Written Submission For

CSC 483/583: Homework 2 (Qns 1 to 8 sans Qn 7)

Spring 2017

By Mithun Paul

Today's Date: 19th Feb 2017

Time: 19:20 hrs.

Due by 11:59 P.M., February 19 (upload to D2L or, if on paper, turn it in class or in the instructor's office)

Because the credit for graduate students adds to more than 75 points, graduate students grades will be normalized at the end to be out of 75. For example, if a graduate student obtains 80 points on this assignment, her final grade will be $80 \times 75/95 = 63.2$. Under- graduate students do not have to solve the problems marked "grad students only." If they do, their grades will not be normalized but will be capped at 75. For example, if an un- dergraduate student obtains 80 points on this project (by getting some credit on the "grad students only" problem), her final grade will be 75.

Note only problem 7 requires coding.

Problem 1 (5 points)

Suggest what tokenization and normalized form(s) should be used for these words (including the word itself as a possibility). Justify your decision.

• ’Cos

• Shi’ite

• cont’d

• Hawai’i

• O’Rourke

• ain’t

• me@privacy.net

• <html> Some text </html>

Ans:

Note: The basic idea of tokenization and normalization is that if a user uses a form of the word which is not present in the text, it should still be returned. Or in other words, the user shouldn’t be penalized for using a colloquial or a different spelling for a certain word. Below are the given words and the normalized tokens which I feel will work, along with their justifications. This task becomes more and more relevant in this internet era, where neologisms are invented day by day by the twitter savvy crowd, which is bound to reduce the length of the words to save space. In any case, as long as we use the same algorithm of tokenization and normalization while doing indexing and query parsing, we will be fine.

- ’Cos
    - **Tokenized form:** Cos
    - **Normalized forms:** cos, because
    - **Justification:** This should in turn point to the index of the term “because”. “ ‘Cos ” is a fancy way of saying because. Other such forms of because include “coz” , “bcoz”, “becoz” etc. Note that methods used for normalization in this case include: case folding, lemmatization (cos -> because) etc. I did try this in wordnet lemmatizer, but it points to “ ‘Cos ” itself as the lemmatized form. I think that is wrong/they havent updated as per the new internet words. However, if you try the phrase “don't cry 'Cos it's over” in google, it returns results for : “don't cry 'Because it's over” . Which means, they have internally mapped the word, “ ‘Cos ” to “Because”, like I suggested
- Shi’ite
    - **Tokenized form:** Shi’ite

    - **Normalized form:** shiite
    - **Justification:** Shia is a version of Islam and its followers are called Shiites. The query can contain any form or variations of it like “Shi’ite, shi-ite, shiite” and it all should be pointed to the same index for the term “shiite”. However, note that these words shouldn’t point to the word “shite”, which is a different word and has different set of results. Note that methods used for normalization in this case include: case folding, lemmatization etc.
- cont’d
    - **Tokenized form:** contd
    - **Normalized forms:** contd.
    - **Justification:** “cont’d” is an abbreviation to the word “continued”. However it is mostly used at the bottom of a page to indicate that a letter or text continues on another page. Which means, it shouldnt be mapped to the term index of the word “continued”. It should instead have an index of its own, which should return documents in which this term is used to denote the continuation of a page or paragraph. In fact if you look at google search, the results for search for "cont'd presence" and “continued presence” is different justifying me previous argument.
    - Note that methods used for normalization in this case include: case folding, lemmatization etc.
- Hawai’i
    - **Tokenized form:** Hawaii
    - **Normalized form:** hawaii
    - **Justification:** The name of the island state is spelt with two i’s, as in “Hawaii”. So I decided to normalize it to a case folded form “hawaii”. However do note that many users do not know this and most probably will type in variations of the word “Hawai” with a single ‘i’. Care must be taken to normalize that to point to the same results as that of “hawaii”.
- O’Rourke
    - **Tokenized form:** ORourke
    - **Normalized form:** orourke
    - **Justification:** O’Rourke is an Irish Gaelic clan based most prominently in what is today County Leitrim and many people of that descent have that name as their last name. I dropped the single quote for tokenization, because otherwise, splitting it into two words (O and Rourke) is clearly wrong. Then I did case folding for normalization so that all similar results point similarly, i.e to the articles containing corresponding full names. (Eg: Mickey O’Rourke)
- ain’t
    - **Tokenized form:** aint
    - **Normalized form:** is not

    - **Justification:** Although widely disapproved as nonstandard, and more common in the habitual speech, the word "ain't" is flourishing in American English to denote any of the following meanings: am not, are not, is not, have not, has not. So I think it should be, at the end of the day, made to point to any of the above, depending on the context it is used. I just picked "is not" as an example
- me@privacy.net
    - **Tokenized form:** me @ privacy.net
    - **Normalized form:** me @ privacy.net
    - **Justification:** email address tokenization is a different ball game in itself. For example, if we decide to split based on dot ".", it will consider privacy.net as two words. However, for domain registration purposes, "privacy.net" is considered as a single word. However, we do want the username in an email address separated by dot, (Eg:mithun.paul@privacy.net) to be separated as two tokens. So a smarter way to do this is split based on @ first, keep the right hand side intact and then split the left hand side based on whatever the user needs (maybe, split based on underscore, dot, dashes etc). However, in most cases the email address is uniquely mapped using the entire string before @ (Eg:mithun.paul) . So the splitting of left hand side itself won't be necessary. Note that the splitting based on @ is necessary because we need to know which SMTP server is catering to this particular email address (Eg:privacy.net) . Also note that we should avoid doing case folding in such cases, since email addresses and domain names are registered as is, including capital letters.
- <html> Some text </html>
    - **Tokenized form:** Some text
    - **Normalized form:** some text
    - **Justification:** This depends on the usage. If you are feeding the entire string to a browser, you probably don't want to do anything to it. The browser does need to see the html tags to render is properly. However, if we are talking about just string parsing, i.e to extract the text out of that string, you can split based on the <html> beginning and ending tags (or < as the case be), and then do case folding and then split based on space to get individual tokens, which can then be run through a stemmer or lemmatization to get the normalized form. Note that here is an assumption that the text is index based on the words it contains and not as a single string.

Problem 2 (5 points)

Assume a biword index. Give an example of a document (could be a made up paragraph) which will be returned for a query of "New York University" but is actually a false positive which should not be returned.

Ans:

Consider the documents pasted below the explanation:

Explanation:

The Document 1 can be a false positive. As we know, in bi-word indexing, normally the query terms "New York" will have an index (Eg: Documents 1,2,3 below) and as will have "York University" (Eg: Documents 1,4 below). So when the engine sees a query like "New York University" it retrieves the results for these two indices and does a conjunction on them to return the result . Thus the paragraph given in Document 1 below, is returned. However, as you can see, it does not talk anything about "New York University" or NYU, as it is known colloquially. It is the blog of a writer who was born in "New York" state and went to "York University".

List of Documents.

Document 1: Presented as a result to the phrase query {"New York University"}

I am a writer. I was born in Albany in New York state. But I went to York University in Canada to get my bachelors in English literature. But my life in York University was bad, because of the weather and heavy winters. I love the New York Jets.

Document 2: is an example document that can be found in the postings list of a bi word index {"New York"}

Silicon Alley, centered in New York City, has evolved into a metonym for the sphere encompassing the New York City metropolitan region's high technology and entrepreneurship ecosystem; in 2015, Silicon Alley generated over US$7.3 billion in venture capital investment.[28] High tech industries including digital media, biotechnology, software development, game design, and other fields in information technology are growing, bolstered by New York City's position at the terminus of several transatlantic fiber optic trunk lines,[147] its intellectual capital, as well as its growing outdoor wireless connectivity.[148] In December 2014, New York State announced a $50 million venture-capital fund to encourage enterprises working in biotechnology and advanced materials; according to Governor Andrew Cuomo, the seed

money would facilitate entrepreneurs in bringing their research into the marketplace.[149] On December 19, 2011, then Mayor Michael R. Bloomberg announced his choice of Cornell University and Technion-Israel Institute of Technology to build a US$2 billion graduate school of applied sciences on Roosevelt Island in Manhattan, with the goal of transforming New York City into the world's premier technology capital.

Document 3: is an example document that can be found in the postings list of a bi word index {"New York"}

The New York Yankees are an American professional baseball team based in the New York City borough of the Bronx. The Yankees compete in Major League Baseball (MLB) as a member club of the American League (AL) East division. The Yankees are one of two Major League clubs based in New York City; the other is the New York Mets. The club began play in the AL in the 1901 season as the Baltimore Orioles (not to be confused with the modern Baltimore Orioles.) Frank Farrell and Bill Devery purchased the franchise (which had ceased operations) and moved it to New York City, renaming the club as the New York Highlanders. The Highlanders were officially renamed as the "Yankees" in 1913.

Document 4: is an example document that can be found in the postings list of a bi word index {"York University}

Through cross-discipline programming, innovative course design, diverse experiential learning and a supportive community environment, York University's 53,000 students get the education they need to have big ideas and endless career opportunities. York University is the 2nd largest university in Ontario, 3rd largest in Canada 200+ university partnerships across the globe $1 billion operating budget. With the founding of York University, Dr. Murray Ross (pictured here in 1960) realized his vision of a leading interdisciplinary research and teaching institution. Ross served as the University's first president for a decade and went on to receive the Order of Canada, among other honours.

Problem 3 (15 points)

Shown below is a portion of a positional index in the format: term: doc1: 〈position1, position2, ...〉; doc2: 〈position1, position2, ...〉; etc.

angels: 2: 〈36,174,252,651〉; 4: 〈12,22,102,432〉; 7: 〈17〉;

fools: 2: 〈1,17,74,222〉; 4: 〈8,78,108,458〉; 7: 〈3,13,23,193〉;

fear: 2: 〈87,704,722,901〉; 4: 〈13,43,113,433〉; 7: 〈18,328,528〉;

in: 2: 〈3,37,76,444,851〉; 4: 〈10,20,110,470,500〉; 7: 〈5,15,25,195〉;

rush: 2: 〈2,66,194,321,702〉; 4: 〈9,69,149,429,569〉; 7: 〈4,14,404〉;

to: 2: 〈47,86,234,999〉; 4: 〈14,24,774,944〉; 7: 〈199,319,599,709〉;

tread: 2: 〈57,94,333〉; 4: 〈15,35,155〉; 7: 〈20,320〉;

where: 2: 〈67,124,393,1001〉; 4: 〈11,41,101,421,431〉; 7: 〈16,36,736〉;

Which document(s) if any match each of the following queries, where each expression within quotes is a phrase query?

1. “fools rush in”

2. “fools rush in” AND “angels fear to tread”

Ans:

Qn 3.1: “fools rush in”

Ans: Document id: {2,4,7}

i.e the phrase “fools rush in” can be found in documents with document ids {2,4,7}

Qn 3.2: “fools rush in” AND “angels fear to tread”

Ans: For the phrase “angels fear to tread”, we get document id: {4}

So combining both for “fools rush in” AND “angels fear to tread”, we get:

{2,4,7} AND {4} =**{4}**

i.e the phrase “fools rush in” and the phrase“angels fear to tread” together can be found **only in document 4.**

Problem 4 (5 points)

Write down the entries in the permuterm index dictionary that are generated by the term “hope”.

Ans:

Say the term “hope” has its own postings list: Eg: [1,4,56,75,424]

For permuterm index dictionary, the following words must be term mapped to the word hope.

1. hope$
2. $hope
3. e$hop
4. pe$ho
5. ope$h

Explanation:

The above is done during Index time.

Now during query time: Say you get the query : “h*pe”

Rotate it to:

6. h*pe$
7. $h*pe
8. e$h*p
9. pe$h*

Match it with pe$ho-which in turn maps to hope - and return the documents in which the word hope appears.

Note that this query, can also match other words: Eg: horoscope.

This would have been stored under: pe$horosco.

On the way to find pe$h*, this also can be returned.

Problem 5 (15 points)

Compute the edit distance between “paris” and “arid”. What are the N (rows) and M (columns) dimensions of the edit distance matrix? Write down the N × M array of distances.between all prefixes as computed by the edit distance algorithm in Figure 3.5 in IIR. For each cell in the matrix, use the four-number representation to keep track of your intermediate results.

Ans:

Am making the below assumptions for the cost of operations:

insert (cost 1),

delete (cost 1),

replace (cost 1),

copy (cost 0)

Also, I am using the below formula to calculate four number representation individual cells

cost of getting here from my upper left neighbor (copy or replace) Cost=1	cost of getting here from my upper neighbor (delete)Cost=1
cost of getting here from my left neighbor (insert)Cost=1	the minimum of the three possible Ämove-ments"; the cheapest way of getting here

A raw version of the matrix without the four number representation is as below. This was one of my intermediate steps.

		A	**R**	**I**	**D**
	0	1	2	3	4
P	1	1	2	3	4
A	2	1	2	3	4
R	3	2	1	2	3
I	4	3	2	1	2
S	5	4	3	2	2

Qn) Compute the edit distance between “paris” and “arid”.

Ans: 2

Qn) What are the N (rows) and M (columns) dimensions of the edit distance matrix?

Ans: N = 7 (rows)

M = 6 (Columns)

Qn) Write down the N × M array of distances between all prefixes as computed by the edit distance algorithm in Figure 3.5 in IIR. For each cell in the matrix, use the four-number representation to keep track of your intermediate results.

Ans:

		A	**R**	**I**	**D**
	0	_|_ 1|1	_|_ 2|2	_|_ 3|3	_|_ 4|4
P	_|1 |1	1|2 2|1	2|3 2|2	3|4 3|3	4|5 4|4
A	_|2 |2	1|2 3|1	2|3 4|2	3|4 5|3	4|5 6|4
R	_|3 |3	2|2 4|2	1|3 3|1	3|4 2|2	4|5 3|3
I	_|4 |4	4|3 5|4	2|2 5|2	1|3 3|1	3|4 2|2
S	_|5 |5	5|5 6|5	5|3 6|3	3|2 4|2	2|3 3|2

Note that the backtracking part is highlighted in red.

Problem 6 (10 points) GRAD STUDENTS ONLY

Consider the following fragment of a positional index with the format:

word: document: 〈position, position, ...〉 ; document: 〈position, ...〉 ...

Gates: 1: 〈3〉 ; 2: 〈6〉 ; 3: 〈2,17〉 ; 4: 〈1〉 ;

Microsoft: 1: 〈1〉 ; 2: 〈1,21〉 ; 3: 〈3〉 ; 5: 〈16,22,51〉 ;

IBM: 4: 〈3〉 ; 7: 〈14〉 ;

The /k operator, word1 /k word2 finds occurrences of word1 within k words of word2 **(on either side),** where k is a positive integer argument. Thus k = 1 demands that word1 be adjacent to word2.

Qn 1. Describe the set of documents that satisfy the query Gates /2 Microsoft.

Ans: Document IDs: {1, 3}

2. Describe each set of values for k for which the query Gates /k Microsoft returns a different set of documents as the answer.

Ans: k ={1, 2, 6}

Explanation:

When k=1:

I.e. the two words are next to each other, on either side.

Results are document IDs: {3}

When k=2:

i.e. the two words have a word in between each other, on either side.

Results are document IDs: {3,1}

Note that this includes results from when k=1, since proximity search is inclusive of the previous results.

As you can see the result for when k=2 the only document that is different/new from when k=1 is

Document IDs:{1}

When k=3:

i.e. the given two words (Gates, Microsoft) have two words in between them, on either side.

Results in this query alone are document IDs: {}i.e there are no such documents.

So the final result output will be:

Results are document IDs: {3,1}

Note that this includes results from previous queries, since proximity search is inclusive of the previous results.

As you can see the result for when k=3 does not have anything different from previous result.

When k=4: Nothing to add

When k=5: Nothing to add

When k=6:

i.e. the given two words (Gates, Microsoft) have 5 words in between them, on either side.

Results in this query alone are document IDs: {2} -this is what the given question asks

So the final result output shown to the user be:

Results are document IDs: {3, 1, 2 }

Note that this result includes results from previous queries, since proximity search is inclusive of the previous results.

As you can see the result for when k=3 does not have anything different from previous result.

A visual perusal shows that as you increase k from here the set of documents in the result set does not change anymore.

Note: There is the possibility of an interesting case when k=16 (i.e there are 15 words in between Gates and Microsoft.) You can see that document 2 can be a result here. Similarly when k=15 (i.e there are 14 words in between Gates and Microsoft.) You can see that document 3 can be a result here. However, I think search engines kind of make the assumption that this is not relevant, since they are far beyond each other. Or at least they define a threshold of k before which they consider the results irrelevant.

Problem 7 (25 points undergrads, 35 grads)

Implement an inverted index that supports proximity search. Your program must take in one file containing one document per line, in a format similar to the one from Assignment #1 (see that assignment for a detailed description of the format). For example, you can use the file below to test your code:

Doc1 breakthrough drug for schizophrenia

Doc2 new schizophrenia drug

Doc3 new drug for treatment of schizophrenia

Doc4 new hopes for schizophrenia patients

To code this problem, you can use any programming language that is familiar to the instructor (C/C++, Java, Scala, Clojure, Python). You can use data structures available in your programming language of choice, e.g., dictionaries in Python or hash maps in Java/Scala, but you are not allowed to use open-source code that implements inverted indices, such as Lucene. You have to implement the inverted index and corresponding search operations from scratch.

The code submitted must compile and run. You must also include in your submission a Makefile/pom.xml/build.sbt file or shell script that allows the instructor to run the code, together with a README file that describes usage.

Please implement the following:

1. (25 points) Construct a positional index and add support for Boolean proximity queries using the /k operator. That is, word1 /k word2 finds occurrences of word1 within k words of word2 (on either side), where k is a positive integer argument. Hint: use the algorithm from Figure 2.12 in the IIR textbook.

Qn) What does your code return for the file above and the query: schizophrenia /2 drug?

Ans returned by my code: Found that the two terms that you asked viz., "schizophrenia" and "drug" exist in the proximity of: 2 in the document with Document Id: 1

Qn) How about schizophrenia /4 drug?

Ans returned by my code: Found that the two terms that you asked viz., "schizophrenia" and "drug" exist in the proximity of: 4 in the document with Document Id: 3

Note: Rest of the code will be submitted through D2L.

2. (GRAD STUDENTS ONLY: 10 points) Modify the above algorithm to be directional. That is, the query word1 /k word2 must return occurrences of word1 strictly before word2, within k words.

Qn: What does your code return for the file above and the query: schizophrenia /2 drug?

Ans returned by my code:

Found that the two terms that you asked viz., "schizophrenia" and "drug" exist in the same document. i.e in the document with Document Id: 1. But in a directional query you cant have the term2 occurring BEFORE term1. So ignoring this result and moving on.

Found that the two terms that you asked viz., "schizophrenia" and "drug" exist in the same document. i.e in the document with Document Id: 3. But in a directional query you cant have the term2 occurring BEFORE term1. So ignoring this result and moving on.

Given terms viz., "schizophrenia" and "drug" don't exist in the same document, atleast not in the proximity and/or direction you asked for.

Note: Code will be submitted through D2L

Problem 8 (5 points)

Artificial intelligence (AI) and automation in general clearly improve the quality of our lives (think Google). However, in many cases, they also eliminate jobs (e.g., the self-driving car impacts the livelihood of taxi drivers). Most often, these negative side effects impact the people least prepared to recover. If you were a policy maker, how would you address this problem? Please describe your solution and explain why you believe it will work.

Ans:

For the younger crowd, let's say who are aged less than 30 years old, picking up a new skill is not that difficult. Assuming they have time, money and energy available. I am talking about the learning caliber of a human being. But for the older crowd, say taxi drivers, who probably are already the only breadwinner of a family of 5, it's not easy to tell them to learn another skill one fine day. Most probably the only job they know is driving. But even if they have time, money and energy, they probably will lack the brain caliber to learn a new skill at that age. So below are a few solutions I can think of to implement this transition smoothly.

1. Transition gradually
    a. Don't make it a coup de grace. Don't make the change on one fine morning. For example if it is decided that the taxis have to be replaced with self-driving vehicles, do it across months.
2. Tax the automated cars more initially.
    a. This can be in the form of higher charges of fare that the passengers have to pay. This is to encourage the slow transition of jobs from a taxi driver to the self driving vehicle.
3. Resettle families.
    a. For every job lost by the automated taxis, make sure the policy maker finds an equivalent job for the driver that loses that job. This needn't be in the same domain. This can be in another related domain. Plus the government must support his family during the time period in which the person learns the new skill.
4. Selection of inventions
    a. Allow only the inventions that supplement human career, not remove or replicate them. For example, machine learning algorithms can now outperform a doctor in diagnosis of a disease, given enough symptoms of the patient. However, this shouldn't be allowed to take over a doctor's job. It should be presented as a secondary input for the doctor to verify/confirm his own findings. Many robotics hands assist the surgeons these days in conducting micro surgeries which would have been impossible say 40 years ago. None of them replaced the doctor, it just assisted him.
5. Human supervision.
    a. Do not allow a machine to operate by itself. Always have a human to monitor it.

For example in case of self-driving car, for every taxi driver that loses his job, make him the secondary driver in such self-driving cars. While it is beneficial for the car manufacturer to allow completely self driving cars, the law makers should intervene and make them stop as semi-supervised learning. Not just completely unsupervised learning. In airplanes now, there is an auto pilot, but still monitored by airline pilots all the time.

6. Arts
    a. Arts is a field which is often overlooked due to the lack of quick income and stability in it. Many humans have many talents and soft skills which they might have given up in pursuit of putting bread in their plate. Instead encourage arts by federal funding. For example if a Taxi driver has a talent of singing, he must be encouraged to become a singer. After all, there are domains where machines cannot replace humans ever. I think arts might be one of them. I think it might be decades before a machine can come up with a creative painting or a beautiful poem.
7. Progeny
    a. Another interesting phenomenon I have noticed in the country I grew up was that, the older generation who became irrelevant due to the invent of computers, were replaced by the equivalent younger generations from the same family. For example, parents, who lost their jobs as typists (on a typewriting machine), because computers took over, found that, by investing the same time, money and energy, their kids were able to pick up this task of learning computers much easily. That way the same family became benefited by the new jobs that were introduced in place of a typist. The same family was even paid more, because there was someone in the family who could take over the new lucrative computer based market, in place of the breadwinner typist dad. Computers might have made the career of a typist go void, but at the same time, it opened so many new venues like Digital desktop operator, photoshop operator, MSWord operator etc.

But do note that I think such technological inventions and advancements are very imperative for humanity as a whole to progress. It was the same concerns that people raised when Henry Ford had introduced assembly lines in the early 1920’s. Things weren't different when computers were becoming widely used in the 1990’s and 2000’s. Everyone kept complaining about the technology going to make them lose their jobs. But the truth is computers and cars brought in much more opportunities also. There weren't just blindly cutting out jobs. Any invention will have its pros and cons. When one door closes, some others will open , given the right amount of time. It is this amount of time which we need for the people and evolution to adapt and refurbish itself in a new way. Rome wasn't built in a day.