CS6370 Natural Language Processing

Assignment 2

Due Date: 20/03/2023 — 11:59 PM

List of Collaborators: ***CS20B070 Sasubilli Yuvan, EE20B029 Chollangi Dheeraj Sai***

---

1. Now that the Cranfield documents are pre-processed, our search engine needs a data structure to facilitate a query's'matching' process to its relevant documents. Let's work out a simple example. Consider the following three sentences:

   - S1: Herbivores are typically plant eaters and not meat eaters
   - S2: Carnivores are typically meat eaters and not plant eaters
   - S3: Deers eat grass and leaves

   Assuming are, and, not as stop words, arrive at an inverted index representation for the above documents (treat each sentence as a separate document).

Solution:

Term	Inverted Index Representation
herbivor	S1
typic	S1, S2
plant	S1, S2
eater	S1, S2
meat	S1, S2
carnivor	S2
deer	S3
eat	S3
grass	S3
leav	S3

2. Next, we must proceed on to finding a representation for the text documents. In the class, we saw about the TF-IDF measure. What would be the TF-IDF vector representations for the documents in the above table? State the formula used.

Solution:

	Counts, tf_i						Weights, = tf_i*IDF_i		
Terms	D_1	D_2	D_3	df_i	D/df_i	IDF_i	D_1	D_2	D_3
herbivor	1	0	0	1	3	0.4771	0.4771	0	0
typic	1	1	0	2	1.5	0.1761	0.1761	0.1761	0
plant	1	1	0	2	1.5	0.1761	0.1761	0.1761	0
eater	2	2	0	2	1.5	0.1761	0.3522	0.3522	0
meat	1	1	0	2	1.5	0.1761	0.1761	0.1761	0
carnivor	0	1	0	1	3	0.4771	0	0.4771	0

	Counts, tf_i						Weights, = tf_i*IDF_i		
Terms	D_1	D_2	D_3	df_i	D/df_i	IDF_i	D_1	D_2	D_3
deer	0	0	1	1	3	0.4771	0	0	0.4771
eat	0	0	1	1	3	0.4771	0	0	0.4771
grass	0	0	1	1	3	0.4771	0	0	0.4771
leav	0	0	1	1	3	0.4771	0	0	0.4771

```
VECTOR SPACE REPRESENTATION
S1 = 0.4771<herbivor> + 0.1761<typic> + 0.1761<plant> + 0.3522<eater> + 0.1761<meat>
S2 = 0.4771<carnivor> + 0.1761<typic> + 0.1761<plant> + 0.3522<eater> + 0.1761<meat>
S3 = 0.4771<deer> + 0.4771<eat> + 0.4771<grass> + 0.4771<leav>
```

3. Suppose the query is "plant eaters", which documents would be retrieved based on the inverted index constructed before?

Solution: Documents S1 and S2 will be retrieved because their vector space representation has dimensions of "plant" and "eater" that match the query's dimensions. Hence there is a cosine similarity between these two vectors.

4. Find the cosine similarity between the query and each retrieved document. Rank them in descending order.

Solution:

```
VECTOR SPACE REPRESENTATION
Q = 0.1761<plant> + 0.1761<eater>
```

$|Q| = \sqrt{0.1760^2 + 0.1760^2} = 0.2489$
$|S1| = \sqrt{0.4772^2 + 0.1760^2 + 0.1760^2 + 0.1760^2 + 0.352^2} = 0.6667$
$|S2| = \sqrt{0.4772^2 + 0.1760^2 + 0.1760^2 + 0.1760^2 + 0.352^2} = 0.6667$
$|S3| = \sqrt{0.4772^2 + 0.4772^2 + 0.4772^2 + 0.4772^2} = 0.9543$
cos(Q,S1) = 0.5599, cos(Q,S2) = 0.5599, cos(Q,S3) = 0
Ranking : S1, S2

5. Is the ranking given above the best?

Solution: No, The given ranking is not the best because the query is " plant eaters," We directly match the question with the documents where the words are present. The third document, S3, has eat, grass, and leaves, similar to eaters and plants. Most importantly, the query plant eaters are relevant to S3, "deers eat grass and leaves," since deers are plant eaters than S2, which talks about carnivores. So The ranking should be S1 > S3 > S2 in the order of relevance.

6. Now, you are set to build a real-world retrieval system. Implement an Information Retrieval System for the Cranfield Dataset using the Vector Space Model.

Solution: Code

7. (a) What is the IDF of a term that occurs in every document?
   (b) Is the IDF of a term always finite? If not, how can the formula for IDF be modified to make it finite?

Solution:

- The IDF of a term in every document is zero according to the formula IDF_i = log(N/n).
- No, it isn't finite when the term is not present in the corpus but present in the query and leads to a division-by-zero. To avoid this, a modification to the formula is made by adding 1 to the denominator, which is called smoothing.
  Modified formula is IDF_i **= log((1 + N)/(1 + n))**

8. Can you think of any other similarity/distance measure that can be used to compare vectors other than cosine similarity? Justify why it is a better or worse choice than cosine similarity for IR.

Solution: Cosine similarity is a popular choice for information retrieval (IR) because it is computationally efficient and works well with high-dimensional data. It is also scale-invariant, meaning that it is not affected by the magnitude of the vectors.
Euclidean distance is another similarity measure that is commonly used. It is sensitive to the magnitude of the vectors, which means that it is not scale-invariant. Manhattan distance is similar to Euclidean distance, but it is not sensitive to the magnitude of the vectors. Spearman's rank correlation coefficient is used for ordinal data. Pearson correlation coefficient is defined as the covariance of the two variables divided by the product of their standard deviations. It is a good choice when the vectors represent continuous variables
Cosine similarity is not a good choice when the vectors are sparse; it treats all dimensions equally, even if they are zero. Other similarity measures, such as Jaccard similarity, are better suited for sparse vectors When the vectors have different lengths when the data is continuous, and When the vectors have opposite directions

9. Why is accuracy not used as a metric to evaluate information retrieval systems?

Solution: Accuracy fails to distinguish different classes properly for problems involving data associated with highly imbalanced data sets. This metric is only useful when the data is more evenly distributed between classes. Information retrieval systems involve data associated with a high-class imbalance, because of which accuracy is not used for evaluation.

10. For what values of α does the F_α -measure give more weightage to recall than to precision?

Solution: For every $\alpha > 1$, F_α measure gives recall more weightage than precision. It can be inferred easily from the formula.
$F_\alpha = (\alpha^2 + 1)pr/(\alpha^2 p + r)$
Where p = precision
r = recall

11. What is a shortcoming of Precision @ k metric that is addressed by Average Precision @ k?

Solution: Precision @ k metric doesn't penalize relevant items in the lower ranks. For two IR systems with the same number of relevant documents, Average precision @ k is a better performance measure as it can penalize recommendations placed at lower ranks.

12. What is Mean Average Precision (MAP) @ k? How is it different from Average Precision (AP) @ k?

Solution: Mean Average Precision (MAP) @ k is the mean of Average Precision @ k over different queries. This metric is used if we use a set of queries for evaluation instead of a single query.
On the other hand, Average Precision @ k is used for evaluating single query retrievals

13. For Cranfield dataset, which of the following two evaluation measures is more appropriate and why? (a) AP (b) nDCG

Solution: For the Cranfield dataset, nDCG is a better metric for evaluation measure since it considers the degree of relevance of the documents (which in our case is provided as a scale of 4) in contrast to the AP approach, which only differentiates relevance in a binary manner.

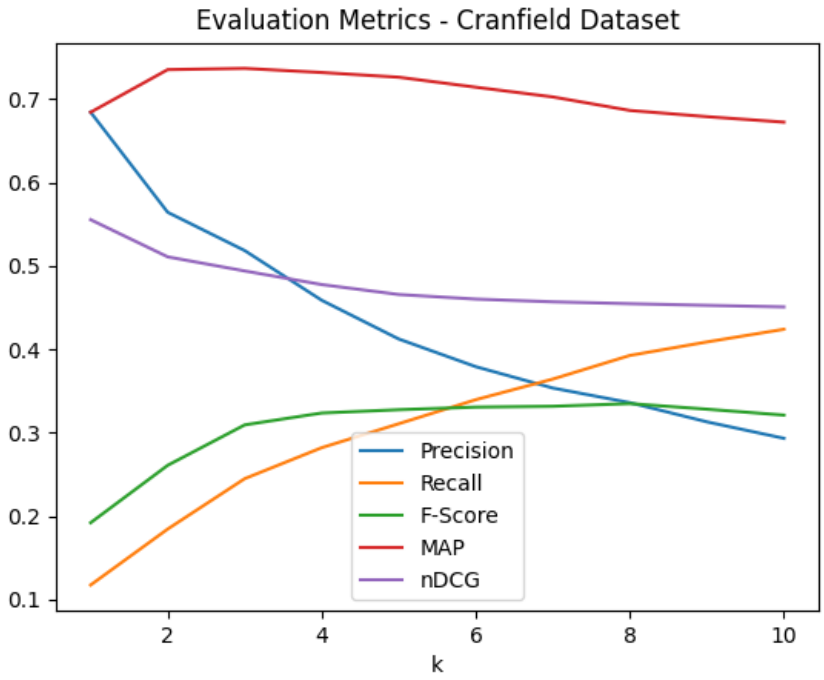
14. Implement the following evaluation metrics for the IR system:

    - Precision @ k
    - Recall @ k
    - F-Score @ k
    - Average Precision @ k
    - nDCG @ k

Solution:
Precision, Recall and F-score @ 1 : 0.6844, 0.1174, 0.1921
MAP, nDCG @ 1 : 0.6844, 0.5556
Precision, Recall and F-score @ 2 : 0.5644, 0.1845, 0.2610
MAP, nDCG @ 2 : 0.7355, 0.5110
Precision, Recall and F-score @ 3 : 0.5185, 0.2450, 0.3094
MAP, nDCG @ 3 : 0.7370, 0.4941
Precision, Recall and F-score @ 4 : 0.4589, 0.2821, 0.3237
MAP, nDCG @ 4 : 0.7321, 0.4777
Precision, Recall and F-score @ 5 : 0.4124, 0.3108, 0.3277
MAP, nDCG @ 5 : 0.7264, 0.4658
Precision, Recall and F-score @ 6 : 0.3793, 0.3397, 0.3307
MAP, nDCG @ 6 : 0.7144, 0.4603
Precision, Recall and F-score @ 7 : 0.3536, 0.3644, 0.3318
MAP, nDCG @ 7 : 0.7028, 0.4571
Precision, Recall and F-score @ 8 : 0.3361, 0.3927, 0.3350
MAP, nDCG @ 8 : 0.6866, 0.4548
Precision, Recall and F-score @ 9 : 0.3131, 0.4089, 0.3283
MAP, nDCG @ 9 : 0.6791, 0.4528
Precision, Recall and F-score @ 10 : 0.2933, 0.4241, 0.3211
MAP, nDCG @ 10 : 0.6726, 0.4509

15. Assume that for a given query, the set of relevant documents is listed in *cran_qrels.json*. Any document with a relevance score of 1 to 4 is considered as relevant. Report the graph with your observations based on it.

Solution:



Observations

1 The IR system is effective when precision decreases as rank increases. It indicates that all relevant data documents are being located in the highest positions.

2 Recall is found to rise monotonically as k increases, which is 0.7, also expected when more documents are retrieved.

3 The harmonic mean of recall and precision is the F-score. The f-score initially rises with k, and the graph becomes flat with rising k, according to the precision and recall graphs.

4 It can be observed that Mean Average Precision increases at first, reaches a maximum, and then starts to decrease. The most relevant documents can be in the top ranks because they represent the average across all queries.

5 Hence, all nDCG calculations are relative values of 0.0 to 1.0 and are similar across queries. At bigger values of k, there is a slight decrease in the nDCG values, which otherwise appear to remain relatively constant.

16. Analyse the results of your search engine. Are there some queries for which the search engine's performance is not as expected? Report your observations.

Solution:

- On query "how much is known about boundary layer flows along non-circular cylinders ." The retrieved docs are 382, 785, 1154, 105, and 1055, but from the Cranfield dataset, the docs are 754,788, 785, 786, and 787. Only one doc is common, and the most relevant doc is absent. The system's performance isn't expected this way. At least the most relevant doc will be retrieved, but that did not happen.

- On query "paper on aerodynamics," The retrieved docsID are 925,137,1379,1066,749. By manually checking the retrieved docs with the given measures, the docs contain aerodynamics words in good numbers, so the IR worked fine on this query.
- On query ”what parameters can seriously influence natural transition from laminar to turbulent flow on a model in a wind tunnel .” The relevant documents for this query are: 546, 608, 406, 606, 710. The IR system retrieved these as the top 5 docs: 294, 418, 315, 295, 1155. Not even one document is in common .The IR system did not give one relevant doc for the given query

17. Do you find any shortcomings (s) in using a Vector Space Model for IR? If yes, report them.

Solution:
Disadvantages of the vector space model

- It doesn't care about the order of terms in the context of both queries and documents.
- It assumes the statistical occurrence of terms to be independent, which is not the case (co-occurrences, collocations)

18. While working with the Cranfield dataset, we ignored the titles of the documents. But titles can sometimes be extremely informative in information retrieval, sometimes even more than the body. State a way to include the title while representing the document as a vector. What if we want to weigh the contribution of the title three times that of the document?

Solution: We can give more importance to the title than the document body by a small modification in calculating term frequency. We can merge the title to the body and give a weight of 3 for the term present in the title, saying the term has double the importance of the term present in the body of the document The modified formula would be
$T_f = 3 \times$ (count of term present in the title) + (count of term present in the body)

19. Suppose we use bigrams instead of unigrams to index the documents, what would be its advantage(s) and/or disadvantage(s)?

Solution:
Advantages: Bigrams are pairs of adjacent words in a text. Using bigrams instead of unigrams to index the documents in NLP can have advantages and disadvantages. Bigrams capture more context than unigrams. This can be useful in tasks such as language modeling, where the goal is to predict the next word in a sequence. They can help reduce the sparsity of the feature space, improving the performance of some machine learning algorithms. Bigrams are better at context and sequence modeling and have better precision than unigrams.
Disadvantages: However, bigrams can also increase the feature space's dimensionality, making some algorithms slower and more computationally expensive. The recall is lower compared to unigrams.

20. In the Cranfield dataset, we have relevant judgments given by the domain experts. In the absence of such relevant judgements, can you think of a way in which we can get relevance feedback from the user himself/herself? Ideally, we would like to keep the feedback process to be non-intrusive to the user. Hence, think of an 'implicit' way of recording feedback from the users.

Solution: Users may indicate relevance explicitly using a binary or graded relevance system. Binary relevance feedback indicates that a document is either relevant or irrelevant to a given query. Graded relevance feedback indicates the relevance of a document to a query on a scale using numbers, letters, or descriptions (such as "not relevant", "somewhat relevant," "relevant," or "very relevant").
Implicit feedback is inferred from user behavior, such as noting which documents they do and do not select for viewing, the duration spent viewing a document, or page browsing or scrolling actions. One can use many signals during the search process for implicit feedback and the types of information to provide in response. An example is **dwell time**, which measures how long a user spends viewing the page linked to in a search result. It indicates how well the search result met the user's query intent and is used as a feedback mechanism to improve search results.

References

1. https://scikit-learn.org/stable/
2. https://en.wikipedia.org/wiki/Relevance_feedback.
3. https://en.wikipedia.org/wiki/Cosine_similarity
4. https://medium.com/swlh/rank-aware-recsys-evaluation-metrics-5191bba16832