CS410 Sp19 - ChengXiang Zhai

Michael Chang (mchang19)
Jeremy Varghese (jvarghs2)
Eric Wang (wcwang2)
Konrad Woo (kwoo3)
Bo Zheng (bozheng2)

Documentation for Course Project Code

1)Overview of Functions:

Plsa.py:-

build_corpus(), build_vocabulary(), build_term_doc_matrix(), initialize(), expectation_step(), maximization_step(), calculate_likelihood(), and plsa() perform exactly what was expected of them in Mp3.

main() - uses 5 topics extracted from user query to match to 5 “scientific articles” that could potentially help the user.

Scraper.py:

There is no real “function” in this script. When run, it scrapes documents from reddit posts about depression and puts them into a file for use as our database.

2)Implementation:

Plsa.py - We used the same code from Mp for most of it. The term_doc_matrix construction and the em algorithm is roughly the same apart from a few tweaks.

main()

This function is the only difference between the em algorithm that we implemented in Mp3. Over here, we extract the 5 topics from each document. The initial document is a user query, to which we match 5 most relevant “scientific research articles”, or “documents”, that the user might benefit from.

Scraper

Using the *Python Reddit API Wrapper* (PRAW), we were able to select a sub reddit board which we would use for document sourcing. Our board of choice is *depression_help*. The 'documents', in this case, are submission posts submitted by various users under the filtering 'flairs' of ***Providing Support***, ***Providing Advice***, ***Inspiration***, and ***Motivation***. Every post with one of those flairs is scraped, encoded in UTF-8, and if it contains more than 200 characters, is added to our document database file which will be fed into the PLSA function to provide appropriate documents.

3)Usage:

Our final interface is a jupyter notebook, where the inputs are transcripts from people's conversations about depression, which we assume could already be provided from [for example] a telephone call. When the user runs the two cells of our notebook, our project will scrape the source documents and output the top 5 documents per input that will provide help for the user with a potential depression problem.

4)Team member contributions:

Michael Chang: plsa.py

Jeremy Varghese：plsa.py

Eric Wang：plsa.py

Konrad Woo：scraper.py

Bo Zheng：scraper.py

Course Project Proposal

Mental health is now one of the biggest problems in the younger generation of America. According to research, suicide hotlines pick up rates have doubled since 2014 and the problem jumps out not only from statistics but from everyday events around us. At the core of numerous historical tragedies in recent years, traits of mental illness issues can be found. However, as the amount of mental illness patients rise, effective methods to treat symptoms have not increased in efficiency. Intending to explore more into this area, our group devised the idea to create a suicidal hotline recommender system.

The first step of the process will be converting user speech into a transcript. Although we consider the realm of audio processing and word recognition to be outside the scope of this course, we will assume that we already have the transcript ready. The second step is to do topic mining and analysis on the transcript to figure out the top 3 topics discussed in the transcript. As an example, a transcript could output the following 3 topics:

1. “Parents divorced”
2. “School not going well”
3. “I can’t make friends”

After the top 3 topics are obtained (above examples represent an extreme case where topics are well versed sentenced, but in reality, the topics will be word distributions as described in week 8 lecture). We will have a database of related documents ready for our algorithm to access and rank according to the extracted topics. In order to do so, we will scrape 500 articles from the web related to various areas of mental illness, for example articles on how to feel better if you are depressed etc.

When the database is ready, we will use the main 3 topics extracted before as user profile text and recommend documents. At this stage, we will take in retrieval techniques to score these 500 documents and use a score threshold for filtering out top 5 articles and return these as the output of the system.

It is usually the case that for an individual suffering mental illness such as depression, searching for solutions on generic search engines such as Google will usually lead to worse

outcomes. When an individual holds a negative mindset, he/she tends to be more attracted to negative materials. It is also a problem that patients cannot usually describe their symptoms and their complicated life experiences in a sentence on Google. Thus we hope that a recommender system that can take in a much bigger input can become a better alternative for individuals to seek online help.

The scope of the project might change as work progresses, but these are the core motivations behind our project.