Visualizing Multi-document Semantics via Open Domain Information Extraction

Yongpan Sheng[1] Zenglin Xu[1] Yafang Wang[2] Xiangyu Zhang[1]
Jia Jia[2] Zhonghui You[1] Gerard de Melo[3]

[1]School of Computer Science and Technology
University of Electronic Science and Technology of China (UESTC),
[2]Shandong University, [3]Rutgers University

The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD), 2018
The 11st Ph.D. Academic Forum holding by Fudan University, together with Shanghai Jiao Tong University, 2019

Outline

1. Motivation
2. Problem Definition
3. System Architecture
4. Approach and Implementation
    - Input Data
    - Fact Extraction
    - Fact Filtering
    - Conceptual Graph Construction
5. Experiments
6. Conclusions and Future Work

Outline

1. Motivation

2. Problem Definition

3. System Architecture

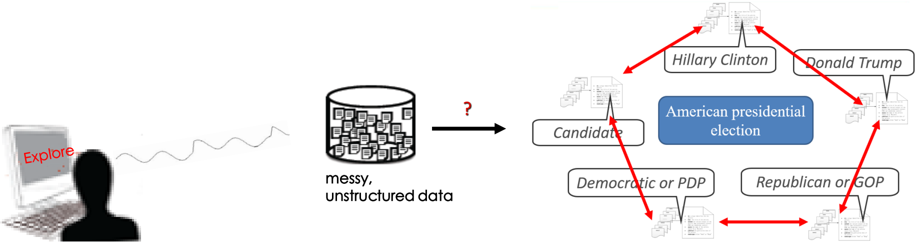
4. Approach and Implementation
   - Input Data
   - Fact Extraction
   - Fact Filtering
   - Conceptual Graph Construction

5. Experiments

6. Conclusions and Future Work

Motivation

With the emergence of massive text corpora in many domains and languages, the sheer size and rapid growth of this new data poses many challenges understanding and connecting significant insights from these massive unstructured texts.



Explore

messy,
unstructured data

?

Hillary Clinton

Donald Trump

Candidate

American presidential election

Democratic or PDP

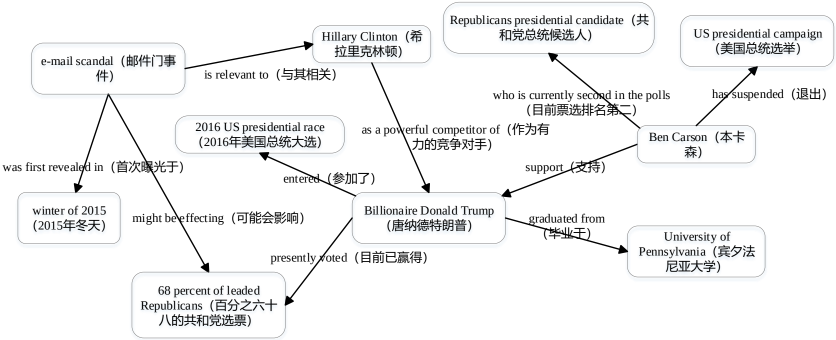
Republican or GOP

Motivation

- How to mine and organize meaningful concepts and their semantic connections from a set of related documents under the same topic.
- Traditional relation extraction systems require people to the pre-specify the set of relations of interest. Obviously, it is not appropriate for the news documents with diverse relation schemas.
- Given a query topic, a user is often expected to understand core topic information serving by a large conceptual graph, rather than having a collection of relevant documents.

Motivation

We present a system that extracts salient entities, concepts, and their relationships from a set of related documents, discovers connections within and across them, and presents the resulting information in a graph-based visualization.



e-mail scandal（邮件门事件）

is relevant to（与其相关）

Hillary Clinton（希拉里克林顿）

Republicans presidential candidate（共和党总统候选人）

US presidential campaign（美国总统选举）

who is currently second in the polls（目前票选排名第二）

has suspended（退出）

2016 US presidential race（2016年美国总统大选）

as a powerful competitor of（作为有力的竞争对手）

Ben Carson（本卡森）

was first revealed in（首次曝光于）

support（支持）

entered（参加了）

winter of 2015（2015年冬天）

might be effecting（可能会影响）

Billionaire Donald Trump（唐纳德特朗普）

graduated from（毕业于）

University of Pennsylvania（宾夕法尼亚大学）

presently voted（目前已赢得）

68 percent of leaded Republicans（百分之六十八的共和党选票）

Figure: Example of a conceptual graph on the topic “*presidential election of the US*”

Outline

1. Motivation
2. Problem Definition
3. System Architecture
4. Approach and Implementation
    - Input Data
    - Fact Extraction
    - Fact Filtering
    - Conceptual Graph Construction
5. Experiments
6. Conclusions and Future Work

Problem Definition

The objective of our system is to assist users in quickly finding meaningful and salient connections and facts from a collection of relevant documents, 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.

Problem Definition

- **Subtask 2: Fact Filtering.** Given a specified document topic T, the goal of the subtask is to find a subset of $F_t \subseteq F_c$, 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 $F_t \subseteq F_c$ 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.

Outline

1. Motivation
2. Problem Definition
3. System Architecture
4. Approach and Implementation
    - Input Data
    - Fact Extraction
    - Fact Filtering
    - Conceptual Graph Construction
5. Experiments
6. Conclusions and Future Work

System Architecture

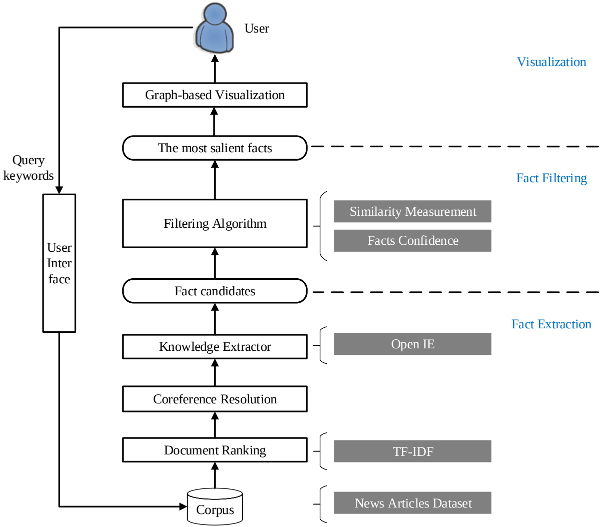


Figure: System architecture

Outline

1. Motivation
2. Problem Definition
3. System Architecture
4. Approach and Implementation
   - Input Data
   - Fact Extraction
   - Fact Filtering
   - Conceptual Graph Construction
5. Experiments
6. Conclusions and Future Work

Outline

1. Motivation

2. Problem Definition

3. System Architecture

4. Approach and Implementation
   - Input Data
   - Fact Extraction
   - Fact Filtering
   - Conceptual Graph Construction

5. Experiments

6. Conclusions and Future Work

Input Data

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[1] clusters of 10 documents.

The articles in our dataset stem from a larger news document collection 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.

[1] Document Understanding Conference, https://duc.nist.gov/

Input Data

Table: 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

Outline

1. Motivation
2. Problem Definition
3. System Architecture
4. Approach and Implementation
   - Input Data
   - Fact Extraction
   - Fact Filtering
   - Conceptual Graph Construction
5. Experiments
6. Conclusions and Future Work

Fact Extraction

Document Ranking. The system first select the words appearing in the document collection with sufficiently high frequency as topic words, and computes standard TF-IDF weights[2] for each word. Documents under the same topic are ranked according to the TF-IDF weights of the topic words in each document. The top-k documents for every topic are selected for further processing.

Coreference Resolution. Pronouns and other form of coreference are resolved in each document using Stanford CoreNLP system. “she” may be replaced by “Angela Merkel”, for instance.

Sentence Ranking. Our system computes the TextRank importance scores[3] for all sentences within the ranked top-k document list. It then considers only those sentences with sufficiently high scores.

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

[3] ttps://github.com/letiantian/TextRank4ZH/blob/master/README.md

Open-Domain Knowledge Extraction

Open-Domain Knowledge Extraction. Our candidate fact extraction is based on a publicly available system for open information extraction, namely the KnowItAll project's Open IE 4[4]

Considering an example consisting of the following two sentences:

"George Washington was the first President of the United States, the Commander-in-Chief of the Continental Army during the American Revolutionary War."

- 0.95 ("George Washington", "was", "the first President of the United States")
- 0.88 ("George Washington", "was", "the Commander-in-Chief of the Continental Army")

[4] https://github.com/knowitall/openie

Open-Domain Knowledge Extraction

Considering an example consisting of the following two sentences:

"**He** *presided over the convention that drafted the current United States Constitution and during his lifetime was called the 'father of his country'* "

- 0.45 ("**He**", "presided", "over the convention")
- 0.90 ("the convention", "drafted", "the current United States Constitution")

Noting that:

When the ambiguous pronoun "**He**" is replaced with "**George Washington**",

- 0.93 ("George Washington", "presided", "over the convention")

Outline

1. Motivation
2. Problem Definition
3. System Architecture
4. Approach and Implementation
   - Input Data
   - Fact Extraction
   - Fact Filtering
   - Conceptual Graph Construction
5. Experiments
6. Conclusions and Future Work

Fact Filtering

The filtering algorithm aims at hiding less representative facts in the visualization, seeking to retain only the most salient, confident and compatible facts. This is achieved by optimizing for a high degree of coherence between facts with high confidence.

The joint optimization problem can be solved via integer linear programming (ILP), as follows:

$$
\begin{aligned}
\max_{x,y} \quad & \alpha^T x + \beta^T y && (1)\\
\text{s.t.} \quad & 1^T y \leq n_{\max} && (2)\\
& x_k \leq \min\{y_i, y_j\} && (3)\\
& \quad \forall \;\; i < j, i, j \in \{1, \ldots, M\}, \\
& \qquad k = (2M - i)(i - 1)/2 + j - i \\
& x_k, y_i \in \{0, 1\} \, \forall i \in \{1, \ldots, M\}, k && (4)
\end{aligned}
$$

Fact Filtering

ILP method:

Here, $\mathbf{x} \in \mathbb{R}^N$, $\mathbf{y} \in \mathbb{R}^M$ with $N = (M+1)(M-2)/2 + 1$. The y_i are indicator variables for facts t_i: If y_i is true, t_i is selected to be retained. x_k represents the compatibility between two facts $t_i, t_j \in T$ $(i, j \leq M, i \neq j)$, where $T = \{t_1, \ldots, t_M\}$ is a set of fact triples containing M elements. β_i denotes the confidence of a fact, and $n_{\max}$ is the number of representative facts *desired by the user*. α_k is weighted by similarity scores $sim(t_i, t_j)$ between two facts t_i, t_j, defined as $\alpha_k = sim(t_i, t_j) = \gamma \dot{s}_k + (1-\gamma)\dot{l}_k$. Here, 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, 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. The constraints guarantee that the number of results is not larger than $n_{\max}$. If x_k is true, the two connected facts t_i, t_j should be selected, which entails $y_i = 1$, $y_j = 1$.

Outline

1. Motivation
2. Problem Definition
3. System Architecture
4. Approach and Implementation
    - Input Data
    - Fact Extraction
    - Fact Filtering
    - Conceptual Graph Construction
5. Experiments
6. Conclusions and Future Work

Conceptual Graph Construction

Merge Equivalent Concepts and Add Relations. Expert annotators can merge potential entities and concepts stemming from the fact filtering process.

- **Literal features of entities**. e.g., Billionaire Donald Trump, Donald Trump, Donald John Trump, Trump, etc. all refer to the same person.
- **Entity linking from search engine**. For NER, they can use the powerful entity linking ability from a search engine for deciding on coreference. *Align, Disambiguate and Walk*[5] tool is used for semantically similarity computation between concepts for coreference.

Annotators were able to add up to three synthetic relations with freely defined labels to connect the subgraphs into a fully connected graph.

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

Conceptual Graph Construction

Conceptual Graph Generation. we further generate final conceptual graph which works in two steps:

- **Trained a binary classifier**. We trained a binary classifier by the topic words with high frequency extracted from different topics to identify the important topic entities and concepts in the set of all potential concepts, with a random forest as the model.
- **Relied on a heuristic strategy**. We iteratively remove the weakest concepts with relatively lower score until only one connected component of 25 entities and concepts or less remains, which is used as the final conceptual graph.

Noting that:

The recommended[a] maximum size of a concept graph is 25 concepts, which we use as a constraint.

[a] Banko, M., Cafarella, M. J., Soderland, S., Broadhead et al., Open information extraction from the web, IJCAI 2007.

Outline

1. Motivation
2. Problem Definition
3. System Architecture
4. Approach and Implementation
   - Input Data
   - Fact Extraction
   - Fact Filtering
   - Conceptual Graph Construction
5. Experiments
6. Conclusions and Future Work

Experimental settings

Parameter setting

- **Sentence-level extractions**. 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[a].
- **An empirical study**. We further conduct 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.

[a] 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% ($\kappa = 0.60$))

Experimental settings

Evaluation measures

Three standard metrics

Sentence-level extractions

- Precision (P)
- Recall (R)
- F-score (F1)

An empirical study

Quality of the conceptual graph

- $TopicCon\_Coverage$
- $Avg\_Confidence$
- $Avg\_Compatibility$

Noting that:

$$Avg\_Confidence(f_i, n) = \frac{\sum_{i=1}^{n} conf(f_i)}{n} \tag{5}$$

where $conf(f_i)$ denotes the confidence score of each fact f_i, n is the number of facts which involved in the final conceptual graph.

Evaluation and Results Analysis

Noting that:

$$Avg\_Compatibility(f_i, f_j, n) = \frac{\sum_{i=1}^{n} \sum_{j>i} cmp(f_i, f_j)}{c_n^2}, \tag{6}$$

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.

Performance analysis of our extraction approach

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

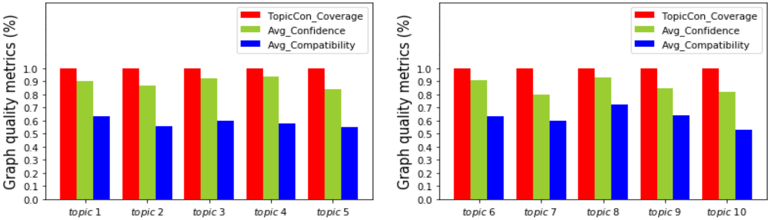
OpenIE methods	**#Topic 1**			**#Topic 2**			**#Topic 3**			**#Topic 4**			**#Topic 5**		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
Our approach (without coref)	0.43	0.29	0.56	0.44	0.27	0.33	0.65	0.24	0.35	0.47	0.33	0.39	0.45	0.30	0.36
Our approach	0.86	0.85	**0.85**	0.78	0.74	**0.76**	0.95	0.92	**0.93**	0.95	0.82	**0.88**	0.92	0.78	**0.84**

Evaluation and Results Analysis

Table: 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
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	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**

Quality analysis of the conceptual graph



(a)

(b)

Evaluation and Results Analysis

The results indicates that:

Our approach achieved 100% coverage rate of topic entities and concepts ($TopicCon\_Coverage$), 87% confidence score ($Avg\_Confidence$), and 68% fact compatibility ($Avg\_Compatibility$) over ten document topics.

- The proposed fact filtering approach is capable to select high confident and salient facts from the extracted candidate facts, however, may not guarantee their better compatibility, which needs to be further explored.
- The final generated conceptual graph has higher coverage rate of topic entities and concepts, which demonstrate the importance of the heuristic strategy in the process of conceptual graph construction.

Outline

1. Motivation
2. Problem Definition
3. System Architecture
4. Approach and Implementation
   - Input Data
   - Fact Extraction
   - Fact Filtering
   - Conceptual Graph Construction
5. Experiments
6. Conclusions and Future Work

Conclusions and Future Work

Conclusions

- Our system is intended to aid users in quickly discerning salient connections in a collection of documents, including via graph-based visualizations. Experiments on two real-world datasets demonstrate the effectiveness of our proposed approach.

Future Work

- The fact filtering algorithm will give greater consideration to the context of the triples, to enhance compact connections.
- The fact fusion problem in generating the final conceptual graph needs to be further explored for the fully automated conceptual graph construction for specified domain is possible.

Codes and Datasets

- We release the codes and datasets related to this system at: https://shengyp.github.io/vmse.

Thanks for your time! Any question?