André Schweizer
Prof. Diego Klabjan
IEMS 308 – Data Science & Analytics
March 13th, 2019

HOMEWORK ASSIGNMENT IV
Q&A Assignment System

INSTRUCTIONS

1. Create a text file with 8 questions according to the example below (example file can be found as *Questions.txt* in the Github directory):

    Which companies went bankrupt in December 2017?
    Which companies went bankrupt in July 2009?
    Which companies went bankrupt in August 2011?

    What affects GDP?
    What percentage of drop or increase is associated with this property?

    Who is the CEO of Apple?
    Who is the CEO of Berkshire Hathaway?
    Who is the CEO of Amazon?

    It's important to note a couple of things:
    - The user may choose to skip as many lines as s/he wants between the questions or before the first one – blank spaces will be ignored by the algorithm.
    - The questions must be in order – companies going bankrupt followed by GDP factors and CEOs of companies.
    - Although there is only a pair of questions related to factors that influence GDP, the algorithm will repeat that question twice more, returning a total of three GDP factors and their associated percentages.
    - The user should always enter 8 questions, respecting the numbers for each of the categories (i.e. 3 questions about companies going bankrupt, 2 questions about GDP factors, and 3 questions about CEOs of companies).

2. Download and open the *.ipynb* file using the Jupyter notebook, available in Anaconda.
3. Change the directory input to the *os.chdir* function in the second cell to be the path of the folder where you saved the text file with the questions.
4. Change the directory input to the *glob.glob* function in the first cell of the *Import corpus* section to be the path of the folder where you have the corpus saved.
5. Run each of the cells until the end of the notebook.
6. The output file will be exported to the directory in which the Jupyter notebook was saved with the name *SampleOutput.txt.*

EXECUTIVE SUMMARY

The Q&A system developed finds answers to the questions provided in a text file with the format described in the previous section. It finds the keywords in the questions, and takes those, together with the documents available in the given corpus, to compute tf-idf scores for each keyword-document combination. In the most relevant documents for each question, it searches for the best sentences by looking at which of them contain the most number of keywords. Finally, it extracts from the best sentences the answers to each of the questions. It is important to note that difficulties were found in this last step, prompting adjustments to be made to the end of the algorithm that potentially "over fit" it to the questions and corpus being used – the level of accuracy of the algorithm will likely decrease if it is prompted questions in other formats.

It is interesting to note that the sample results obtained for the company-related questions have a low degree of accuracy in comparison to the other types of questions. That is due to the fact that answers to the GDP factors and CEO questions can usually be found within the same sentence as the keywords. For example, "Tim Cook" and "Apple" are very likely mentioned in the same sentence a number of times throughout the corpus – same, although to a lower degree, for the GDP factors and their respective percentages. That isn't true for the companies that go bankrupt in a particular time period (i.e. a sentence that contains a company name, the keyword "bankrupt", and the date of the happening is a very specific instance, making it very rare). Hence, a much more complex algorithm would be needed to si6gnificantly increase the accuracy of the answers to those questions. It would entail a much deeper analysis of the context of each of the documents, as well as the consideration of a range of complicated financial terms related to bankruptcy.

Another noteworthy finding is the impact of tf-idf scores on the efficiency of the algorithm. As can be seen on slide 2 of the attached *Findings* file, the relevance of documents to the questions seems to follow a power law (i.e. a particular score is inversely proportional to the number of documents that have that score). Hence, the tf-idf algorithm allows us to only look at that small group of highly relevant documents, greatly cutting down the computational effort required by the Q&A system.

Similarly, by selecting only the sentences that have compatible tags to the questions, the NER tagger decreased the number of sentences in which keywords are searched by 92.6% (from 882,980 to 65,421). Then, by selecting only sentences that contain the question keywords, the algorithm was further able to reduce the search space to only 4,265 relevant sentences (a 99.5% total reduction from the initial corpus).

Lastly, it is important to note that Q&A systems have an enormous business potential. First, the market for Intelligent Virtual Assistants, which are heavily dependent on systems like the one developed, is quickly growing at 32% CAGR, resulting in a 429% expected aggregate growth between 2017 and 2023. One of the main reasons for that is the time savings these systems brings to companies that adopt them. The system developed, for instance only takes about 1h32min to look for answers in the provided corpus, a process that would take an average person about 50 days – 99.9% savings in time.