.newsleak.io/.

the extractions are meaningful on their own and indicative of what is being expressed in the input text. [12] used information extraction-based features to improve multi-document summarization. [11] investigated logical constraints to aggregate a closed set of relations mined from multiple documents. Our approach, in contrast, addresses the task of extracting and connecting salient entities and facts from multiple documents, enabling a deeper exploration of meaningful connections.

3 Problem Statement

3.1 Problem Formulation

Our task is aimed to assist users in quickly finding salient connections and facts from a collection of relevant articles, and in summary, it can be best described as a combination of three major subtasks:

- **Subtask 1: Candidate Fact Extraction.** Given a collection of documents $D = \{d_1, d_2, ..., d_M\}$ clustered around a topic T. The goal of this subtask is to extract a set of facts $F_c = \{f_1, f_2, ..., f_N\}$ from D. Each of facts is essentially (s, r, o) triple, for *subject* s, *relation* r, and *object* o. Since we need to estimate the coherence of these preferred facts for T, we refer to them as *candidate facts*.
- **Subtask 2: Topic Coherence Estimation of Candidate Facts.** Given a specified document topic T, the goal of the subtask is to find a subset of $F_c^{'} \subseteq F_c$, and each of them should be coherent with T.
- **Subtask 3: Conceptual Graph Construction.** The goal of the subtask is to determine which of the facts from $F_c^{'} \subseteq F_c$ generated by 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 The Framework of Our Approach

The framework of our approach can be shown in Fig. 1. It also comprise three major phases in order to address the problems associated with the subtasks as we have discussed earlier. In the candidate fact extraction phrase, given a specified document topic, we first preprocess relevant input texts in natural language, which can be more efficient for later procedures. To achieve this, we rely on a series of natural language processing methods, including document ranking, coreference resolution and sentence ranking. Then, three existing popular extractors including OLLIE [7], ClausIE [8], MinIE [9] are utilized to extract candidate facts from processed text documents, which can be expressed as subject-predicate-object triples[4].

[4] During the extraction, we applied a few straightforward transformations to correct two types of common errors such as wrong boundaries and uninformative extraction, which were caused by the syntactic analysis in extraction approaches.

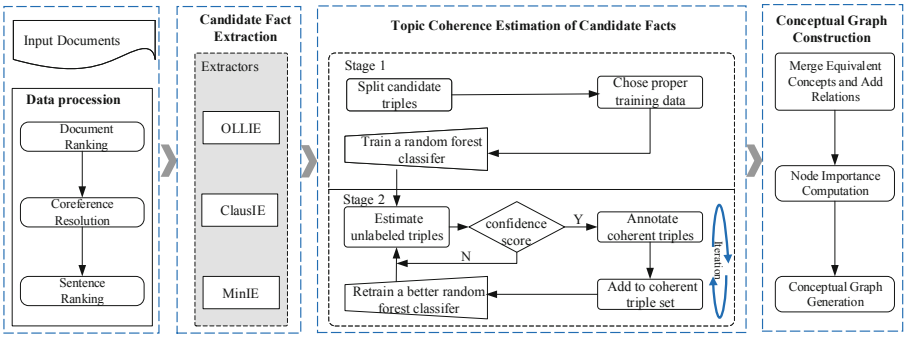


Fig. 1. The framework of our approach

The above phase can generate a set of candidate facts from the input documents. However, many of extracted relational triples are correct ones but irrelevant to the specified document topic. Therefore, we propose a Two-Stage Candidate Triple Filtering (TCTF) approach to further filter out those irrelevant triples by how coherence they are for the given document topic of interest. In the first stage, we firstly split the candidates and annotate specific smaller set of triples from them as available training data, and then used to train an random forest classifier. In the second stage, based on the trained classifier, it estimate each of unlabeled facts from the candidates by its confidence score. The triples that are annotated as coherent ones will add to corresponding coherent triple set for retaining a better random forest classifier used to annotate the unlabeled triples. The whole procedure works in an iterative manner based on a self-training framework until the convergence of the unlabeled triple set or achieving the max iteration number. Finally, we aggregate those coherent facts into an initial graph by further merging their equivalent entities and concepts, and adding synthetic relations. We further construct this graph by a heuristic strategy that iteratively removes the weakest entities and concepts with relatively lower importance scores computed by the extended TextRank algorithm, so that it ensures the final conceptual graph only consist of facts likely to represent meaningful and salient relationships where users may explore graphically.

4 The Proposed Approach

4.1 Candidate Fact Extraction

Document Ranking. As the first step, given a specified document topic T, we selected the words appearing in its relevant document collection $D = \{d_1, d_2, ..., d_M\}$ with sufficiently high frequency as topic words, and computes standard TF-IDF weights[5] for each word. The topic words are used to induce

[5] https://en.wikipedia.org/wiki/Tf%E2%80%93idf.

document representations. Next, we rank D by the TF-IDF weights (see footnote 5) of the topic words in each document, and by default, to obtain a final ranked top-k document list $D^{'} = \{d_1, d_2, ..., d_k\}$.

Coreference Resolution. Pronouns and other form of coreference are resolved in each document $d_i \in D^{'}$ using Stanford CoreNLP system [1] (i.e., *"she"* may be replaced by *"Angela Merkel"*).

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 approach computes the TextRank importance scores[6] for all sentences within $d_i \in D^{'}$. It then considers only those sentences with sufficiently high scores, denoted by $S = \{s_1, s_2, ..., s_K\}$.

Candidate Relational Triples Extraction. As a crucial step of the high-quality fact corpus for conceptual graph construction, we therefore have to committed to the high-accuracy of information extractors. For this, we adopt an Open IE approach to extract relational triples of the form (*noun phrase*, *relation phrase*, *noun phrase*) for raw texts, rather than just focusing on named entities (e.g., *"Billionaire Donald Trump"*), as some previous approaches do, our approach supports an unbounded range of noun phrase concepts (e.g., *"the snow storm on the East Coast"*) and relationships with explicit relation labels (e.g., *"became mayor of"*). The latter are extracted from verb phrases as well as from other contributions. More specifically, We leveraged three existing excellent and popular extractors namely OLLIE [7], ClausIE [8], MinIE [9] to process all selected sentences from S after the sentence ranking, only a relational triple is obtained simultaneously by the extractors is incorporated into the collection of candidate facts[7].

Due to all of the extractors we chose are based on the dependency parsing trees, which may lead to two categories of errors: (1) Wrong boundaries, especially when triples with conjunctions in the sentence were not properly segmented, which the Open IE approach sometimes fails to do; (2) Uninformative extraction, which might occur in the *relation phrases*. i.e., the *relation phrases* are expressed by a combination product of a verb with a noun, e.g., light verb constructions. Moreover, the distance between the adjacent words in the *relation phrase* may, in fact, be distant in the original sentence. The adjacency order of the words in a triple may be different from that in the sentence, e.g., we may obtain a triple of (*"The police"*, *"are driving away"*, *"paraders"*) from such a sentence *"The police are driving paraders away from GOP front-runner Trump"*. Therefore, we applied a few straightforward transformations to correct above errors: (1) We broke down triples with conjunctions in either of the *noun phrase* into separate triple per conjunction; (2) We defined some word order constraints, i.e., the *noun phrases* in a triple (*noun phrase1*, *relation phrase*, *noun phrase2*) are ordered. All words in *noun phrase1* must appear before all words in *relation*

[6] https://github.com/letiantian/TextRank4ZH/blob/master/README.md.

[7] As for a triple extracted by OLLIE, ClausIE, and MinIE, only when its confidence is greater than 0.85 can it be judged that the triple is a correct extraction.

phrase. All the words contained in *noun phrase2* must appear after the *predicate phrase*. The order of words appearing in *predicate phrase* must be consistent with them appearing in the original sentence. In addition, each *relation phrase* must have appeared in the original sentence, is not the modified word or the added word.

4.2 Topic Coherence Estimation of Candidate Facts

Once above extracting process is completed, the framework will yield numerous candidate triples $F_c = \{f_1, f_2, ..., f_N\}$. However, not all the triple $f_i \in F$ is indicative of what is genuinely being expressed in a given text for the specified topic. E.g, the triple *("Trump", "will visit", "the Apple Inc")*, this is a correct one but meaningless for the news topic like *US president election*. Therefore, the task of this phrase seeks to filter out those irrelevance triples from the candidate triples for different document topic. In this work, we propose a TCTF approach based on Self-training for explaining when to annotate a triple as coherent with the specified topic. A pseudo-code of TCTFA is provided in Algorithm 1.

In the first stage, we randomly select a small fraction of triples $F_s \subseteq F_c$ under the document topic, and divide them into training set F_{str}, validation set F_{sv}, and test set F_{st} with a 8:1:1 ratio. We then train a random forest classifier M over F_s and obtain a F1-score θ (Line 3). Specifically, the triples used in the training are which both subject and object occurring as the topic words list are labeled as positive examples $F_c^{'}$ (i.e., initial coherent triples set), while neither nor them are labeled as negative examples. The identification of topic words rely on their sufficiently high frequency that can be computed using TF-IDF (see footnote 5), which we described earlier for the document ranking part in Sect. 4.1. To measure the topic coherence for each $f_i \in F_s$ from different aspects, we defined several features which are divided into three groups namely *topic features*, *text features* as well as *source features*, as described in Table 1. Besides, we further illustrate the computations of a series of critical features within above groups, including the following:

- *In_Titles*, the frequency of candidates in document titles is divided into three cases: greater than 13.40, less than 3.16, and between 3.16 and 13.40. Three binary features are used to represent these cases.
- *Redundance*, this is defined as the ratio of the size of candidates that have same *subject*, *relation* and *object* with the candidate triple f to the total number of candidates, these extracted redundant facts are 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.,

$$Sim(f, l) = \frac{count_{sim}(f, l)}{count(l)} \tag{1}$$

Algorithm 1. Two-Stage Candidate Triple Filtering Approach based on Self-training

Input: F_c: Candidate triple set;
F_s: A small fraction of training triple set;
$F_c^{'}$: Coherent triple set;
F_{sv}: Validation set;
kF: flag for the first update for the model;
Output: F_l^*: Combined coherent triple set;
Initialization: $F_l^* \leftarrow F_c^{'}, \theta \leftarrow 0, \theta_{new} \leftarrow 0, kF \leftarrow true$;
The first stage → Train a Random Forest Classifier
Learn a random forest classifier M from F_s and obtain F1-score θ;
The second stage → Extend Coherent Triple set
repeat
  if kF is $false$ **then**
    $\theta \leftarrow \theta_{new}$;
    $M \leftarrow M*$;
  end if
  for each unlabeled triple $f \in F_c$ **do**
    Use M to label f and obtain $f^{'}$;
    if $confidence(f^{'}) > \alpha$ **then**
      $F_l^* \leftarrow F_l^* + f^{'}$;
      $F_c \leftarrow F_c - f$;
    end if
  end for
  Retrain model $M*$ on F_l^*;
  Test model $M*$ on F_{sv} and obtain a new F1-score θ_{new};
  $kF \leftarrow false$;
until $\theta_{new} - \theta < \epsilon$
return F_l^*

where $count_{sim}(f, l)$ denotes the number of candidates that are similar to f in the whole candidates[8], $count(l)$ is the total number of candidates.

– *Relation_Context*, the ratio of the size of context of the relation r in the candidate triple f over the total number of candidates, i.e.,

$$RelCxt(r_f, l) = \frac{count_{context(r_f)}}{count(l)} \tag{2}$$

where $count_{context(r_f)}$ denotes the size of *relation context* of r in f, which is consist of the candidates that have same *relation type* with f, and $count(l)$ is the total number of candidates.

[8] The similarity scores between two candidate fact f_i, f_j is computed as $sim(f_i, f_j) = \gamma \dot{s}_k + (1 - \gamma)\dot{l}_k$, where s_k, l_k denote the semantic similarity and literal similarity scores between the facts, respectively. We compute s_k using the *Align, Disambiguate and Walk* algorithm [2], while l_k are computed using the Jaccard index. $\gamma = 0.8$ denotes the relative degree to which the semantic similarity contributes to the overall similarity score, as opposed to the literal similarity.

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 occurring as the topic words list	0 or 1
2	Is_Subject_tw	Whether subject in a candidate fact occurring as the topic words list	0 or 0.5
3	Is_Object_tw	Whether object in a candidate fact occurring as the topic words list	0 or 0.5
4	In_Titles	Some binary features based on the frequency of occurrence of a candidate triple in document title in the relevant documents	0 or 1
5	Redundance	The ratio of redundant candidates with the candidate fact	[0, 1]
6	Similarity	The ratio of candidates are similar to the candidate fact	[0, 1]
7	Relation_Context	The ratio of candidates involving the same type of relation with the candidate fact	[0, 1]
8	Compatibility	The compatibility between the relation context of the candidate triple and the semantic information itself	[0, 1]
Text features			
9	Is_In_Title	Whether a candidate triple appears in the document title	0 or 1
10	Is_In_Abstract	Whether a candidate triple appears in an automatic summarization of the document	0 or 1
11	Is_In_MaxSent	Whether a candidate triple appears in the sentence with maximum TextRank importance score in the document	0 or 1
12	Sum_tfidf	Sum of TF-IDF of subject and object in the candidate triple in the relevant documents	[0, 1]
13	Avg_tfidf	Average of TF-IDF of subject and object in the candidate triple in the relevant documents	[0, 1]
Source features			
14	Source_Num	The number of sources where the candidate triple is extracted	1 or 2
15	Sentence_Num	The number of sentences where the candidate triple is extracted	1, 2, ... 50
16	Relevant_Docs	The ratio of documents which contain the candidate triple	[0, 1]

– *Compatibility*, the compatibility between the *relation context* of the relation r in the candidate triple f and the semantic information of f itself, i.e.,

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

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

In the second stage, based on the trained classifier M, it calculates the confidence score for every unlabeled triple $f \in F_c$ using classifier's confidence and

regard the triples that are assigned with the score above a fixed threshold α as the coherent triples. They will make up the next iteration's labeled triple set F_l^* for retraining a better model $M*$ (Line 10–17). Correspondingly, the parameters of $M*$ are tuned according to the precision metric on F_{sv}, and obtain a new F1-score θ_{new} (Line 18). We perform $\theta_{new} - \theta$ and compare it to a threshold, ϵ. If the result to be smaller than ϵ, it will enter the next iteration of the learning process based on a self-training framework. After several numbers of iterations, the final extended coherent triple set F_l^* will be generated and used to form a conceptual graph in the next phrase of our approach.

4.3 Conceptual Graph Construction

Merge Equivalent Concepts and Add Relations. In order to establish a single connected graph that is more consistent, we further merge potential entities and concepts in F_l^* stemming from former process which works in two steps: (1) We made use of Stanford CoreNLP entity linker which is based on [5] for identifying entity or concept mentions and link them to Ontological KBs such as Wikipedia, Freebase entity linking. Roughly, in about 30% cases, we get this information for the entities. If two entities and concepts are linked to the same Wikipedia entity, we assume them to be equivalent as per this information. e.g., *US* and *America* can get linked to the same Wikipedia entity *United_States*; (2) It was found that approximately 53% pairs of entities or concepts, whose labels present slightly similarity in terms of their literal meaning, to suggest possible connection. When we can't obtain sufficient context information, it's difficult to decide to whether should merge to form a label that is appropriate for both of them, e.g., *all the Democratic candidates* and *US Democratic Parties*. The identification of them requires more human-crafted knowledge. For this, we decided to use three expert annotators with NLP background for such subtask: they could connect these entities and concepts according to their background knowledge step by step, and observed an agreement of 84% ($\kappa = 0.66$[9]). To support the annotators, once again the *Align, Disambiguate and Walk tool* [2] is used for semantically similarity computation between concepts for coreference.

After that, on average, there remains not more than 5 subgraphs that can further be connected for different document topics. Hence, for each topic, annotators were allowed to add up to three synthetic relations with freely defined labels to connect these subgraphs into a fully connected graph $G^{'}$, observing 87% ($\kappa = 0.71$) agreement.

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 [14], 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 $G^{'} = (\mathcal{V}, \xi)$ denotes a weighted directed graph generated by former step, where $\mathcal{V} = \{v_1, v_2, ..., v_R\}$

[9] Kappa implementation: https://gist.github.com/ShinNoNoir/9687179.

represent a set of preferred nodes which correspond to entities and concepts in $G^{'}$, and ξ is a directed edge set, associated with each directed edge $v_i \rightarrow v_j$ representing a dependency relation originating from v_i to v_j. We assign a weight $w_{ij} = 1$ to $v_i \rightarrow v_j$ and its reverse edge $v_j \rightarrow v_i$ with $w_{ji} = 0.5$. By adding lower weighted reverse edges, we can analyze the relation between two nodes which are not connected by directed dependency links while maintaining our preferences toward the original directions.

TextRank [10] is a ranking algorithm can be used to compute the importance of each node within $G^{'}$ based on graph random walks. Similarity, 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 higher preferences toward these nodes when computing the importance scores since 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 $1/|\mathcal{V}|$ for $v_k \in \mathcal{V}$, otherwise 0. Let I be the $1 \times |\mathcal{V}|$ importance vector to be computed over all nodes in $G^{'}$ as follows.

$$I_i = (1 - d) \sum_{j \in \mathcal{N}(i)} \frac{w_{ji}}{\Sigma_{k \in \mathcal{N}(j)} w_{jk}} I(j) + d \cdot p_i, \tag{4}$$

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

Conceptual Graph Generation. The recommended [3] maximum size of a concept graph is 25 concepts, which we use as a constraint. We rely on a heuristic to find a full graph that is connected and satisfies the size limit of 25 concepts: We iteratively remove the weakest concepts with relatively lower importance score is computed using Eq. 4 until only one connected component of 25 entities and concepts or less remains, which is used as the final conceptual graph G. This approach guarantees that the graph is connected with salient concepts, but might not find the subset of concepts that has the highest total importance score.

5 Experiment

5.1 Dataset

Our dataset include 5 categories, and for each category we have 2 popular events and each of which represents a document 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. It is 3 times larger than typical DUC[10]

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

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 articles in our dataset stem from a larger news document collection[11] released by Signal Media as well as crawled from Web Blogs by ourselves, 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.

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, Blog
	2	North Korea nuclear test	2017-08-09–2017-11-20	30	1713 ± 122	News
Business and economy	3	Chinese cooperation with Sudan	2015-09-01–2015-09-30	30	768 ± 132	News, Blog
	4	Trump TPP	2016-12-23–2017-02-23	30	879 ± 306	News
Politics and elections	5	US presidential election	2016-06-14–2016-08-14	30	1175 ± 207	News, Blog
	6	US-China trade war	2018-03-23–2018-06-15	30	2412 ± 542	News, Blog
Arts and culture	7	Muslim culture	2013-02-01–2013-05-01	30	972 ± 161	News, Blog
	8	Turing Award winner	2019-03-15–2019-04-01	30	1563 ± 464	News, Blog
Information technology and application software	9	Next-generation search engine	2016-11-07–2017-01-03	30	729 ± 280	News, Blog
	10	Program repair for Android system	2018-02-01–2018-05-10	30	772 ± 453	Blog

5.2 Experimental Setting

We conduct two types of experiments on above datasets for validating the effectiveness of our proposed approach: (1) The first experiment focuses on sentence-level extractions. Therefore, we first randomly sample 10 documents from every document topics (100 documents in total) and perform coreference resolution. Then, once again a random sample of 10 sentences from every extracted document (1,000 sentences in total) for further analysis. Each sentence is examined by three expert annotators with NLP background independently to annotate all of correct triples[12]; (2) We further conduct an empirical study to investigate 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.

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

[12] A triple is annotated as correct if the following conditions are met: (i) it is entailed by its corresponding clause; (ii) it is reasonable or meaningful without any context and (iii) when these three annotators mark it correct simultaneously (The inter-annotator agreement was 82% ($\kappa = 0.60$)).

OpenIE Baseline Methods. We compare our extraction method with the baselines of using just the OpenIE systems: OLLIE [7], ClausIE [8], MinIE [9], and OpenIE-4.x [6]. Apart from this, we also evaluate two variants of the approach to study the effect of coreference resolution and several transformations used to correcting errors caused by OpenIE methods.

Evaluation Metrics. The metrics used in our experimental analysis are Precision (P), Recall (R), F-score (F1), coverage rate of topic entities and concepts, confidence score, compatibility of involved facts in the graph, defined as follows:

- Three standard metrics are Precision (P), Recall (R), F-score (F1), respectively
$$P = \frac{\#\ correct}{\#\ extractions}, R = \frac{\#\ correct}{\#\ relations}, F1 = \frac{2PR}{P+R} \tag{5}$$
where "# correct" denotes the number of extractions deemed as correct, "# extractions" denotes the total number of extractions, and "# relations" denotes the number of triples are annotated as correct extractions (see footnote 12).
- The coverage rate of topic entities and concepts, i.e.,
$$TopicCon\_Coverage = \frac{\#\ topic\_concepts}{\#\ concepts} \tag{6}$$
where "# topic_concepts" denotes the number of entities and concepts for which annotated as topic concepts[13], and "# concepts" denotes the total number of all entities and concepts in the conceptual graph.
- Confidence score, i.e.,
$$Avg\_Confidence(f_i, n) = \frac{\sum_{i=1}^{n} conf(f_i)}{n}, \tag{7}$$
where $conf(f_i)$ denotes the confidence score[14] of each fact f_i, n is the number of facts which involved in the final conceptual graph.
- Compatibility of involved facts in the graph, i.e.,
$$Avg\_Compatibility(f_i, f_j, n) = \frac{\sum_{i=1}^{n} \sum_{j>i} cmp(f_i, f_j)}{c_n^2}, \tag{8}$$
where f_i and f_j are any facts are in the final conceptual graph, which contains n facts. $cmp(f_i, f_j)$ denotes the compatibility between f_i and f_j, similar to Eq. 3, it can be calculated using $cmp(f_i, f_j) = (RelCxt(r_{f_i}, n) + RelCxt(r_{f_j}, n)) \cdot (1 - \epsilon + \epsilon \cdot sim(f_i, f_j))$, where $sim(f_i, f_j)$ denotes the similarity scores (see footnote 8) between fact f_i and fact f_j, parameter ϵ is used for smoothing as well as to control the influence of the relation context, and is fixed to 0.5 in our implementation.

[13] An entity or concept is regarded as topic concept when it occurs in the topic words list.

[14] For popular OpenIE systems such as OLLIE, ClausIE, and MinIE, we use the confidence value computed by each system itself as the confidence score of each of facts.

Table 3. Evaluation of precision, recall, and F-score on five independent document topics (including topic 1 to topic 5) from two datasets

OpenIE methods	#Topic 1			#Topic 2			#Topic 3			#Topic 4			#Topic 5		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
OLLIE [7]	0.72	0.34	0.67	0.28	0.37	0.32	0.84	0.35	0.49	0.75	0.43	0.55	0.62	0.29	0.40
ClausIE [8]	0.63	0.62	0.66	0.58	0.55	0.56	0.80	0.53	0.64	0.59	0.55	0.57	0.79	0.52	0.63
MinIE [9]	0.70	0.69	0.72	0.61	0.64	**0.62**	0.86	0.58	0.69	0.70	0.62	0.66	0.81	0.66	**0.73**
OpenIE-4.x [6]	0.74	0.46	0.54	0.35	0.80	0.49	0.79	0.41	0.54	0.76	0.40	0.52	0.69	0.34	0.46
Our approach (without coref)	0.43	0.29	0.56	0.44	0.27	0.33	0.65	0.24	0.35	0.47	0.33	0.39	0.45	0.30	0.36
Our approach (without trans)	0.79	0.70	**0.82**	0.61	0.55	0.58	0.92	0.68	**0.78**	0.82	0.71	**0.76**	0.81	0.67	**0.73**
Our approach	0.86	0.85	**0.85**	0.78	0.74	**0.76**	0.95	0.92	**0.93**	0.95	0.82	**0.88**	0.92	0.78	**0.84**

5.3 Evaluation and Results Analysis

Performance Analysis of Extraction Approaches. We selected and presented the evaluation results of our method and OpenIE baselines on ten document topics in Tables 3 and 4[15]. We can draw from that our method has more superiorities compared with the baseline methods across all metrics as illustrated in Equation. In particular, our approach enhanced F-score with an average improvement of 2.4% compared with the baseline methods. The reason is mainly that: (1) Our method takes advantage of the results (see footnote 7) of three extractors including OLLIE [7], ClausIE [8], MinIE [9], which may achieve better performance compared with using single extractor; (2) Using a few straightforward transformations, we observed our approach is better able to identify the boundary of triples for long sentence with conjunction structure, but generally other methods including OLLIE [7] and Open IE-4.x [6] cannot. Moreover, we observed the number of uninformative extractions especially appearing in the relation phrases can significantly reduce by using a relaxed constrain for the word order, i.e., decreasing the frequency of this type of error from 36.8% to 17.1%, this illustrated the effectiveness of the above operation. It also indicated that two types of extraction errors as above in our approach caused by depending on intermediate structures such as dependency parses could also be well figured out. Additionally, it can be seen that our approach (without coref) performs the worst among all the methods, which is due to many cases in which the selected sentences from different topics are not readable without the context when we do not perform coreference resolution, so that it is difficult to correctly identify the extractions relaying on the output of our approach.

Feature Selection and Parameter Tuning. In the process of estimating the topic coherence of candidate facts, although there are fifteen designed features for measuring the topic coherence for each fact from different aspects. To our best knowledge, simply combining all of them will not lead to the best triple classification performance. Therefore, we study the effectiveness of each feature

[15] We mark top-2 performance results in F-score in bold face.

Table 4. Evaluation of precision, recall, and F-score on five independent document topics (including topic 6 to topic 10) from two datasets

OpenIE methods	#Topic 6			#Topic 7			#Topic 8			#Topic 9			#Topic 10		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
OLLIE [7]	0.62	0.27	3.38	0.69	0.44	0.54	0.68	0.35	0.46	0.81	0.24	0.37	0.59	0.21	0.31
ClausIE [8]	0.70	0.53	0.60	0.73	0.55	0.63	0.59	0.43	0.50	0.66	0.64	0.65	0.61	0.49	0.54
MinIE [9]	0.82	0.55	**0.66**	0.77	0.64	**0.70**	0.71	0.62	0.66	0.81	0.60	0.69	0.78	0.55	0.65
Open IE-4.x [6]	0.73	0.51	0.60	0.64	0.30	0.41	0.66	0.58	0.62	0.74	0.59	0.66	0.71	0.65	0.68
Our approach (without coref)	0.43	0.29	0.35	0.44	0.32	0.37	0.47	0.30	0.37	0.55	0.42	0.48	0.40	0.29	0.34
Our approach (without trans)	0.73	0.57	0.64	0.71	0.62	0.66	0.82	0.71	**0.76**	0.81	0.70	**0.75**	0.76	0.71	**0.73**
Our approach	0.90	0.73	**0.81**	0.78	0.69	**0.73**	0.95	0.78	**0.86**	0.88	0.73	**0.80**	0.78	0.74	**0.76**

for the trained random forest classifier. We selected χ^2 and information gain [13] as the classification criteria, and ranked features by χ^2 are shown in Table 5. The results show that only Is_Topic_Word is the top-1 feature ranked by both two measures. Moreover, we further evaluate the contribution of each feature for the classifier when top-k features (sorted by χ^2) are used, all performance results including accuracy, recall and F-score on average are reported in Fig. 2(a). We can observe that the top-7 features dominate the performance of the classifier, i.e., all of measures will converge to an upper limit after the top-7 features are leveraged. The results suggest that the classification accuracy is better when a few effective features are included.

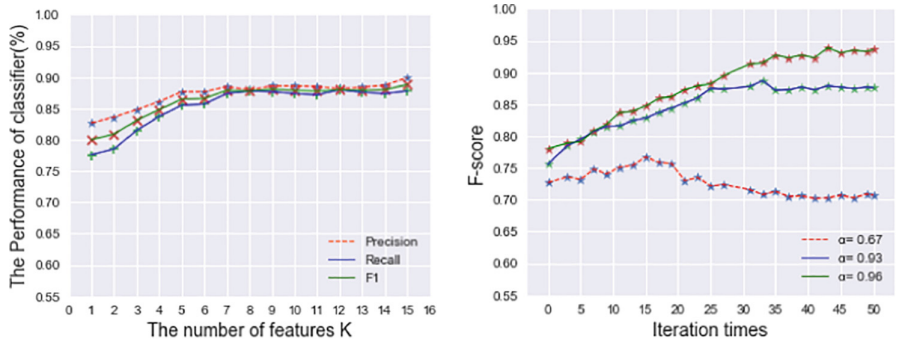
Subsequently, we compared three different thresholds (i.e., $\alpha = 0.67$, $\alpha = 0.93$, $\alpha = 0.96$) used for control noise and select proper triples at the second stage of the proposed TCTF approach. The results are shown in Fig. 2(b). We can observe that, based on the self-training framework, improper thresholds not only could not improve the model (i.e., the random forest classifier), but also render the model fail to annotate improper triples as coherent ones for the specified document topic. The reasons for such phenomena probably are as follows: (1) If the threshold is set to be too small (e.g., $\alpha = 0.67$), the training of the model might be easily affected by the noise data (i.e., added triples), leading to less improvement of triple classification task; (2) If the threshold is set to be too large, on the other hand, the training speed of the model could decrease on the certain of level. E.g., we set α as 0.96, the model converges for a total of 35 epochs approximately. Hence, through considering both training speed and the performance of self-training process in the TCTF approach, we set confidence α as 0.93, which could achieve better performance at the expense of few training epochs.

Quality Analysis of Conceptual Graph. The result of the empirical evaluation of quality of the final generated conceptual graph is shown in Fig. 3. Our approach achieved 100% coverage rate of topic entities and concepts ($TopicCon\_Coverage$), 87% confidence score ($Avg\_Confidence$), and 58% fact compatibility ($Avg\_Compatibility$) over ten document topics. The results

Table 5. Effectiveness of features (sorted by χ^2)

#	Feature	χ^2	IG%	#	Feature	χ^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				

indicates that: (1) The proposed TCTF approach is capable to retain only coherent triples from the candidates towards different document topics[16] whereas the threshold selection which is significant tough; (2) The extracted facts have higher confidence, which demonstrate the importance of node importance computation in conceptual graph construction; (3) Obviously, our approach may not guarantee that the extracted facts have better compatibility, which needs to be further explored.

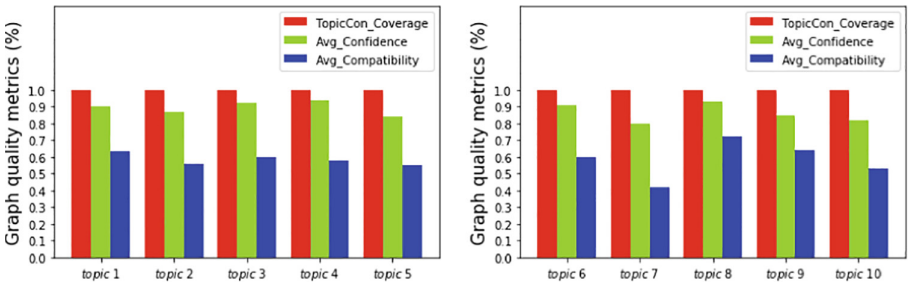


(a) The performance of top-k features for triple classification task

(b) Learning curve of self-training process in TCTF approach at three different confidence thresholds α on the validation set

Fig. 2. The performance evaluation of the model and parameter tuning

[16] The random forest classifier has an average absolute 87% higher on the F-score metric for different topics when the model has converged.



(a) #Topic 1 - #Topic 5 (b) #Topic 6 - #Topic 10

Fig. 3. The quality analysis of the final generated conceptual graphs for ten document topics from three aspects (i.e., *TopicCon_Coverage*, *Avg_Confidence*, and *Avg_Compatibility*).

6 Conclusion

In this paper, we presented a novel framework that aids users in quickly discerning coherence and salience-based connections in a collection of documents, via graph-based visualizations of relationships between concepts even across documents. Experiments on two real-world data sets demonstrate the effectiveness of our proposed approach. In the future, we will give greater exploration to fact fusion problem before fully automated conceptual graph construction for specified domain is possible.

Acknowledgments. This paper was partially supported by National Natural Science Foundation of China (Nos.61572111 and 61876034), and a Fundamental Research Fund for the Central Universities of China (No.ZYGX2016Z003).

References

1. Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. In: ACL, pp. 55–60 (2014)
2. Pilehvar, M.T., Jurgens, D., Navigli, R.: Align, disambiguate and walk: A unified approach for measuring semantic similarity. In: ACL (Volume 1: Long Papers), pp. 1341–1351 (2013)
3. Banko, M., Cafarella, M.J., Soderland, S., Broadhead, M., Etzioni, O.: Open information extraction from the web. In: IJCAI, pp. 2670–2676 (2007)
4. 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, pp. 211–220 (2014)
5. Spitkovsky, V.I., Chang, A.X.: A cross-lingual dictionary for English Wikipedia concepts. In: LREC, pp. 3168–3175 (2012)
6. Mausam, M.: Open information extraction systems and downstream applications. In: IJCAI, pp. 4074–4077 (2016)
7. Schmitz, M., Bart, R., Soderland, S., Etzioni, O.: Open language learning for information extraction. In: EMNLP-CoNLL, pp. 523–534 (2012)

8. Del Corro, L., Gemulla, R.: Clausie: clause-based open information extraction. In: WWW, pp. 355–366 (2013)
9. Gashteovski, K., Gemulla, R., Del Corro, L.: Minie: minimizing facts in open information extraction. In: EMNLP, pp. 2630–2640 (2017)
10. Mihalcea, R., Tarau, P.: TextRank: bringing order into text. In: EMNLP (2004)
11. Mann, G.: Multi-document relationship fusion via constraints on probabilistic databases. In: Human Language Technologies 2007: NAACL, pp. 332–339 (2007)
12. 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. In: Multi-source, multilingual information extraction and summarization, pp. 177–201 (2013)
13. Fuchs, C.A., Peres, A.: Quantum-state disturbance versus information. Uncertainty Relat. Quantum Inf. Phys. Rev. A **53**(4), 20–38 (1996)
14. Yu, D., Huang, L., Ji, H.: Open relation extraction and grounding. In: IJCNLP (Volume 1: Long Papers), pp. 854–864 (2017)
15. 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: ECML-PKDD, pp. 695–699 (2018)