**Summary for an Entity in QA System**

**Research Problem:**

To summarize a text document in accordance with the query specified by the user.

**Motivation:**

Some existing summarisation methods give the same summarisation based on an entity. But in Question-Answer system, the summary for the answer is also related to the query specified by the user.

For example, we have two questions that have the same answer “Berlin”. One question is that “What is the capital of Germany”, while the other is “Where is the Freie Universitat Berlin located in”. The summaries for the answer ‘Berlin’ would be more accurate if one focuses on the demographics and the other focuses on the education.

**Methodology:**

Predicates are used to identify and exploit links among subject and object so that they could facilitate the extraction of relevant text. In our method, we have three main phases to get a specific summary.

* First, we clustered all predicates from DBPedia to select the several main predicates which could briefly express the relation between query and answer.
* Second, we selected sentences regarding to the answer entity based on the main predicate and the question keyword.
* Finally, an (abstractive) summary is generated on the candidate sentences.

Here comes to the details of the above three phases.

**Predicates Preparation:**

***Step 1. Cluster the Predicates***

We used GoogleNews-vectors-negative300 word2vec model to transpose words into a high dimension vector space (300 dimension).  A vector’s position within the high dimensional space gives a good indication of the word’s semantic class. We clustered the vectors into coherent semantic groupings.

There are a variety of methods for clustering vectors, including density-based clustering, hierarchical clustering, and centroid clustering. One of the most intuitive and most commonly used centroid-based methods is K-Means. Given a collection of points in a space, K-Means uses random initialization to pick a few central points, then assigns each of the other points to the centroids to which it’s closest. Using thesis preliminary groupings, the next step is to find the “centroid” of each group, using the same technique to one would use to find the centre of a square. The centroids are updated again. This process continues until the centroids settle down and stop moving, after which the clustering is complete.

By using K-Means, the predicates are clustered into 52 clusters.

***Step 2. Get a word label for each cluster.***

We take the average of each word vector in each cluster, and then get the top 1 similar word with word2vec model. This top 1 similar word is used as a word label for each cluster.

**Candidate sentences selection:**

For each question and corresponding answer, we have three useful words:

***relation word***: expressing the relation between question and answer entity;

***keyword***: the most important word in question;

***answer***: the answer to the question.

With these three words, we could start our candidate sentences selection.

**Step 1. Extract the page related to answer entity on Wikipedia.**

Get the Wikipedia page given the *answer entity*.

Since not every answer will have a Wikipedia page, in this case, we will search the Wikipedia, and get the top 1 page used for this answer.

**Step 2. Get the general summary for the answer entity.**

In most cases, the first paragraph in Wikipedia page is used to provide a general summary based on the search entity. This paragraph will be selected as a part of candidate sentences.

**Step 3. Get the specified sentences for the question.**

After selecting general summary sentences in step 2, we want more candidate sentences related to the focus of the question.

In Wikipedia, after first paragraph stating a general summary, the other parts are composed of some sections describing different aspects of the entity. The structure is shown below:

General summary

--- Section 1

------ Subsection 1.1

------ Subsection 1.2

------ …….

--- Section 2

------ Subsection 2.1

------ Subsection 2.2

------ …….

--- Section 3

------ Subsection 3.1

------ Subsection 3.2

------ …….

………………….

Notes

References

External Links

In this step, the *relation word* will be assigned to one of the 52 predicates cluster word, since using more general predicate would avoid losing information by using a relation word in a specified question.

After selecting the most appropriate predicate cluster word, we use this word to get the most related top section. If this top section does not have any subsections, the sentences below this top section will be selected as specified candidate sentences. If the top section has subsections, we will use the *keyword* to find the most related subsections. Then, the sentences under this subsection will be selected as specified candidate sentences.

**Abstractive Summary:**

This phase is using the candidate sentences to get abstractive summarization.