EventSKG: a Framework for Public Safety Event Knowledge Graph Construction using Social Network Data

Xing Yang
National University of Defense Technology

Chenhui Xu (✉ xu.chenhui@outlook.com)
Chinese Academy of Sciences Hefei Institutes of Physical Science

Dongtao Zhu
National University of Defense Technology

Yuanmiao Gui
Chinese Academy of Sciences Hefei Institutes of Physical Science Institute of Intelligent Machines:
Chinese Academy of Sciences Institute of Intelligent Machines

Shaomin Sun
Chinese Academy of Sciences Hefei Institutes of Physical Science Institute of Intelligent Machines:
Chinese Academy of Sciences Institute of Intelligent Machines

Wenbo Li
Chinese Academy of Sciences Hefei Institutes of Physical Science Institute of Intelligent Machines:
Chinese Academy of Sciences Institute of Intelligent Machines

Yuyang Tang
Chinese Academy of Sciences Hefei Institutes of Physical Science Institute of Intelligent Machines:
Chinese Academy of Sciences Institute of Intelligent Machines

Research

Keywords: Knowledge Graph, Social Network Data, FLAT, EventSKG

Posted Date: October 20th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-944615/v1

License: ☑️ This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License
EventSKG: a Framework for Public Safety Event Knowledge Graph Construction using Social Network Data

Xing Yang¹, Chenhui Xu²,³*, Dongtao Zhu¹, Yuanmiao Gui², Shaomin Sun², Wenbo Li² and Yuyang Tang²,³

Abstract

The social network data contain a great deal of unstructured knowledge which is useful to understand public safety incidents. The structure of social network data, such as news, statistical statements, social network messages, government reports and so on, is complicated. So, it is difficult to obtain knowledge from social network data because social network data are multi-source heterogeneous. Knowledge graph is one of the effective tools for processing such multi-source heterogeneous data. However, the previous works didn’t notice the particularity of entities in public safety incidents and the linguistic characteristics of Chinese. Moreover, previous works on event knowledge graph mainly focus on single data source which make it is difficult to process multi-source social network data. Thus, the construction of a public safety incident knowledge graph using complex social network data in the Chinese environment is still a difficult problem. In this paper, a framework for public safety event knowledge graph construction towards Chinese social network data, namely EventSKG, is proposed. EventSKG contains a direct mapping module for knowledge extraction from structured data, an Event-Specific FLAT model for entity extraction from unstructured data, a manual template for relation extraction from unstructured data, and an integration surgery to connect the triples obtained from different data source. An event knowledge graph related to the flood disaster in 2020 in China is constructed via EventSKG framework, which contains 14397 entities and 43168 relations. The event knowledge graph will form associations around network data from various sources for the same event, not only enabling humans to better understand the event, but also providing basic knowledge base for future intelligent understanding of events.

Keywords: Knowledge Graph; Social Network Data; FLAT; EventSKG

Introduction

There will be a large number of data published on the Internet when an event occurs. The data is helpful to better understand the event. When humans are faced with this kind of social network data, humans can easily find the main theme, entities, relations, and attributes of the events or entities, but machines don’t get such ability[1, 2]. With the increasing data sources and amount of data, multi-source heterogeneity makes it much more difficult for machines to organize knowledge hidden in the data. For example, when a disaster occurs, various kinds of data including statistical statement, social media message, news and government reports will be generated[3], machines have no capacity to acquire knowledge among all
kinds of data mentioned above, because machines don’t know semantic information embedded in loosely organized data of the event.

Knowledge graph has excellent ability of expressing relation between large-scale multi-source heterogeneous data, which makes it a good way for machines to extract and store knowledge related to events. Knowledge graph is a structured knowledge base represented as graph, the nodes in the graph represent entities and the edges represent the relations between the entities [4]. Therefore, with a knowledge graph everything in the world could be connected, and every relation between entities can be represented in the computer system.

Knowledge graphs are divided into general knowledge graphs and domain knowledge graphs[4, 5]. General knowledge graph refers to a knowledge graph that is oriented to all fields and is often applied to business scenarios such as search, recommendation and question answering system. It emphasizes the breadth of knowledge and more on the scale of entities, so it is difficult to produce a complete and global ontology layer unified management. The representative works of general knowledge graph include WikiData[6], DBpedia[7], and YAGO[8]. Domain knowledge graph is a domain oriented knowledge graph, which requires more rapid scale expansion, more complex knowledge structure and higher knowledge quality[5, 9]. Domain knowledge graphs have been built in many fields, such as education[10, 11], materials[12], health[9, 13], recommender systems[14–19], workload predication[20]. To better understanding the events, a domain knowledge graph is helpful. The core techniques of constructing a domain knowledge graph include entity extraction, relation extraction, knowledge fusion and knowledge storage[4].

However, few knowledge graphs have been constructed for single event. Researchers have proposed the necessity and advantages of building a knowledge service framework for events[21]. Following this idea, some domain knowledge graphs for emergencies have been constructed in domains like earthquake[22], typhoon[23]. Unfortunately, these researches concentrate on single-source data but not multi-source data with different structures, which can’t take full advantage of the open connectivity capabilities of the knowledge graph. Previous works[24] on event related knowledge mainly focus on the availability of reference knowledge repositories containing comprehensive representations of events and temporal relations, which using existing knowledge and semi-structured data [25, 26]. EventKG[25] constructs a multilingual event-centric temporal knowledge graph that addresses this gap, but it doesn’t take notice of the importance of unstructured text data which contain rich information.

In addition, knowledge extraction in event scenario is complicated. Knowledge extraction contains entity extraction and relation extraction. For unstructured data, the combination of conditional random fields(CRF) and neural network has been confirmed to be an effective entity extraction method[27]. BiLSTM-CRF[27] achieves great success in entity extraction tasks, and the introduction of attention mechanisms enhances this advantage[28]. The appearance of Transformer[29] makes things to another level for its adopting fully-connected self-attention to model the long-distance dependencies in a sequence[30]. Unfortunately, these methods can’t fit event-specific scenario well because the entities are much more different with the entities in common scenarios. Simply adopting pre-trained language model to
event-specific scenario can’t achieve great effect. This requires us to make some modifications to these models. Moreover, compared with English entity extraction task, Chinese entity extraction task is more complex and difficult since it usually involves word segmentation[30]. It makes it more difficult to extract entity in event-specific and Chinese scenario. There are also some problem in relation extraction in this scenario. For example, the lack of priori relation types makes machines can’t know what relation they actually want to know.

To fill in the blank of event knowledge graph based on social network data and solve the problems mentioned above, event-centred knowledge graph is introduced as the way to organize the knowledge hidden in the multi-sources heterogeneous data. To construct an event-centred knowledge, a novel framework named Event-Specific Knowledge Graph(EventSKG) of building the event-centred knowledge graph is proposed. The EventSKG takes triples from both structured and unstructured data separately and fuses them. Specially, to triples obtained from structured data provide relation types to help extract relation from unstructured data.

**Method**

In this section, we discuss the construction method of the event-centred knowledge graph.

**Event-Specific Knowledge Graph Construction Framework**

In order to construct a event-centred knowledge graph, in this paper, a novel approach named EventSKG is proposed. EventSKG aims to construct a a knowledge from multi-source heterogeneous data related to single event.

Figure illustrates the event-centred knowledge graph construction framework. At first, a direct mapping method is used to transfer structured data from tables to knowledge triples. Then, FLAT model is fine-tuned to a Event-Specific FLAT using unstructured social network data in Chinese. Using Event-Specific FLAT, entities hidden in the unstructured data can be extracted. Then, a manual template is generated which obtains priori relation types from previous triples, to extract the relation between the entities in the unstructured data. After extraction task, an integration module is followed to fuse the triples obtained from multi-source data. Finally, the fused triples are stored in a graph database to get the visible knowledge graph.

**Direct Mapping**

For structured data, we can use direct mapping (DM) method as shown in Figure to transfer data from relational database to triples. Triple is the basic atom of a knowledge graph, often represented as \((h, r, t)\), where \(h\) refers to a head entity (subject), \(r\) refers to a relation (predicate) and \(t\) refers to a tail entity (object). By connecting numbers of triples contain flood disaster knowledge, a knowledge graph simply regarding every head entity and tail entity in triples as a node and every relation in triples as edge can be simply gotten.

DM contains the following four basic rules:

1. Map tables in the relational database to subgraphs;
2. Map the column names in the relational database to relations or attributes;
(3) Map the primary key of each row in the relational database to a head entity;
(4) Map the value of each cell in the relational database to a tail entity. If the
value of the cell corresponds to a foreign key, the column corresponds to a relation,
and the cell should map to the entity to which the foreign key corresponds.

For semi-structured data, we need to design a rule-based wrapper to handle dif-
derent data. For different types of data, the wrapper chooses different types of pro-
cessing methods. In flood disaster scenario, the main form of data is structured
statistical reports collected by different organizations. Although this kind of data
is semi-structured data, there is no fixed data structure, but the table function in
modern relational database can store this kind of statistical report in a single table.
The semi-structured data can then be transferred into structured data. Therefore,
the knowledge in semi-structured data of flood disaster can be extracted accord-
ing to the extraction method similar to structured data with some differences. In
semi-structured situation, it should be considered that the potential NULL value,
which will lead to incomplete triples, which will be harmful when align and embed
the knowledge graph.

**Event-Specific Flat LAttice Transformer**

As show in Figure 3 FLAT[30] model first gets a lattice from characters and flatten
it into counterpart to get a flat-lattice. Flat-lattice is a set of spans, each of which
is contains a token, a head and a tail. The token is a character or a word, and
the head and tail denote the representing the corresponding token’s first and last
character’s position in the sequence. Due to the different lengths of the spans, to
encode the interactions among spans, we should use the relative position encoding of
spans. That means instead of directly encoding three kinds of relations (intersection,
inclusion and separation), we should use dense vector calculated by continuous
transformation of the head and tail information to model the rations among all
the spans. To indicate more detail information, we let \(head[i]\) and \(tail[i]\) denote
the head and tail position of span \(x_i\). Thus, we can define four kinds of relative
distances indicate the relation between \(x_i\) and \(x_j\).

\[
d_{ij}^{(hh)} = head[i] - head[j], \tag{1}
\]

\[
d_{ij}^{(ht)} = head[i] - tail[j], \tag{2}
\]

\[
d_{ij}^{(th)} = tail[i] - head[j], \tag{3}
\]

\[
d_{ij}^{(tt)} = tail[i] - tail[j], \tag{4}
\]
where $d_{ij}^{(hh)}$ denotes the distance between head of $x_i$ and head of $x_j$. Then we can get the final relative position encoding:

$$R_{ij} = ReLU(W_r(p_d^{(hh)} \oplus p_d^{(ht)} \oplus p_d^{(th)} \oplus p_d^{(tt)})),$$  \hspace{1cm} (5)

where $W_r$ is a learnable parameter, $\oplus$ denotes the concatenation operator, $p_d$ is calculated by following equation:

$$p_d^{(2k)} = \sin(d/10000^{2k/d_{model}}),$$ \hspace{1cm} (6)

$$p_d^{(2k+1)} = \cos(d/10000^{2k/d_{model}}),$$ \hspace{1cm} (7)

where $k$ denotes the index of dimension of position encoding.

In traditional Transformer model, the self-attention is calculated by concatenating the result of multi heads. The result of per head is calculated as:

$$\text{Att}(A, V) = \text{softmax}(A)V,$$ \hspace{1cm} (8)

$$A_{ij} = \frac{Q_iK^T_j}{\sqrt{d_{head}}},$$ \hspace{1cm} (9)

$$[Q, K, V] = E_x[W_q, W_k, W_v],$$ \hspace{1cm} (10)

where $E$ is the token embedding or the output of the last layer, $W_q, W_k, W_v$ are learnable parameters, $d_{head}$ is the dimension of each head. This method use absolute position encoding.

In FLAT, a variant of self-attention is adopted by replacing $A$ with $A^*$ in Eq.(9),

$$A^*_{i,j} = W_q^TE_x^T_{x_j}x_{k,E} + W_q^TE_{x_j}^TR_{ij}W_{k,R} + u^TE_{x_j}W_{k,E} + v^TR_{ij}W_{k,R},$$ \hspace{1cm} (11)

where $W, W_{k,E}, W_{k,R}$, $u$ and $v$ are learnable parameters.

After FLAT, we only take the character representation into output layer, followed by a Conditional Random Field(CRF) [31], just like BiLSTM-CRF[27].

As FLAT is designed as a general-purpose named entity recognition model that is trained on different Chinese datasets, despite its high performance, it’s still challenging to maintain its performance when adapting for events domain text that
contain a considerable number of domain-specific terms. This paper fine-tunes the FLAT model to adapt event-specific entity extraction task.

While fine-tuning FLAT, in the input layer, event-related text (in this case it is flood disaster) is used to get their lattice the same way as [32]. This operation not only collects the semantic information of individual words in sentences, but also collects the semantic information of phrases, making the model able to capture the integral semantic of phrases. For example, '北京' will be both known by the model as 'north capital' when understanding it word by word while '北' means 'north' and '京' means 'capital', and known as 'Beijing' or 'Peking' which is a famous city in China when understanding it as a whole.

In the output layer, since the named entities classes in event-specific scenario have different tags with which in original task in FLAT, the output of neural networks should be adjusted to fitting the needed tags of event. The FLAT model is fine-tuned as follows:

\[
p(T_i) = \text{softmax}(T_i W^T + b)_k,
\]  

where \( k \) represents the indexes of tags of characters, \( p \) is the probability distribution of assigning each \( k \) to token \( i \), and \( T_i \in \mathbb{R}^H \) is the final hidden representation, which is calculated by FLAT.

Consequently, Event-Specific FLAT is able to extract diverse types of event-specific entities.

**Template-based Relation Extraction**

Since the accuracy of existing methods can not meet the requirement of constructing knowledge when extracting complex relation, we must do some manual intervention.

So far, it has been gotten that a lot of triples from structured data and a lot of entities from unstructured data, but not the relation hidden in the entities extracted from unstructured data. Because the relations between entities in each event are very different, it not easy to obtain relation directly from text. Luckily, the triples obtained from structured data provide sufficient prior knowledge. Using the relation from triples obtained from structured data, we can manually establish a template contains various kinds of relation. In the template, every relation should be defined a group of triggers, when traverse the entire sentence, once the trigger is found, a triple should be generated. The trigger should be defined not only able to notice the relation, but also the upper and lower bits of the entity since the triples are directional.

**Integration**

After extracting knowledge from different data sources, it is needed to fuse those subgraphs to one knowledge graph to form associations between every entity. The core problem of knowledge fusion is entity alignment. An intuitive approach to entity alignment is to calculate the similarity of entity names, since in the event scenario, the main problem is co-reference resolution rather than polysemy.

An easy way to calculate the similarity of an entity is to calculate the edit distance of the attribute. The edit distance is defined as:
\[ D(i,j) = \min \begin{cases} 
D(i-1,j) + \text{del}[x(i)], \\
D(i,j-1) = D(i-1,j) + \text{ins}[y(j)], \\
D(0,j) = D(i-1,j-1) + \text{sub}[x(i),y(j)]. 
\end{cases} \] (13)

with boundary conditions:

\[ \begin{cases} 
D(0,0) = 0, \\
D(i,0) = D(i-1,0) + \text{del}[x(i)] & 1 < i \geq N, \\
D(0,j) = D(0,j-1) + \text{del}[y(j)] & 1 < j \geq M. 
\end{cases} \] (14)

where \( D(i,j) \) means distance between \( i \)th character of string \( x \) and \( j \)th character of string \( y \). And \( \text{del}[x(i)],\text{ins}[x(i)],\text{sub}[x(i),y(j)] \) represent the cost of delete, insert and substitute this character. Thus we can customize a threshold to determine the acceptable fusion editing distance.

**Knowledge Storage**

After the knowledge extraction is completed, with the increasing of the extracted knowledge, a reasonable storage model will help to store massive knowledge and facilitate the reasoning engine to use the knowledge in the knowledge base. Neo4j is the most popular graph database at present. Based on the attribute graph model, it designs specific storage schemes for the nodes, node attributes, edges and edge attributes of the attribute graph. In addition to graphs, Neo4j supports multiple types of data. Neo4j has powerful graph traversal capabilities thanks to its strategy of storing the graph structure and the attributes on the graph separately. More search algorithms can be better applied on the knowledge graph stored by Neo4j.

For triples of knowledge extracted from text, you can use the Python and Py2neo module to build an automated storage script to store the knowledge as algorithm 1.

**Algorithm 1 Knowledge Storage**

```python
1: Load three concepts of Nodes, relation and Graphs in py2neo module.
2: for triple \( \in \) triples do
3: Match all the exist node with current triple,
4: if No nodes exist then
5: Create nodes \( h \) and \( t \),
6: Create edge \( r \),
7: else
8: Create the nonexistent nodes,
9: Create edge \( r \).
10: end if
11: end for
12: Import the created nodes and edges into the graph.
13: return result
```

**Experiment**

In this section, specific data are used to realize the construction of Event-Specific Knowledge Graph mentioned above, so as to verify the feasibility of proposed method.
Data
As mentioned, this paper uses the data related to flood disasters in China in the spring of 2020 to construct the knowledge graph. The data of this experiment includes the statistical statement of flood disaster in 1058 countries in 12 Provinces of China in 2020 collected by different statistical institutions, including the data of affected population, death population, damaged houses, damaged crops, damaged property and other aspects. This paper also uses unstructured text data including 39 news articles, 1049 social network messages and 24 government reports as a supplement of knowledge.

Event-Specific FLAT
We take BiLSTM-CRF [27] as the baseline models. By comparing with Lattice LSTM and basic FLAT model, Event-specific FLAT achieves the best in precision, recall and F1 score. As shown in Table ??, Event-specific FLAT model outperforms FLAT by 3.17% in F1 score, 3.56% in precision and 2.77% in recall rate in average. FLAT’s high performance reveal its superior ability in Chinese entity extraction scenario. Event-specific FLAT further improves performance by mining features of event data.

Using trained Event-Specific FLAT, EventSKG extract 3154 entities from government reports, news articles and social network messages. Since there is no open relationship extraction model capable of establishing a knowledge graph with high confidence so far, EventsKG adopts the method of manual template matching to extract the relations that have appeared in the previously obtained relation list from structured data.

Knowledge Extraction from Structured Data
For the structured and semi-structured data, the direct mapping method is adopted. Firstly, the tables in the relational database are imported into the a file, that is, the tables in the relational database are created as classes. Data is imported using lists in Python, and head entities are created according to primary keys in the table. Creates a tail entity for the numeric values of cells in the table other than the primary key, and creates a relation with the established head entity based on the column name of the column in which the cell is located. Then we get the raw triples contains abundant knowledge.

Then, by combining entities from unstructured data and calculating the edit distance between entities, the entities in different tables are aligned. Direct Mapping finally gets 13178 entities and 42781 groups of relations from structured data.

Integration and Storage
Using editing distance, EventSKG aligns the entities from news, social network messages, government reports, and statistical statement whose structure are totally different, and gets 14397 entities and 43168 groups of relations.

Using Py2neo to concatenate Neo4j, for the head entity and tail entity in the triples, create non-repeating nodes, and create the edges between nodes according to the relation in the triples, and store the obtained relation and edges in Neo4j, so as to complete the storage of knowledge graph by using graph database.

The knowledge graph constructed in this paper contains 47 types of entities and 46 types of relations. Figure shows part of the knowledge graph.
**Result and Discussion**

**Result**

Using EventSKG, a event-specific knowledge graph which contains 14397 entities and 43168 groups of relations was constructed with multi-source heterogeneous event data from social network. Specially, the part of entity extraction from the unstructured data achieves a great performance that a precision rate of 77.45%, a recall rate of 76.43% and the F1 score of 76.94%. Table ?? shows the confidence interval of the precision, recall and F1 score of Event-Specific FLAT model after 10 repeated experiments.

Since other parts of the EventSKG framework don’t contain randomness, all the processes are uniquely determined by data. Direct Mapping gets 13178 entities and 42781 relations from the disaster statistical statement of 1058 countries. Relation extraction part gets 46 types of relations from the triples and extract 383 relations from entities found by Event-Specific FLAT.

**Discussion**

The EventSKG successfully construct an event knowledge graph using multi-source heterogeneous social network data and the Event-Specific FLAT achieves a great performance in Chinese event text scenario in entity extraction task.

The reason why EventSKG can work is that the methods processing different types of data form interaction with each other, the triples from structured data provide priori of the relation types to the relation extraction part. The Event-Specific FLAT use lattice to solve the problem of semantic group in Chinese scenario, and the fine-tuned model can better adapt to event scenarios.

However, the EventSKG framework still has some shortcomings. Firstly, the framework relies on structured data. Since open retrieval of relations from unstructured data is currently not easy to achieve, EventSKG must get a reliable source of relation type to extract relation from open data. Secondly, this framework is currently limited to Chinese scenario, although we think it will be easy to transfer to other language by replacing the FLAT part to other model. Finally, due to the limited scale of data, this framework is still expected to be validated with more data.

**Conclusion**

In this paper, in view of the lack of an effective organization and utilization of multi-source heterogeneous data from Internet, and the lack of knowledge base in the event-specific domain, a framework of the event-centred knowledge graph named EventSKG is proposed. EventSKG summarized the information and data characteristics of event data, and the general principles and steps of constructing knowledge graph. This paper also carries out the construction of the event-centred knowledge graph, which provides the basic knowledge source for the knowledge inference machine based on machine learning. This paper also proposes a event-specific improved FLAT model which has excellent performance on event entity extraction in Chinese scenario.

In this paper, multi-source heterogeneous data of flood disaster are used for the construction of knowledge graph for the first time, which solves the problem that
it is difficult to form a unified data management and application method due to
the wide source of flood disaster data and low organization efficiency. In this paper,
a large amount of knowledge contained in flood disaster data is excavated and
organized into the form of knowledge graph, and the Neo4j graph database is used
to conduct unified management of flood disaster knowledge, forming a flood disaster
knowledge base with organizational structure.

For the future research, the development of reasoning application based on event-
centred knowledge graph is the focus of the work. Based on vectoquantization represen-
tation[1, 33], machine learning method can be used to design knowledge inference
machine for flood disaster scenario.

Acknowledgements
We would like to thank the authors of the literature cited in this paper for contributing useful ideas to this study.

Funding
This work was supported by Anhui Provincial Natural Science Foundation under Grant 2108085J19, the National Nature Science Foundation of China under Grant 41871302 and the Development Program of China under Grant 2018YFB1004600.

Abbreviations
EventSKG: Event-Specific knowledge graph; FLAT: Flat Lattice Transformer; CRF: Conditional random field;
LSTM: Long short-term memory; BiLSTM-CRF: Bidirectional long short-term memory conditional random field

Availability of data and materials
The datasets used and/or analysed during the current study are available from the corresponding author on
reasonable request.

Competing interests
The authors declare that they have no competing interests.

Authors’ contributions
Yang came up with the initial framework of Event-Specific knowledge graph. Xu designed the Event-Specific FLAT
model and implemented the experiments. Zhu and Gui designed the Direct Mapping part. Li provided great help in
writing. Sun supported the experimental design process and give some advice. Tang designed the integration and
storage parts. All authors read and approved the final manuscript.

Author details
1State Key Laboratory of Pulsed Power Laser Technology, Advanced Laser Technology Laboratory of Anhui Province, Electronic Countermeasure Institute, National University of Defense Technology, Hefei 230037, China.
2Institute of Intelligent Machines, HFIIPS, Chinese Academy of Sciences, Hefei 230031, China. 3University of Science and Technology of China, Hefei 230026, China.

References
   IEEE Transactions on Knowledge and Data Engineering 29(12), 2724–2743 (2017)
2. LiuQiao, L., DuanHong, L., et al.: Knowledge graph construction techniques. Journal of computer research and
development 53(3), 582 (2016)
   knowledge graph leveraged by wikidata. In: Companion Proceedings of The 2019 World Wide Web Conference,
   pp. 1232–1239 (2019)
5. Lin, J., Zhao, Y., Huang, W., Liu, C., Pu, H.: Domain knowledge graph-based research progress of knowledge
   57(10), 78–85 (2014)
    for academic data mining. In: Proceedings of the 27th ACM International Conference on Information and

**Tables**

**Figures**

Figure 1. EventSKG framework. The overview of EventSKG framework, the framework transfers structured and unstructured data to knowledge graph.
Figure 2 Direct Mapping  Figure 2. Direct Mapping, in this figure, a table which contains affected population and death information will be mapped into several triples that can constitute a small-scale knowledge graph, with primary keys mapped into a head entity, column names mapped into a relation, values mapped into a tail entity.

Table 1 Comparison of results

<table>
<thead>
<tr>
<th></th>
<th>BILSTM</th>
<th>Lattice LSTM</th>
<th>FLAT</th>
<th>Event-Specific FLAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision %</td>
<td>71.39</td>
<td>73.21</td>
<td>73.89</td>
<td>77.45</td>
</tr>
<tr>
<td>Recall %</td>
<td>69.79</td>
<td>71.27</td>
<td>73.66</td>
<td>76.43</td>
</tr>
<tr>
<td>F1 %</td>
<td>70.58</td>
<td>71.27</td>
<td>73.66</td>
<td>76.94</td>
</tr>
</tbody>
</table>

Table 2 Confidence Interval of Event-Specific FLAT

<table>
<thead>
<tr>
<th></th>
<th>Confidence Lower Limit</th>
<th>Mean</th>
<th>Confidence Upper Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision %</td>
<td>76.59</td>
<td>77.45</td>
<td>78.31</td>
</tr>
<tr>
<td>Recall %</td>
<td>74.91</td>
<td>76.43</td>
<td>77.95</td>
</tr>
<tr>
<td>F1 %</td>
<td>75.74</td>
<td>76.94</td>
<td>78.13</td>
</tr>
</tbody>
</table>
Figure 4: Visual presentation of knowledge graph constructed using EventSKG framework.

Figure 4: Knowledge Graph of Flood Disaster happened in China in 2020