Building (a part of) Watson

Mark Hadley
CS 483
Spring 2017
https://github.com/hadleym/Watson

Indexing and Retrieval

There were three models for indexing the Wikipedia documents. 1) Utilizing NLP Core by preprocessing the raw Wikipedia files, referred to as the 'NLP Core Model'. 2) The Lucene English Analyzer and 3) The Lucene Standard Analyzer. All three models represent all of the terms in a Wikipedia article as bag of words, disregarding position of the terms within the text and queries.

NLP Core Preprocessing

For the NLP Core Model, the raw files were lemmatized using the Stanford NLP Core API. These lemmas were stemmed with the API and then filtered to only allow adjectives, nouns, adverbs, and verbs [Appendix 1] to pass through. Once this parts of speech filtering was completed, the sentence was filtered again with a set of manually created stop words [Appendix 2]. These filtered sentences were then outputted to a newly created preprocessed directory. This lengthy processes only needed to occur once.

Stemming and Lemmatization

The English Analyzer allowed for stemming using a PorterStemmer algorithm. The English Analyzer Model did not receive any lemmatization and the NLP Core Model did not receive any stemming. The Standard Analyzer model had neither lemmatization nor stemming on any of its queries or indexing, to allow a base of comparison between the techniques.

Issues with Wikipedia Pages

There were certain lines contained within the corpus that could be identified and ignored. Lines within the Wikipedia articles that began with '==' indicated a subject heading within the Wikipedia page. These were ignored because typically they held no relevance to the document, but were instead abstract references to the text below it, which beyond the scope of these models to interpret.

Lines that began with the substring '#RED' indicated that the line was marking a redirection topic and the text contained within that line was never relevant to the current Wikipedia article. Both of these issues could be addressed in the preprocessing phase of the NLP Core branch.

URLs and image references were not handled in any of the indexes, and caused unrelated text to inflate the size of certain articles that contained them, as the actual text of the URL are not required to contain relevant words that define the material they link to. This caused some articles to become inflated with unrelated text.

Indexing

Each article became its own document within the index, and the index of each model contained two fields. 'Content' fields contained the articles text while the 'Category' field contained the Wikipedia article's name. The Category field was returned as the answer to a Jeopardy question, based on the text contained within the indexed Content field.

In all three of the models the title of the Wikipedia page was added to the 'Content' field of the indexer to allow it to become part of the searchable text. This was done based on the assumption that the title of the Wikipedia article is relatively unique to all of the Wikipedia articles and titles, and therefore adding it to the article it would increase the relevance between the text of the article and the title of the article.

Query Building

For the NLP Core Branch, all queries were preprocessed using the same parts of speech filters and stop word filters as the preprocessed index. Then the WhitespaceAnalyzer was used to parse the query.

For the Standard Analyzer Model and English Analyzer Model, queries to the system needed to have special characters removed [Appendix 3] as these cause the QueryParser to throw ParseExceptions, some of these special characters are interpreted as regular expressions. After the special characters were removed, the query is filtered through the Lucene StandardAnalyzer.

Issues with Jeopardy Questions

In both branches, the Jeopardy Category was added to the original Jeopardy Question to form the query. An exception to this was the Category 'POTPOURRI', as this is a game-specific category that indicates no category. Some of the questions have the Host's name (Alex) as a prompt for understanding the question.

Measuring Performance

Two measures were used: Precision @ 1 and MRR. Precision @ 1 is the main measurement of performance since the game design of Jeopardy is purely based off of a single answer, so having a high P@1 score is a direct correlation to performance in this game.

MRR is a secondary measure of performance, which can measure smaller improvements of a system better than the binary P@1 score. MRR was calculated observing the top 10 results and answers to the questions that were beyond the top 10 of the results were counted as 0 score. Results of these scores are addressed later.

Changing the Scoring Function

Two different scoring functions were used: Classic Similarity (tf-idf) and BM25 Similarity. Tf-idf is convenient for finding relative documents given the text based purely off of the frequency of the terms in the article offset by the frequency of the terms in the entire corpus. Scores calculated using tf-idf can easily be ranked relative to the same query, resulting in a 'best-match' having the highest tf-idf score.

BM25 is a probabilistic ranking function that also can relatively rank documents given a query, with higher scores indicating a more relevant document.

Results

	NLP Core Branch (lemmatization)	English Analyzer (Stemming)	Standard Analyzer
Lucene Classic (tf-idf) Similarity, P@1 Score	19 (19% accuracy)	20(20% accuracy)	18 (18% accuracy)
Lucene BM25 Similarity, P@1 Score	14 (14% accuracy)	18(18% accuracy)	14 (14% accuracy)
Lucene Classic (tf-idf) Similarity, MRR Score	26.263	28.592	28.111
Lucene BM25 Similarity, MRR Score	18.478	24.486	20.251
100 Total Questions	*Lemmatized, Parts of Speech Filtered, Stop word Filtered, Lucene Whitespace Analyzer (no stemming).*	*Lucene EnglishAnalyzer, Stop Word filtered and PorterStemmer (no lemmatization)*	*Lucene StandardAnalyzer, Stop word filtered*

This data can be viewed in the 6 files contained in the project:

CoreNLPLemmasandwhitespaceAnalyzerandBM25.txt
CoreNLPLemmasandwhitespaceAnalyzerandtf-idf.txt
LuceneEnglishAnalyzerandBM25.txt
LuceneEnglishAnalyzerandtf-idf.txt
LuceneStandardAnalyzerandBM25.txt
LuceneStandardAnalyzerandtf-idf.txt

Discussion

The preprocessed NLP Core Model with tf-idf weighting performed with 19% accuracy, and P@1 is the best measure for evaluating these models since the Jeopardy game only allows for contestants to give a single response as their answer. The Standard Analyzer Model had only a slight dip in accuracy (18%) for the same weighting formula. When the BM25 formula was applied there was a 4% point dip in both branches. However, the best performance was from the English Analyzer (stemming) with a 20% accuracy.

What is remarkable, is that the Standard Analyzer Model, which was the simplest model having no stemming, no lemmatization, and only stop word removal, performed at a P@1 score of 18% accuracy. This is impressive because shows how well a very low-level information retrieval system can perform, even with a relatively complex task of answering Jeopardy questions. It also demonstrates that with the basic indexing and querying tools provide sufficient groundwork for this type of system to perform at what I would call a 'decent' level, and can give insight into where improvements can be made (Language Models, biword indexing). The 2% improvement from no stemming to stemming represents how effective a basic tf-idf scoring method is.

When evaluating MRR scoring, only the top 10 hits from the query were evaluated. When the search was expanded to top 50 hits, each MRR score only improved by ~.25%, but doubled the amount of processing time.

Error Analysis

Only 8 out of 100 questions were answered correctly for all 6 variations [Appendix 4]. These 8 questions do not require semantic interpretation and were simply using the search terms to determine the document's relevance.

Semantic Errors

Many of the Jeopardy questions involve understanding how the parts of speech of a question form some sort of semantic meaning that must be interpreted to derive the correct answer. This system is far too simple, especially because of the bag-of-words modelling, to achieve the correct answer. These types of semantic questions failed to have correct results in the top 10.

Many of the questions that the system failed to identify involved the use of language that the models could not parse correctly. For example the term “Capital” needs to be inferred as ‘Washington’ or a two-step interpretation of “veep” be interpreted as “Vice President (of the United States)”.

Errors from Inflation

URLs and image references embedded within the Wikipedia articles were not removed. They inflated documents with unique terms. These terms are not necessarily relevant to the website they link to, or the image they represent. This means that URLs and image links create unique, but unrelated terms within the document, and overall lower the ability of the tf-idf models to identify relevant terms and relevant documents.

Errors from the Category Hint

Some false positives occurred when the category of a question changes the meaning of a question, or again, requires a semantic interpretation. Examples include where the category gives near explicit directions to apply to the question given. “Parent Company: post it notes”, which is also beyond the scope of pure information retrieval. This example requires the system to understand that a product has ownership by another entity.

Errors from Question Format

Some of the questions have special ‘hints’ that allow human players to interpret what is being asked, and even had the host’s name (Alex) within the question. None of the models were able to answer any questions in this format correctly, as the extra instructions were not terms that were relevant to the search, and therefore lowered precision.

Other shortcomings of the system came about from adding the Category to the query. The category would not prove to be relevant to the other terms of the query, and therefore would only reduce the importance of the terms within the query. However, performance was generally better when the Category was added to the query and offset the number of questions missed because of its addition..

Stemming vs. Lemmatization

Both stemming (from the EnglishAnalyzer) and lemmatization (removing parts of speech with Core NLP) proved to be better than doing neither. The optimal configuration was achieved with a 20% accuracy using stemming with tf-idf document weighting.

When stemming is used, unique terms gain greater weight because stemming reduces the size of the vocabulary. These models are leveraging the tf-idf scoring method by identifying the unique terms in a question and in relevant documents. Conversely, when stemming occurs,

common words are stemmed and grouped to form even larger groups of common terms, and the systems can place less importance on even more terms than before stemming. Both sides of stemming positively affect how tf-idf weights queries to documents.

Lemmatizing and removing the parts of speech, as done with the NLP Core model, does not reduces the size of the vocabulary as much as the Porter Stemmer of the English Analyzer Model. And because these models are simple and not able to interpret semantic meaning, even by cleaning up the semantics of a question, they still are not as effective as stemming.

Please see the attached Maven Project containing source code.
All files (including this report) can be viewed on github and will be made a public repository after the due date passes:
https://github.com/hadleym/Watson

Appendix

1) Parts of Speech Tags from NLPCore
    a) Adjectives: JJ, JJR, JJS
    b) Nouns: NN, NNS, NNP, NNPS
    c) Adverbs: RB, RBR, RBS
    d) Verbs: VB, VBD, VBG, VBN, VBP, VBZ
2) Manual Stop Word List:
    a, an, and, are, as, at, be, but, by,for,
    if, in, into, is, it,no, not, of, on, or, such,
    that, the, their, then, there, these,they,
    this, to, was, will, with
3) Special Characters
    a) ()!#&\"\'-
4) 8 Correct Answers for all 6 variations of the models:
    a) Category: SERVICE ORGANIZATIONS, Question: Father Michael McGivney founded this fraternal society for Catholic laymen in 1882
    b) Category: THE RESIDENTS, Question: Title residence of Otter, Flounder, Pinto & Bluto in a 1978 comedy
    c) Category: UCLA CELEBRITY ALUMNI, Question: Neurobiologist Amy Farrah Fowler on "The Big Bang Theory", in real life she has a Ph.D. in neuroscience from UCLA
    d) Category: GREEK FOOD & DRINK, Question: Because it's cured & stored in brine, this crumbly white cheese made from sheep's milk is often referred to as pickled cheese
    e) Category: 1920s NEWS FLASH!, Question: 1927! Gene Tunney takes a long count in the squared circle but rises to defeat this "Manassa Mauler"! Howzabout that!
    f) Category: RANKS & TITLES, Question: Italian for "leader", it was especially applied to Benito Mussolini

g) Category: THE RESIDENTS, Question: Kinch, Carter & LeBeau were all residents of Stalag 13 on this TV show
h) Category: GREEK FOOD & DRINK, Question: The name of this dish of marinated lamb, skewered & grilled, comes from the Greek for "skewer" & also starts with "s"