		grades		newsletter		carl hayden		fall semester		stimulant web	
		TF	TF/IDF	TF	TF/IDF	TF	TF/IDF	TF	TF/IDF	TF	TF/IDF
Document ID Ranked	1.	22149	22156	1701	1701	21827	15128	845	871	14348	26
	2.	22156	233	1718	13778	16874	16676	882	5	13973	2085
	3.	233	22149	13778	2161	16276	17678	103	103	26	18590
	4.	22919	19822	1309	21523	15128	15130	20258	3762	10895	2094
	5.	18699	19590	16653	1238	16689	16276	3595	30	13988	14348
	6.	1851	22913	2161	1309	17429	16904	30	882	13984	13973
	7.	19590	21047	21523	1718	16911	21827	88	79	10768	18600
	8.	18750	1851	18378	16653	18362	18362	275	21231	10749	13948
	9.	22913	22936	1287	18378	16853	17429	3599	20258	13948	18602
	10.	19822	18699	1238	18372	18343	15114	871	845	10750	13988
Time to (ms)	get results	3	3	6	2	22	20	14	10	22	28
	sort results	2	5	5	2	16	21	13	10	19	21
	handle the query	76	74	72	66	313	316	257	267	478	493
Total results		384		365		3619		2333		5994	

Table: 1

Q: A few sentences explaining your algorithm?

A: The algorithm is divided into 2 parts: data preprocessing and query handling.
Data preprocessing:

- The IDF for each term is computed as log (total no. of docs/ no. of docs which contain the term).
- Inverse of each document magnitude is computed and stored, and also the frequency of the maximum frequent term in a document is stored.

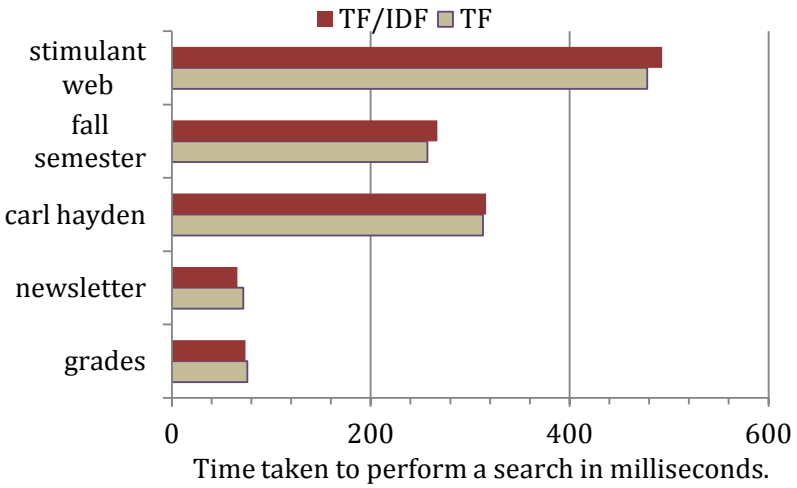
Query handling:

- Query is inputted by the user until the input is 'quit'.
- For each term in the query, the relevant documents are fetched from the inverted index and cosine product is computed and stored/updated in a hashtable.
- All the results stored in hashtable are then sorted according to the similarity using 2-d array.
- And then the results are displayed.

Analyze the speed/efficiency of your algorithm:
Q: How long did it take to compute document norms? How long did it take to get the results? To sort them?
A: Time taken to compute documents norms for TF is approximately 7 seconds. Time taken to compute documents norms and IDF is approximately 9 seconds. The results and sort time is mentioned in table1.

Q: Plot a bar chart showing the time taken to perform a search of the five queries above for TF and for TF/IDF. Is the difference significant? Is it expected?

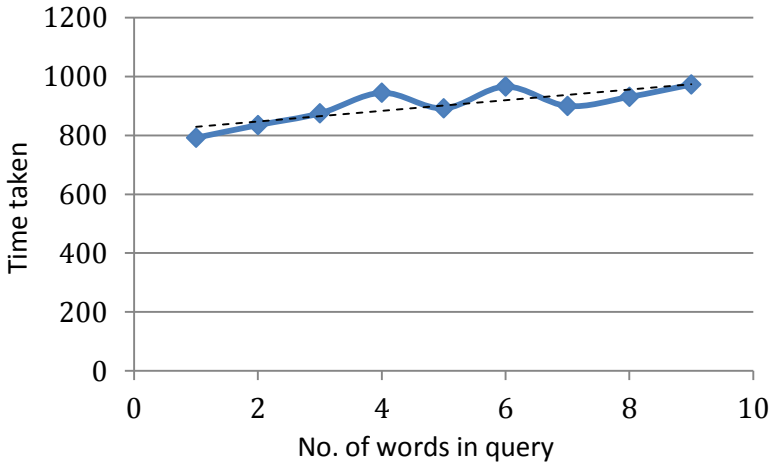
A: As we can observe from plot 1, that the difference in time taken between TF and TF/IDF is not significant. This is the expected behavior because all the processing in either case is done before any query is entered and after the query is entered similar computations are done for both of them.



Plot 1: Bar chart showing the time taken to perform the search of the five queries for TF and TF/IDF.

Q: Plot a chart showing the time taken to perform a search vs the number of words in the query. Explain the result?

A:



Plot2: curve to show time taken vs number of words in the query.

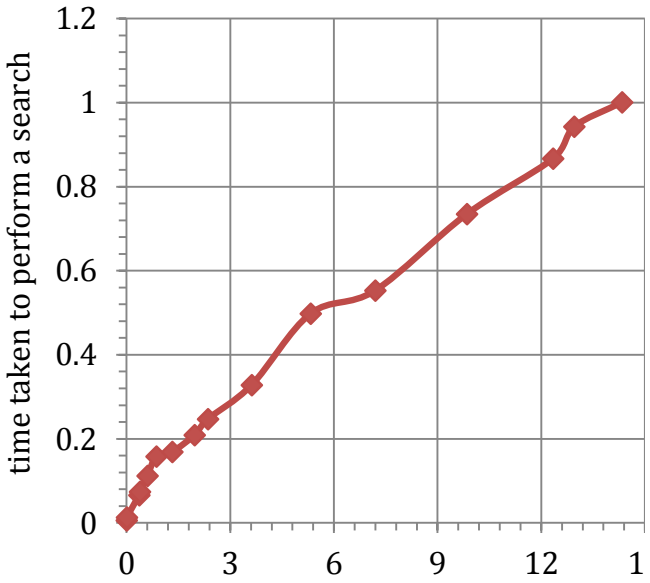
Query used: "arizona state university tempe america international student admission sponsorship"

The curve will be monotonically increasing* because when we add another term to the query, the new query will either return the same number of documents or will increase the documents. The query processing loop will run for this term and will either add or update results. So the processing time will increase.

**Assumptions: new term is added to the already existing query.*

Q: Plot a curve showing the time taken to do a query vs the number of results in the query. To do this question, you will need to figure out query keywords of your own that have various number of results, so that you can plot a meaningful curve?

A:

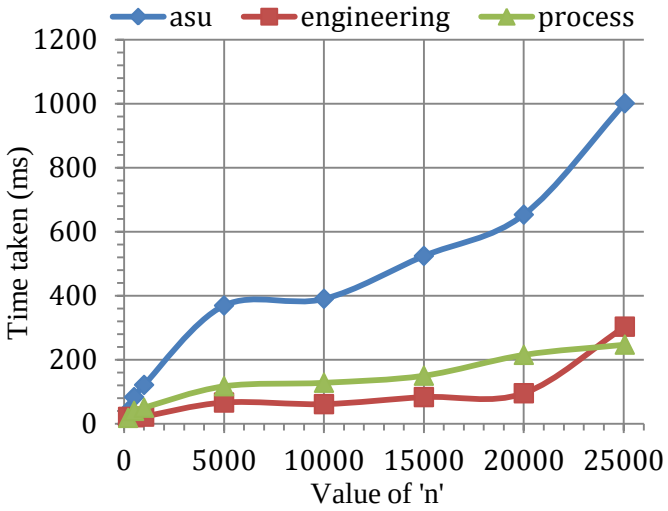


No. of results	Time Taken	query
0	6	the
6	13	brute
365	66	newsletter
384	74	grades
603	112	admission
862	158	america
1320	169	hall
1967	209	application
2353	247	process
3620	328	international
5325	498	tempe
7195	553	library
9848	735	state
12341	867	university
12953	943	arizona
14337	1001	asu

Plot3: Plot to show the time taken to do query vs the number of results in the query

Q: Artificially restrict the size of the document set by ignoring any documents with id above a certain number 'n'. Plot the time taken to do the query against n. Explain the result?

A: As we can observe from the plot4: that as the corpus size is increasing the time taken to handle a query also increases monotonically. The curve would also depend on the concentration of results for a query. The time might increase for range of value of 'n' if most of the results for that query lie in that range.



Value of 'n'	Time taken		
	process	asu	engineering
200	18	39	21
500	40	83	20
1000	50	121	22
5000	117	369	66
10000	128	390	61
15000	150	524	83
20000	215	653	95
25054	247	1001	303

Plot4: Artificially restricted the size of document set to a certain number 'n'. time taken vs 'n' curve

Q: Provide a theoretical complexity analysis of the algorithm?
A: Let's consider the total number of documents in corpus is 'n'. The total number of terms indexed is 'm'. Number of terms in the query is 'q'.

Data Preprocessing

- To compute IDF of each term and square of document magnitude, the worst case running time would be O(mn).
- To compute the inverse of document magnitude O(n).
- So, for preprocessing the worst case running time will be **O(mn)**.

Query Handling

- To fetch the results O(qn).
- To convert the hashtable to an array O(n).
- To sort the fetched results O(n.log n).
- To display the results O(n).
- So, for query handling the worst case running time will be **O(n.log n)**.
  **For practical scenario we can assume the size of query to be small, that is q << (log n)*

Analyze the correctness of the results:

Q: Are the TF results relevant, as judged by a human? Are the TF/IDF results relevant? Which is better?
A:

- **fall semester**
  The top 2 results given by TF are taking about "Withdrawal from Classes-General Policy" and "Student Employment Eligibility Criteria" which is *not relevant* according to me. The top 2 results given by TF/IDF are "Enrollment Status" and "Course-Related Deadlines" which again are *not relevant*.

- **Grades**
  I was expecting the grading system used by ASU but the top 2 results given by TF is taking about "Plus/Minus Grades" which according to me are not so relevant. I found the second result given by IDF to be quite relevant to my expectation whereas the 1st result is again related to "plus/minus -grades".
- **carl hayden**
  Expected results were regarding Carl Trumbull Hayden but the first result by TF was regarding Carl J.Schramm and the second result was related to Hayden library "Marketing Resources" with reference to Carl Hayden. So both the results are not relevant. The 1st result is relevant as it aligns my expectation but the second result talks about "Collection Development" of Hayden library which is not relevant.

So, TF/IDF results are relatively better than TF results.

Q: Compare TF vs TF/IDF: Is the order of the results different with a single keyword? two keywords?
A: As we can note from table1, that the order of results is different for TF and TF/IDF with a single or a two word query.

Q: Which term(s) have the lowest IDF in the corpus? Is this expected?
A: Terms which have lowest IDF:

- html
- body
- title
- head
- href
- p
- text
- tr
- type

This is expected because these are html tags and will occur in most of the documents.