CS47300 Fall 2020 Assignment 6

Due 1 December 2020 11:59pmEST

Note: Late submissions only accepted until 4 December 11:59pmEST

This will be the final regular assignment for CS47300. We view it as a one week assignment, so even with the break (and possibly moving) there should be plenty of time to get it done. We still have a couple of questions to add (some will involve material not covered yet), but we wanted you to be able to get started; remaining questions will be out by Friday evening. If you start now, you should be able to get it done before Thanksgiving.

While late work will be accepted until the last day of class, we can't promise to get things graded before the Final if they aren't turned in by the due date.

1 Search Engine Optimization

Google recommends having a single URL refer to each page. Is this for their benefit, or for yours (as a web site designer trying to get your page seen)? It would make sense that having more ways to refer to the page would make it easier to find (just like products may have many different sizes to give them more shelf space in the supermarket.)

Given what you know, analyze this question. Assume that you can have either a single url for your page (e.g., `https://www.cs.purdue.edu/~clifton/cs47300/assign6.pdf`), or two (the above, and `https://www.cs.purdue.edu/~clifton/cs473/assign6.pdf`). Furthermore, assume you have defeated any duplicate detection, so the search engine will separately index both pages if you have two URLs.

Analyze if you would expect your search result ranking to be higher or lower with two URLs pointing to the page. Do this for each of:

1. A content-based ad-hoc retrieval approach (e.g., the vector-space model),
2. A network structure based model (e.g., PageRank), and
3. A collaborative filtering approach.

You should be able to come to a conclusion for each, based on a mathematical analysis, if you are better off having one URL or two.

2 Metric Limitations

We often talk of improving information retrieval with respect to some metric (recall, precision, mean average precision, F1 score, false positives / false negatives, etc.) However, there may be different ways to achieve a high score that can lead to some serious ethical issues.

For example, assume we have a "fake news detector" that has a 20% false positive rate, in other words, it identifies 20% of true articles as fake. This might seem high, but if we had an influx of, say, spam messages on facebook it might seem worthwhile if this is what it took to get the amount of spam shown (false negatives) to reasonable levels.

Imagine two different scenarios. In one, the false positives are randomly distributed across all types of true articles. In the second, all true articles critical of the ruling party are falsely deemed to be fake news. I think most would consider the second situation to be much worse that the first.

1. Which do you consider more likely, randomly distributed false positives, or false positives that fall on a particular topic/viewpoint? Hint: Think of this as a text categorization problem, and frame your answer in terms of the way a text categorization method would work.

2. Come up with another situation, where two different outcomes might have the same result with respect to some metric, but ethically there is a significant difference in which seems acceptable and which does not. This should be a situation other than fake news detection. Give a specific example, stating what you are using to measure "success" and two different ways that achieve the same score where you would consider one acceptable and the other unacceptable.

3. See if you can come up with a different situation that is a different type of information retrieval task (e.g., if in 2 you used a collaborative filtering example, perhaps use ad-hoc retrieval here.) You don't need to give much detail for this, just the idea.

3 Sentiment Analysis

We have the set of opinion words and sentiments: "must-have" (+1), "terrible" (-1), "amazing" (+1), "awful" (-1), "small" (0)

Given the example sentence: "These headphones sound amazing."

1. Give a sentiment estimate of the first sentence using aggregate opinion and the provided opinion words.

Now, given the sentence: "This premium handkerchief, this $10,000 used napkin, is a must-have."

2. Give a sentiment estimate of the second sentence using aggregate opinion and the provided opinion words.

3. Does the second statement's contents seem to match the sentiment assigned by the aggregate opinion?

Finally, if we take the sentence "The new, glorious $20,000 toothbrush is a must-have." and know that it is labeled with an overall negative sentiment:

4. What does that tell us about our set of opinion words?

4 Evaluating Non-linear Results

We often assume that the user examines search results from top to bottom and in sequential order. However, some applications give results in a grid view (this is common in product search, e.g., hotels, or in searches where the result is shown as an image rather than text.) Users might go either left to right (column-wise) or top to bottom (row-wise) in perusing the results. Design a rank-based evaluation metric for such a search result. You can modify an existing metric such as MAP or propose a new one.

5 Question Answering and NLP

Given the query "When is the CS47300 final exam?", run through steps a question-answering system would use to answer this as a question (rather than just returning a web page such as `https://www.cs.purdue.edu/~clifton/cs47300/` or `https://roomschedule.mypurdue.purdue.edu/Timetabling/gwt.jsp?page=exams`). For example, NLP tasks in parsing the question, search for pages containing the answer, etc. Most of this follows from the last 10 mintes of the lecture on question answering, but you are welcome to use methods from other sources (textbook, NLP lecture, etc.)

1. What is the type of query? (Informational, Navigational, Transactional). Explain briefly how this might be determined for this specific query.
2. How might you identify appropriate sources for the information? Assume there isn't a predefined structured knowledge base, but you have to use web search.
3. How might you go from a web page to identify specifically what in that web page answers the question?
4. How might you go from a fragment of text (say, a line or sentence containing the answer) to something that actually sounds like an answer? Hint: If you are more specific in your answer to part 1 than just informational, transactional, or navigational (which should be easy to figure out from word in this query), you can determine a ¡q¿template¡/q¿ for what an answer would look like.

You should be able to answer each of these in a few sentences.

6 Big Data methods

Sketch out a means to use a Map/Reduce style system to build an inverted index supporting TF/IDF queries. In other words, you would like to produce a term, document frequency, and list of documents with their term frequencies.

A simple option would be to view this as producing two indexes, one with term/document/term frequency, one with term/document frequency. A better option would be to produce something that combines all the information. Don't worry about producing a "file" - simply collecting and emitting the appropriate sums is fine.

You may use the pseudo-code syntax from the 11/18 lecture, or spark syntax (java or python) from 11/20. You do not need to produce something even remotely executable, the goal is to capture the basic logic used in a map-reduce framework.

If you do feel like creating something executable, we do have a spark cluster you can try it on. We did a project on this last year, and it is still available:
`https://www.cs.purdue.edu/homes/clifton/cs47300/Project3.sxhtml`
While this isn't the same task, it does give a simple example program in both java and python. A caveat - this is not a production environment. In particular, I've see situations where students with a lot of python cruft (unwanted software) from prior courses that was auto-installed by scripts run as part of those courses can result in simple programs (even the samples provided) failing, and sometimes spawning multiple processes and bringing the whole system down. So feel free to try it, but if something doesn't work, don't keep trying the same thing.