ELEG/CPEG457

Final Report

Jianbo Pei, Yifeng Liu

Extraction Based Multi News Summarization

Introduction:

There are massive news and articles on the Internet. Most of them are much similar. It is time-consuming to read through every of them. As the name of the project states, the function of this project is to extract high valuable sentences from multiple news and articles to form a concise summary based on a given user query. As a result, it could save so much time for users to capture the important information they are interested in.

In the process of summary generation, the project could calculate sentences scores, compare sentences similarity, and select the sentences that meet the requirements. The first requirement is that the summary must contain the introduction of some news, the impact, and the result. Second requirement is that the length must be within 800 characters.

Although the aiming users of the project is not everyone, anyone who would like to have a general idea or feeling about something could benefit from this project. In other words, for people who would like to briefly and quickly know something instead of deeply looking through details, it could complete such task.

Motivation:

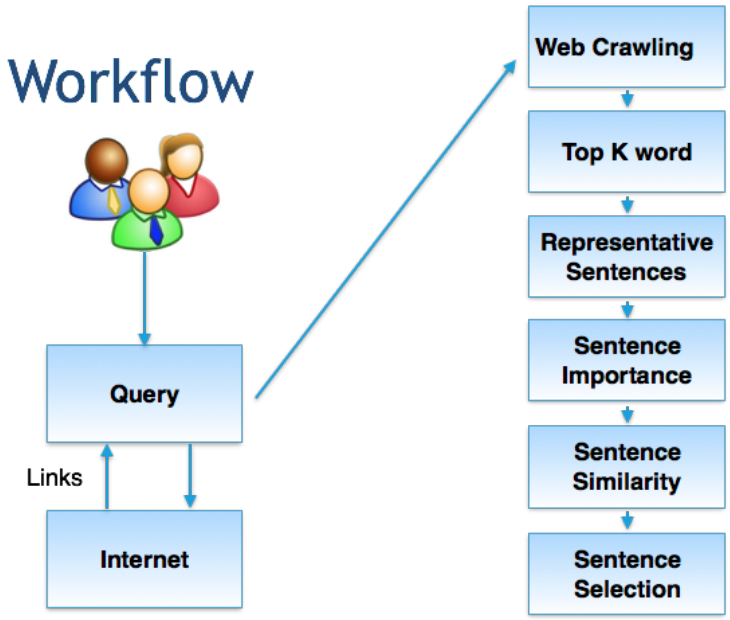
The motivation substantially could be addressed from two aspects. The first is from the course we are taking. In fact, the course has covered most topics and techniques we have

implemented in our project. For example, the technique we are using for sentences comparison is cosine similarity. Therefore, throughout the construction of our project, we could practice what we have learned in class.

Second motivation is from people needs. As mentioned above, the aiming users are those who would like to briefly and quickly know something. In fact, there are so many existing tools or application can do some summarization. The tools can be divided into two categories. The first is news application or websites. For example, the app “News” on iPhone can display brief introduction for the news. The second is online summarization tool. For example, the website called “Tools4noobs. The website only can summarize one single document. So, what is the point of our project? Clearly, those tools cannot cover all user needs. For users who want a decent summary from multiple articles, neither of these can meet their requirements. Therefore, after doing some research, we decided to design a tool that can summarize multiple articles.

System description:

---Workflow



Our project is based on the diagram above. The basic process is starting from user query. After users typed their queries, the system will do the web crawling and capture the useful urls. Then, based on the queries, it will generate a list of top K words. Next, it will find representative sentences with top K words from all articles or news. Next, the system will compute the sentence score from selected sentences. After that, the system will do a similarity comparison with scored sentences and select sentences. Finally, the summary will be generated with certain requirements.

---Approaches:

For developing platform, we used Python 3.4.4 version as our platform to develop the project. The reason is that this software system is easier to implement and can be installed in lots of useful tool packages.

Before midterm, we designed the prototype of the project. For web crawling, we used Google search API which can be easily installed in Python to obtain the urls. Then, we used the BeautifulSoup toolkit to capture the content from obtained ursl. However, the problem is that the content contains some annoying noise. As a result, the noise might have negative impact on the final result. To filter the noise, we used top k word method. The top k words are from titles that are related to user query. The result ends up with less noise than before. For representative sentences, we selected the first ten sentences from each news or article and used NLTK to find the sentences that have future tense. Because in most news or articles, the first ten sentences often contain much more important information. And the sentences with future tense can show the impact or result of the news. To score or assign weight to the selected sentences, we used TF-IDF method. Because we found that Scikit-learning library already has the TF-IDF package, to easily implement, we imported the TF-IDF package directly from Scikit-learning. Move on to

sentence similarity, we used cosine similarity, which can compare two non-zero vectors with cosine of angle between them. With final process sentence selection, we set a length limitation for the summary. For the prototype we set it to 400 characters. However, the selected sentences might not be the most informative ones.

After midterm, we kept the methods for web crawling, computation of sentence importance scores, and finding representative sentences, but we improved our project in the sections of top K word list, sentence similarity, and sentence selection. The top K word list before was created from titles, so we extended the list by adding related words, which are synonyms of words. The tool we used is NLTK wordnet. For sentence similarity, we used another technique called TF-IDF similarity instead of cosine similarity. For the final sentence selection, we used a method called Maximal Marginal Relevance(MMR). The improvement here is that using MMR can select the most informative sentences and reduce redundancy. However, this method does have some limitation, which is that the optimal selection for summary might not be that effective.

Related work:

For web crawling, we did some research online using Google. We found the useful tool to do the web crawling are Google search API and BeatifulSoup. The Google search API is not hard to implement. However, implementing BeatifulSoup needs some research. After looking up some websites and watching some tutorial videos, we successfully implemented the BeatifulSoup toolkit and obtained content from urls. However, there was some noise in the content. Then, we asked professor that if there is a way to filter all noise. Unfortunately, the noise couldn't be fully removed. Nevertheless, top K word method could filter most of the noise.

For other sections of the project, there are not direct resources that we can use. Most of them are articles that describe the concepts and theories. Therefore, we needed to understand the

concepts first and then wrote code. Except our own research, we also have received some great advice and ideas from professor. Here are the research articles we have read to help develop our project:

1) “A Survey of Text Summarization Techniques” by Ani Nenkova and Kathleen Mckeown.
2) “An Exploration of Proximity Measures in Information Retrieval” by Tao Tao and ChengXiang Zhai
3) “Improving Summarization Performance by Sentence Compression - A Pilot Study” by Chin-Yew Lin
4) “Multi-document Extraction Based Summarization” by Sandeep Sripada, Venu Gopal Kasturi, and Gautam Kumar Parai
5) “Assessing Sentence Scoring Techniques for Extractive Text Summarization” by Rafael Ferreira, Luciano de Souza Cabral, Rafael Dueire Lins, Gabriel Pereira e Silva, Fred Freitas, George D.C. Cavalcanti, Rinaldo Lima, Steven J. Simske, Luciano Favaro
6) “Sentence Similarity Based on Semantic Nets and Corpus Statistics” by Yuhua Li, David McLean, Zuhair A. Bandar, James D. O’Shea, and Keeley Crockett
7) “Text Summarization in Data Mining” by Colleen E. Crangle

Evaluation:

In fact, after we improved the system with new techniques, the result did not change much. It might be the limitation from the method extraction summarization. We could not find similar system using abstraction summarization to compare with our system. But, the result looks nice to us. The summary contains all the requirements we set. We were planning to compare our result with other similar systems. However, we could not find online. Most systems are used to

summarize single document. Therefore, we asked our friends to read the summary and evaluate it. We asked three friends and theirs responses to the result are optimistic.

Example:

Query: wanna cry computer virus

Summary:

Wanna Decryptor, also known as WannaCry or wcry, is a specific ransomware program that locks all the data on a computer system and leaves the user with only two files: instructions on what to do next and the Wanna Decryptor program itself.

How to respond to a Wanna Cry Ransomware Attack?Disconnect your device from the internet to ensure there is no further infection or exfiltrating of data as the ransomware will be unable to reach the command and control servers.

As Bloomberg reports that Matt Suiche, founder of United Arab Emirates-based cyber security firm Comae Technologies warns a new version of the ransomware may have also been spreading over the weekend.

You should only rely on professional ways to remove WannaCry virus and not try to uninstall this malicious program manually.

Although it may seem like a small amount to charge, the ransomware attacks are often widely distributed, so the ransom payments can stack up.

"There is a component of the ransomware that spreads laterally, unconfirmed reports suggest this could potentially be via SMB shares or leveraging a recent Microsoft bug to spread," Travis Farral, Director of Security Strategy for Anomali, told Mirror Tech.

"Bossert tells ABC's 'Good Morning America' that the malware is an "extremely serious threat" that could inspire copycat attacks.

Conclusions:

The result from our system looks fine. However, there are some other work that we could do in the future to make the system work better. First, we could improve our implementation to make our program faster. Second, since the selected sentences are individually from each news, there are not transition words between each sentence. The summary may be very stiff. Therefore, adding the transition words in summary will make it more readable and smooth. Last but not least, the system is currently running in a terminal emulator. In the future, we could build a GUI or Web application.

Discussions:

From developing the system, we learned a lot new things. Also, we practiced with things we learned from class. Although we have encountered some problems, we tried different techniques to find the solution. Of course, we have some challenges as well. First again is understanding some sophisticated concepts from papers. Second is that sometime we could not successfully to implement some difficult methods, we need to find some alternative methods. For example, we tried to use the combination of semantic and word order method in sentence similarity. However, it is much complicated than we expected. Therefore, we ended up with TF-IDF similarity method.

Work Division:

---System

Before the midterm, Jianbo mainly contributed to the web crawling, generating top K word, and computing sentence score. Yifeng mainly contributed to finding the representative sentences, sentences similarity comparison, and sentence selection.

After the midterm, since some articles that introduce new techniques are much more complicated, we read together and help each other to understand the concepts. If needs to be more specific, Jianbo mainly contributed to extending top K word. Yifeng mainly contributed to sentences similarity comparison and sentence selection.

---Report

Jianbo is mainly in charge of Introduction, Motivation, Evaluation, and Example.

Yifeng is mainly in charge of Related work, Conclusion, and Discussions.

We did the System description part together.

Also, we reviewed each other's writing and revised together.