

HW3: NLP on the Web

Note: Please submit the written aspects of assignments in postscript or PDF format only. I *highly* recommend you use L^AT_EX to prepare the assignments. A solution will be posted here after the due date. See <http://www.cs.utah.edu/classes/cs5964/handin.html> for handin instructions.

Part A

(due 19 Nov)

Suppose we have the following document collection:

| | |
|----|------------------------------------|
| D1 | the man ate a tasty sandwich |
| D2 | the sandwich was tasty and filling |
| D3 | the man is tasty to the alien |
| D4 | an alien is not a man |
| D5 | the man is filling the box |

Question 1: Compute the term frequency matrix for the above document collection.

Question 2: Compute the document frequencies for each word in the document collection.

Question 3: Compute the tf.idf vectors for each of the documents in the collection.

Question 4: Compute the cosine similarity between each of the document and the query “tasty alien”. Sort the documents by their cosine similarity. Do the results make sense? If not, why not?

Part B

(due 5 Dec)

Many of the questions on this part of the assignment relate to the list of questions below:

1. What temperature does water boil at?
2. How long does it take to do ANLP projects?
3. Who is the prime minister of France?
4. Who was the prime minister of France in 1832?
5. What countries border Hawaii?
6. What is the best grad school to attend for NLP?
7. How often are the Olympics held in North America?

Question 1: Classify each of the above questions into “factoid”, “list” or “definition.” Do any not really fit these categories?

Question 2: Using the hierarchy on slide 21 of the QA1 notes, classify the answer type for the above questions (wherever possible).

Question 3: For each question in the above list, convert it into “answer” format by doing question inversion. For instance, the first one might map to “Water boils at XXXX.” Issue each of these queries to your favorite search engine and look at the snippets from the top 10 answers. Can you find the correct answer for each in there? If not, why not?

Question 4: For those questions to which you couldn’t find an answer in Question 3, try relaxing the query, ala the AskMSR system and repeat. Can you find any additional answers (in the top ten results)? If so, what did you have to do to the query? Could this be automated?

Question 5: For each of the questions, write something that might pass for first-order logic ala LCC's theorem prover. For those that you found answers for in Question 3, try to write the answer snippet as a a first-order logic statement. Is the matching process obvious or would you have to revert to external knowledge like WordNet?