--computer <-cosine formula had no gain from single-line queries so every document was a 1
fi rely on raw weights of documents for single line queries. A separate scorer for single word queries was implemented, ranking just off of the weighted term frequency.

--computer jgdfashglka

--jhkdfgalkja computer <- bad results from a poorly designed try-loop. The try except could except without initializing certain variables which would result in a crash. Fixed by correctly placing the variables

--this is a super long query homework computer test engineer ai uci zot thank you us we

---super mega uci zot professor engineering computer building uci weather whether bad poop toilet bathroom you quetza'lkop ><<>=*& shotr mssplеed up in big the tob e or not to bne that is the question
took a long time to run, so during the tf-idf construction for the query any terms that had a low normalized tf-idf score were dropped so that there were fewer terms to consider and look through,

----the teacher was eating an apple when your mom came in and sat on her
threshold for low normalized tf-idf looked a little high, this query had a lot of words thrown out (perhaps rightfully so, but still looked like a lot for such a small query) Threshold lowered accordingly

---the teacher ate an apple

--engineer
i had forgotten to stem the input for special case with a single word query input. fixed accordingly

---master of software engineering
Now correctly works as opposed to the milestone 2 bugs
---Modeling and in vivo live imaging of the Arabidopsis shoot apical meristem

----kgn ajkagfop afjg agoq; o
Query with all garbage text gave a div by zero error because the entire query vector length ended up being 0 with no matches. Added an appropriate try-except to catch a div by zero and manually set to zero instead.

---!@#$!#$%@!%}|{|<">?:!/
--- !#@$ ^%!@&^ }{ {!^}!$^{ ^
----apple %!$%!%$! computer ^!kaf science
---7pm

----machine learning

At this point i checked some links manually to make sure that they were relevant, and i found that they were completely irrelevant. I had forgotten to put reverse=True in the sorted call so that the order would be descending instead of ascending. Results were much much better after

----software engineer
----cristina lopes
--- taylor.ics.uci.edu

It felt weird that this would return nothing since technically that single ‘word’ does not exist within the text file of indexed words. I ran the query through the same regular expression i used to parse the tokens. Now it will match the filtered url without periods in the index

After that change, duplicate urls were being shown since i had not done duplicate detection. I added a list to keep track of which urls had already been displayed so that no doubles are shown.

---pattis.ics.uci.edu
---arcadia

Less than 5 results were showing as a result of awkward variable usage in fixing the last part

Added a separate tracker for number of links actually posted and added an except statement incase there were less than 5 links actually found, in which case the program would be allowed to stop trying to output more urls and continue asking for further search queries