Question Answering System: Business Insider

IEMS 308 - Data Science and Analytics, Homework 4: QA System

Saif Bhatti

March 20, 2020

1 Executive Summary

Business Insider provides important second-hand news stories regarding the latest in innovation and progress in the business world. An important task for BI might be using its extensive corpus of business related content to answer questions. This leads to the formation of a Question-Answer system which is able to parse select types of queries and respond by collecting results across the corpus of text.

- After intaking all of the 2013 and 2014 articles as the corpus, the text data is organised and indexed with Elastic-Search - a distributed search and analytics engine. At this point, the documents are split into constituent sentences, and keywords from a user question are used to find documents that contained relevant information. These documents are then parsed into sentences, and the results are extracted using regex. At this point, the answer is printed to the screen, and the user is prompted to ask a new question, a follow up question, or to type 'Stop' to exit the program.
- The Question-Answer system was successful in that it provides the output for the questions asked in a reasonable time frame. However, there are numerous areas for improvement. Notably, text retrieval is constrained to the corpus available and reflects information that is accurate to that time period, which results in answers that may be outdated. For instance, the CEO of IBM stepped down in 2020, and as such the question "Who is the CEO of IBM" returns Ginni Rometty, despite the fact that she is no longer CEO.

2 Problem Statement

The Business Insider articles were .txt files, each containing a full article, extracted from their API for the years 2013 and 2014. There was a little less than one article per day, for a total of 730 articles (this will be henceforth referred to as the 'corpus' and the required tasks were as follows:

- Question Type 1: Who is the CEO of company X?
- Question Type 2: Which companies went bankrupt in month X of year Y?
- Question Type 3: What affects GDP? What percentage of drop or increase is associated with this property?

Each of these questions required a slightly different pipeline to be able to answer, and the qa_nofluff.ipynb contains all three pipelines taken to their respective answers. As a result of these nuanced differences, the codebase ended up being significantly less tidy than it might have been.

3 Data Methodology

Developing a question and answer system capable of answering the following questions using Elastic Search. The corpus used Business Insider articles from 2013 and 2014, and as a direct result, all answers will be from this time period.

3.1 Elastic Search

This project would be near impossible without the flexibility and speed offered by the Elastic Search Engine. The ES Engine allows for distributed, No-SQL style searching across the entirety of the corpus, and runs swiftly across the 730 files indexed.

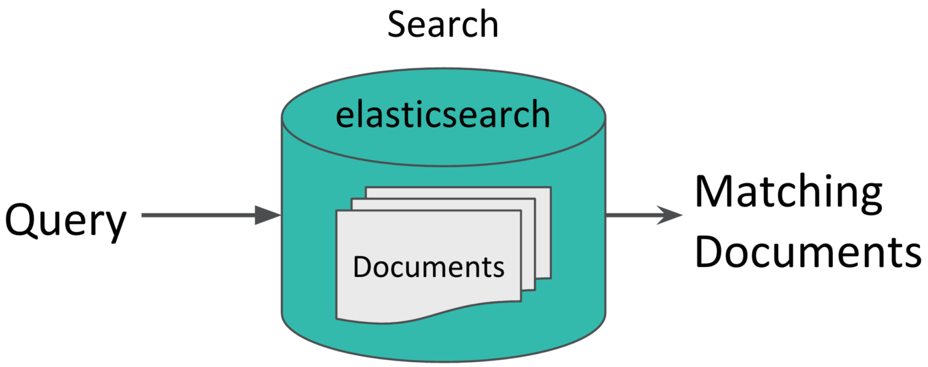


Figure 1: Visualising Searching with Elastic Search

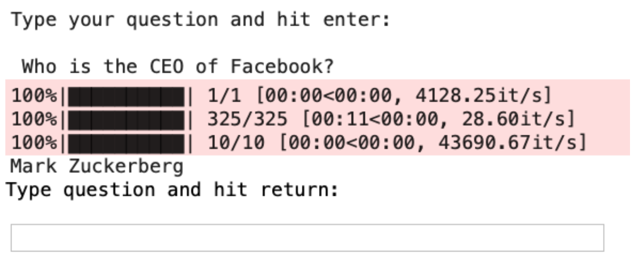
3.2 General Framework

For each question type, execute the following:

- Index articles.
- Perform sentence segmentation on the corpus.
- Separate article into sentences.
- Narrow the search for answer to sentence level.
- Tokenise the question, find keywords.
  - Based on question classification, find articles, select highest scoring documents.
  - Index sentences.
  - Search on sentences using keywords, select highest scoring sentences.
  - Use regex to return answer from highest ranking sentences.

4 Results

4.1 Question Type 1 - CEO

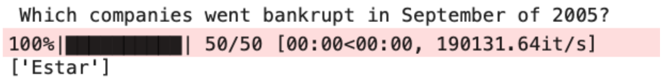


```
Type your question and hit enter:

 Who is the CEO of Facebook?
100%|██████████| 1/1 [00:00<00:00, 4128.25it/s]
100%|██████████| 325/325 [00:11<00:00, 28.60it/s]
100%|██████████| 10/10 [00:00<00:00, 43690.67it/s]
Mark Zuckerberg
Type question and hit return:
```

Figure 2: Example of CEO question

4.2 Question Type 2 - Bankruptcy



```
 Which companies went bankrupt in September of 2005?
100%|██████████| 50/50 [00:00<00:00, 190131.64it/s]
['Estar']
```

Figure 3: Example of Company Bankruptcy question

4.3 Question Type 3 - GDP Affectations

- Please run code to see output.

5 Conclusions

- For all three questions types, the Question-Answer system is able to provide an answer based on information found in the corpus.

- The program conclusively works. This is perhaps the best thing that can be said about it; in a lot of ways (speed, code base) there are improvements to be made. Details on these are available in the Next Steps section. Implementing these details, as well as improving and expanding the corpus that fed into the index will result in a more robust and capable engine.

6 Next Steps

- **Indexing Speed Optimisation**: There are perhaps better ways to use the Elastic-Search Engine than I found close to the end of this project, and did not have time to implement. For instance, the current approach still uses an 'offline' version of the corpus to find the , and does not fully incorporate the results from the ES Engine. Incorporating the results will speed up both startup times (since the corpus information does not need to be index offline), and in subsequent search result analysis (since the online results can be estimated outright).

- **Improving code base**: The main areas of improvements include using Python's classes and writing cleaner, reusable code. The current format resulted in substantial debugging and redundancy errors. Using functions to clean this up would be a major boon. Combining this approach with spaCy 's NER capabilities might also be productive, but there are still issues with this. For instance, during the EDA phase, when I iterated over a list of company names to understand the viability of 'ORG' entity recognition in parsing a user question, the following situation occurred:

  - 'Netflix' was found to either be unrecognised as an entity, or classed as a 'PERSON'.
  - 'Netflix, Inc.' was found to be classified correctly as an 'ORG'.

- However, within the remits of natural language processing engine, this does not do a good job of capturing the natural phrasing that users will likely use (dropping all company-entity jargon, such as Inc. and LLC, etc). As a result, I decided to just stick with Elastic Search and basic regex to extract answers, which also worked quite well.

- **Parallel Computing**: With the advancement of computing, it is possible to split processing effort across multiple cores on a machine, and as a result this speeds up runtime. Short of buying compute time, using cuda and Pytorch would speed up the 'offline' indexing time significantly, to cut overall answer lead-time down significantly.

7 Appendix

1. Data can be found at these hyperlinks: Business Insider, 2013, Business Insider, 2014, Business Insider, Trained Test.

2. Source code can be found at this hyperlink: saif1457-github