NAI-SEA at SemEval-2018 Task 5: An Event Search System

Yingchi Liu, Quanzhi Li and Luo Si
Alibaba Group, Inc
Bellevue, WA 98004, USA
{yingchi.liu,quanzhi.li,luo.si}@alibaba-inc.com

Abstract

In this paper, we describe Alibaba's participating system in the semEval-2018 Task5: Counting Events and Participants in the Long Tail. We designed and implemented a pipeline system that consists of components to extract question properties and document features, document event category classifications, document retrieval and document clustering. To retrieve the majority of the relevant documents, we carefully designed our system to extract key information from each question and document pair. After retrieval, we perform further document clustering to count the number of events. The task contains 3 subtasks, on which we achieved F1 score of 78.33, 50.52, 63.59 , respectively, for document level retrieval. Our system ranks first in all the three subtasks on document level retrieval, and it also ranks first in incident-level evaluation by RSME measure in subtask 3.

1 Introduction

In this paper we present our system developed for participanting in the semEval-2018 Task5: Counting Events and Participants in the Long Tail. Given a set of questions and a corpus of documents mainly from news articles, the system needs to provide a numeric answer together with the supporting documents that directly relate to the answer (Postma et al., 2018). According to the official task rule, participants can also optionally provide the text mentions of events in the documents, but we did not participate this year. The task contains 3 subtasks. Subtask 1 is to retrieve all the relevant documents related to one single event asked in the question. Subtask 2 and subtask 3 require the system not only retrieve relevant documents, but also count the number of events or number of participants. Event detection and extraction has been intensively studied (Choubey and Huang, 2017; Nguyen et al., 2016a,b; Nguyen and Grishman, 2016; Feng et al., 2016; Ji and Grishman, 2008). Most of those research used a corpus of annotated documents for training. In this task, annotated documents were not provided, but the key factors to retrieve relevant documents are provided by the questions. Therefore, our system starts from a document retrieval system via key information extraction and matching and follows a document clustering component.

2 System Description

We developed a pipeline system for the task, including question parsing, document feature generation, document event type classifications, document retrieval and document clustering.

2.1 Question Properties

In this task, each question contains three components: the event type and two event properties. The two event properties provided are either the time, the location or the participant of the event. And specifications for these properties can vary in granularity (e.g. day/month/year, city/state, first/last/full name). Details can refer to official task description (Postma et al., 2018). In this task, we consider four event types (i.e. killing, injuring, fire_burning, job_firing). But in training data, only killing and injuring events are provided. Our system first processes each question to extract the question event type and propeties. Later each question and document will be paired (q-d pair) and assigned question properties as binary features. For instance, if a question asks for killing event(s) that happened at specific location and time, the features related to the asked event type and properties (i.e. ask_killing, ask_time, ask_location) will be 1 and others

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 674–678
New Orleans, Louisiana, June 5–6, 2018.

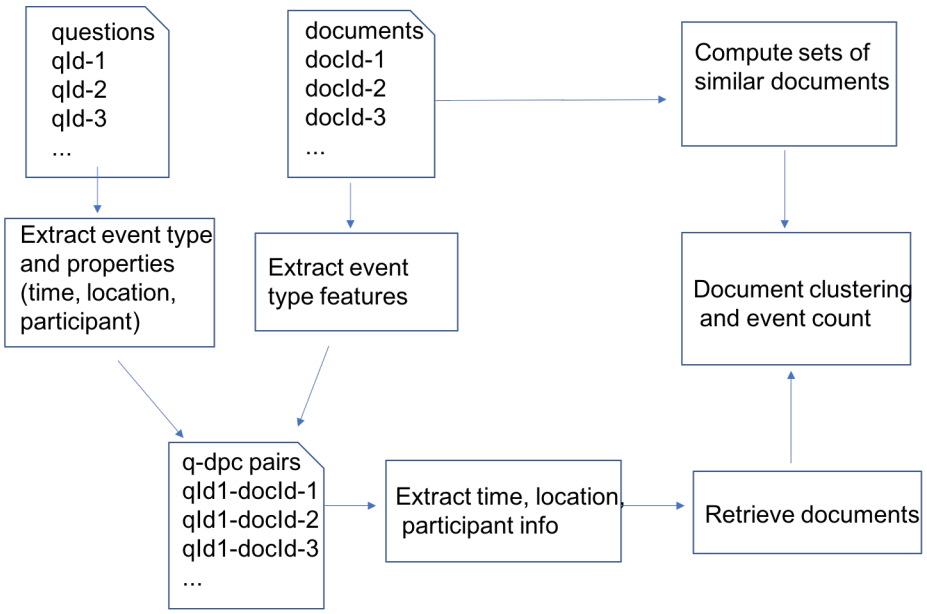

Figure 1: System Overview.

(i.e. ask_injuring, ask_fire_burning, ask_job_firing, ask_participant) will be 0.

2.2 Document Features

2.2.1 Event Type Feature

Event type is a feature which can be defined by the document itself. To decide if a document is one of the four types defined in this task, we ultized both word count feature and classification results. Given the root event trigger keywords for an event type, we first made a synonym word list by including the top similar words from word2vec, based on a cosine similarity score threshold (Mikolov et al., 2013a,b), and adding a couple of missing common words associated with the event. Then we scanned through all documents and count total number of words from the event word list in the document. These counts are then used as word count feature for each event type.

To prepare the training data for the classifier, we selected the killing and injuring documents according to the answer document in trial data. And we used a keyword list to select fire_burning ("firefighter", "fire department", "wildfire", "burn") and job_firing ("employee", "employment", "fired for") related documents. To remove confusing documents that are actually gun violence related, a short list of gun violence keywords ("gun", "shot", "bullet", "shoot") were used when selecting fire_burning and job_firing documents. Two SVM models were trained for event classification. One for classification between gun violence and non-gun violence. Another one is to determine if an event is an injuring event (not killing-only).

2.2.2 Location Feature

If a question asks for a specific location, the system will extract a location feature from each document according to such location. For instance, if a question asks for an event that occurred in Phoenix, Arizona, it will scan each token in the document and decide if "Phoenix" and "Arizona" are in the document. Based on the granularity of the location asked, a score is assigned to the q-d pair as document location feature. As the states in the U.S. are often abbreviated, a dictionary of the U.S. states with their abbreviations [1] is used for better recall. Additionally, a database of cities in states of the U.S. [2] is used to retrieve a document in the case that a state is asked but the document only mentions the city in that state. In the current system, a partial score is assigned to such a document. And one problem raised by this method is the possibility to retrieve the false positive documents which contain the city with the same name that is in a different state. This issue has not been resolved in the current system, although

[1] https://en.wikipedia.org/wiki/List_of_U.S._state_abbreviations

[2] https://github.com/agalea91/city_to_state_dictionary

Feature	Description
DCT_DIFF	document creation time difference, in days
Title_Cosine	cosine similarity of title words
Title_NE_Match	number of common NE in title
Title_NE_Jaccard	Jaccard similarity of NE in title
All_Person_Cosine	cosine similarity of Person entities in the document
All_Person_1gram_Cosine	cosine similarity of 1gram of Person in the document
All_Location_Cosine	cosine similarity of Location in the document
All_Organization_Cosine	cosine similarity of Org in the document
All_NE_Cosine	cosine similarity of all NE in the document
All_NE_Jaccard	jaccard similarity of all NE in the document
All_Cosin_1gram	cosine similarity of 1-gram
All_Cosine_2gram	cosine similarity of 2-gram

Table 1: Features used for clustering documents talking about same events.

	S1	S2	S3
F1	78.33	50.52	63.59

Table 2: Document-Retrieval Performance.

attempts with using additional county information are tried.

2.2.3 Participant Feature

If a question asks for a specific participant name, the system will extract a participant feature based on the match of the name and the granularity of the name that is asked. For example, if a name "Michael Farrow" is asked in the question, it will assign a max score to this participant feature when "Michael Farrow" is found, or partial score when only first or last name is found. To avoid the case that partial name matches to a wrong entity, we used named entity recognition tool from Stanford CoreNLP [3] to extract all named entities, and if the partial name is found in the named entities other than a person, a relatively low score will be assigned as the feature. In addition, in real life a location name (e.g. a street, a mountain or a station etc.) may sometimes be named after a person (e.g. Franklin Street). The system combines these partial names with the common location words and their abbreviations (e.g. street and st.), and use regular expression to check if the documents contain such combination. If so, a relative low score is given.

[3] https://nlp.stanford.edu/software/CRF-NER.html

2.2.4 Time Feature

In case a question asks about a specific time, the system extracts potential times via word level matching and Stanford named entity recognition tool. And all the extracted time phrase are normlized to the format as 'DD/MM/YYYY'. Each document has a creation time, i,e. DCT. Because errors may occur during time extraction, it select a potential time only if it is before the DCT, and then such time is compared with the question's time.

2.3 Document Retrieval

To decide the event type, we considered both event keywords and classification results. For violence events (killing and injuring), whether the document contains killing keywords is the only factor to decide if it is killing event, but for injuring event the classification result is used to decide if it is an injuring incidence. For the other two events, both the keyword count and gun violence classification results are considered. With all the features prepared for each q-d pair, a threshold is set for each feature and only q-d pairs with all the feature scores above their thresholds are considered as one corret q-d pair, which means the document is considered as an correct answer for the corresponding question.

2.4 Document Clustering

In subtask 2, after we identify documents that meet the requirements of the question, we need to group the documents that talk about the same event into the same cluster. We used the features listed in Table 1 to train a classifier, to determine

	Time	Location	Participant
questions	892	734	438
non-recalled correct documents	70	86	265

Table 3: Number of questions asked for each properties and number of correct documents missing the corresponding key information.

if two documents are taking about the same event. It is a binary classifier, with two classes: Yes and No. Based on the classification result, we used 1NN for clustering. The trial data of subtask 1 and 2 are used as the training data to train the classifier. In total, there are 599 positive samples (document pairs that talk about the same event), and we randomly generated 1000 document pairs as the negative samples. The classification algorithm used is Random Forest. Based on 10-fold validation, the F measure of the classifier is 0.96. This classifier was also used in subtask 1 and 3 for expanding the root documents for improving recall.

3 Evaluation

Document retrieval performance for all the three subtasks are shown in Table 2.

To understand the loss of the recall, we counted the numbers of questions that ask for time, participant, and location, respectively, and the numbers of documents our system did not retrieve due to the property (e.g. time, participant, and location) was not found, which is shown in Table 3. We use the following two documents as examples:

1. "Probable cause hearing being held for 3 accused in fatal shooting of 3-year-old DETROIT (AP MODIFIED)- Three men charged in the shooting of a 3-year-old Detroit girl were in court today for a probable cause hearing. The Wayne County prosecutor's office said the three are charged in the death of Makanzee Oldham, who died after she was shot while in a car with her father after a fight erupted and someone poured juice on a woman getting ready for prom. Thirty - year - old Cleveland Smelley is accused of firing the shot that killed Makanzee. He and two other men, Deonta Bennett and Antoine Smelley, are also charged with attempted murder because there were oth-ers in the car ", and

2. "Suspect arrested in Detroit shooting, 2-year-old girl still in critical condition DETROIT (WXYZ)- A 2-year-old girl remains in critical condition after she was shot in the head on Detroit's east side on Wednesday One of the men is the father of the little girl , the other is the suspected shooter , Cleveland Smelley Police say Smelley pulled out a gun and fired one shot at the other man , missing him and hitting the little girl in the head She was sitting in her car seat when she was shot Police confirmed they arrested Cleveland Smelley on Thursday afternoon "

The above two documents are correct answer to this question,*"Which ['killing'] event happened in 05/2016 (month) that involve the name Deonta (first)?"*

Document 1 was created on 06/09/2016, but there is no indication in it showing that the event happened in 05/2016, though it has the correct person name "Deonta". In contrast, document 2 does not have the name "Deonta". Therefore, neither of the two documents were retrieved via our current system. The document clustering method described in Section 2.4 does help in retrieving documents missing certain question properties, but there are still documents we cannot retrieve, especially for questions that we cannot find any document containing both required properties as the root document for document expansion using the clustering method.

As shown in Table 3, missing participants information in the documents is most common. We did not participant in the coreference competition and our system cannot correctly retrieve such documents without the key information that is asked in the question. Recent works on event nuget detection and coreference have proposed nerual networks models (Nguyen et al., 2016a,b; Nguyen and Grishman, 2016; Choubey and Huang, 2017). Those works could be studied in the future, where one can elaborate the coreference information across the document to boost the recall of correct documents without key information.

References

Prafulla Kumar Choubey and Ruihong Huang. 2017. TAMU at KBP 2017: Event nugget detection and coreference resolution. *CoRR*, abs/1711.02162.

Xiaocheng Feng, Lifu Huang, Duyu Tang, Bing Qin, Heng Ji, and Ting Liu. 2016. A language-independent neural network for event detection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, page 66. Association for Computational Linguistics.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *ACL-08: HLT - 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*, pages 254–262.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016a. Joint event extraction via recurrent neural networks. In *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016 - Proceedings of the Conference*, pages 300–309. Association for Computational Linguistics (ACL).

Thien Huu Nguyen and Ralph Grishman. 2016. Modeling skip-grams for event detection with convolutional neural networks. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, page 886891. Association for Computational Linguistics.

Thien Huu Nguyen, Adam Meyers, and Ralph Grishman. 2016b. New york university 2016 system for kbp event nugget: A deep learning approach. In *Proceedings of Ninth Text Analysis Conference*.

Marten Postma, Filip Ilievski, and Piek Vossen. 2018. Semeval-2018 task 5: Counting events and participants in the long tail.