Assignment 2

CS6370: Natural Language Processing

Submitted by:
R Rishaab Karthik (MM19B046),
B Aditi (MM19B022),

Contents

Toy search engine

1 Solutions

1.1 Inverted Index representation

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

1.1.1 Tokenization,Stop word removal, and Lemmatization

S1: {Herbivore,typical, plant, eat, meat}
S2: {Carnivore, typical, meat, eat, plant}
S3: {Deer, eat, grass, leaf}

1.1.2 Inverted index representation

- Herbivore: S1
- typical: S1,S2
- eat: S1,S2,S3
- plant: S1,S2
- meat: S1,S2
- Carnivore: S2
- Deer: S3
- grass: S3
- leaf: S3

1.2 TF-IDF

$$IDF = log(\frac{D}{d_i})$$

$$Weights = tf_i * IDF$$

The below table summarizes IDF and weights of each of the terms in each document.

		Counts tfi					Weights			
TERMS	Q	S1	S2	S3	dfi	IDFi	Q	S1	S2	S3
Herbivore	0.00	1.00	0.00	0.00	1.00	0.48	0.00	0.48	0.00	0.00
typical	0.00	1.00	1.00	0.00	2.00	0.18	0.00	0.18	0.18	0.00
eat	1.00	1.00	1.00	1.00	3.00	0.00	0.00	0.00	0.00	0.00
plant	1.00	1.00	1.00	0.00	2.00	0.18	0.18	0.18	0.18	0.00
meat	0.00	1.00	1.00	0.00	2.00	0.18	0.00	0.18	0.18	0.00
Carnivore	0.00	0.00	1.00	0.00	1.00	0.48	0.00	0.00	0.48	0.00
deer	0.00	0.00	0.00	1.00	1.00	0.48	0.00	0.00	0.00	0.48
grass	0.00	0.00	0.00	1.00	1.00	0.48	0.00	0.00	0.00	0.48
leaf	0.00	0.00	0.00	1.00	1.00	0.48	0.00	0.00	0.00	0.48

1.3 Query results

Based on the inverted index constructed before we would get all the three documents, as the query has both the terms plant and eat.

1.4 Cosine similarity

$$cos\theta_{S1} = \frac{Q \cdot S_1}{|Q||S_1|} = 0.31$$

$$cos\theta_{S2} = \frac{Q \cdot S_2}{|Q||S_2|} = 0.31$$

$$cos\theta_{S3} = \frac{Q \cdot S_3}{|Q||S_3|} = 0$$

1.5 Ranking

Based on the cosine similarity scores documents S1 and S2 are ranked first and Doc S3 is not part of the retrieved docs as it has a zero dot product. From observation we know neither the retrieved set of documents nor the ranking is not the best. The performance of the TF-IDF along with cosine similarity is poor in this scenario.

1.6 Vector space model for cranfield dataset

The completed code functions are attached with the report along with the output.

1.7 IDF

(a) The IDF for a term present in all documents is 0 as the value of $log(\frac{D}{d_i}) = log(1) = 0$.

(b) The IDF of a term is not always finite when the term frequency of a document is zero. Thus $log(\frac{D}{d_i})$ is undefined. We can use methods like laplace or additive smoothing, this ensures IDF values to be always finite. This formula ensures that the IDF value is always greater than or equal to 1, and it avoids the problem of zero or very small document frequencies for certain terms in the corpus.[1]

$$IDF = log(\frac{D+1}{d_i+1}) + 1$$

1.8 Distance measures

- **Euclidean distance:**It is a measurement of the distance in a high-dimensional space along a straight line between two vectors.The total of the squared differences between the relevant dimensions in the two vectors is used to compute it. When the magnitude of the vector dimensions matters, the Euclidean distance is a solid option for IR systems. When the direction of the vectors matters more than their size, it may be less successful than cosine similarity.
- **Jac card similarity:** Jaccard similarity is a measure of the similarity between two sets of elements. In the context of an IR system, we can represent each document as a set of words and each query as a set of query terms. Jaccard similarity is calculated as the size of the intersection of the two sets divided by the size of the union of the two sets. Jaccard similarity can be a good choice for IR systems when the frequency of terms in the documents is less important than their presence or absence.
- **Pearson correlation coefficient:**The linear correlation between two vectors is measured by the Pearson correlation coefficient.It is derived by dividing the covariance of the two vectors by the sum of the standard deviations of the two vectors.When the connection between the vectors' dimensions is crucial, the Pearson correlation coefficient might be a useful option for IR systems. [2]

1.9 Accuracy in IR systems

At the surface level, accuracy is not used in an IR system due to class imbalance. Accuracy maybe defined using the following formula:

$$\text{Accuracy} = \frac{\text{No. of relevant words retrieved + no. of irrelevant words NOT retrieved}}{\text{Total no. of words}}$$

In this, the number of irrelevant words not retrieved >> number if relevant words retrieved. Since a typical IR system focuses on retrieving more relevant documents only, the accuracy will not be reflective of the efficiency of the model.

1.10 F_α Score

The F_α measure can be written as follows:

$$F_\alpha = \frac{(1+\alpha^2)PR}{\alpha P + R}$$

Where P is precision and R is recall. This Equation an be transformed in the following manner:

$$F_\alpha = \frac{1}{\frac{\alpha}{1+\alpha^2}\frac{1}{R} + \frac{1}{1+\alpha^2}\frac{1}{P}}$$

From the above equation, we can conclude the recall will get higher weight (for the harmonic mean) when $\alpha > 1$.

1.11 Average Precision @ k

Consider a IR system that retrieves top k documents for a query in the form of a ranked list. Precision directly calculates the ratio of relevant documents retrieved to total retrieved documents. This does not take into account the order in which the documents are retrieved. Average Precision helps solve this issue by returning the average precision value for each document rank from 1 to k. This in turn enhances recall as well.

1.12 Mean Average Precision @ k

While AP considers taking average for each document rank from 1 to k for one query, Mean Average Precision takes it one step further by calculating the average value of average precision across multiple queries.

1.13 AP vs nDCG

In Cranfield dataset, we majorly aim to retrieve all relevant documents. Thus, Average Precision method is best suited evaluation technique. normalised discounted cumulative gain, while considers the importance of highly important documents, has a tendency ti ignore documents lower in the ranking and thus focuses on quality of retrieved documents rather than quantity.

Hence, nDCG is not a suitable metric for cranfield dataset.

1.14 Evaluation Implementation

The listed methods have been implemented and available along with the code.

1.15 AP graph and Observation

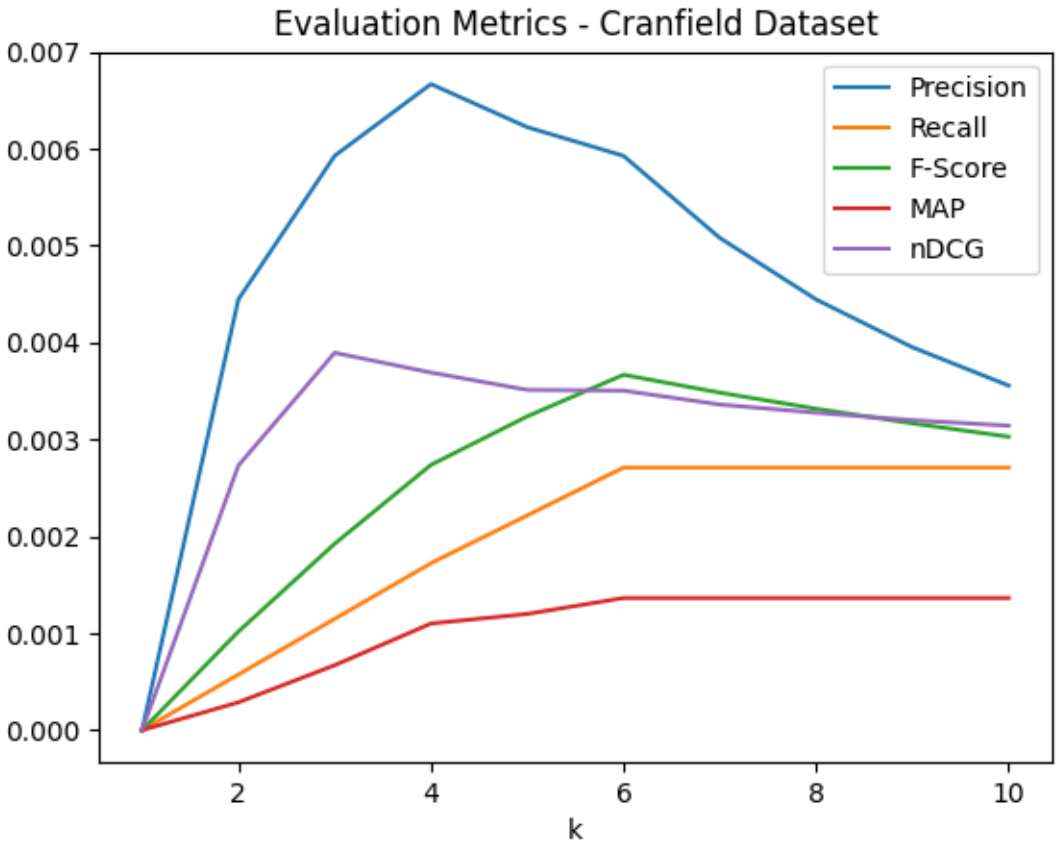

Figure 1: Evaluation curve

From this graph, we can note that the respective evaluation metrices follow the

required trends.

1.16 Queries with unsatisfying results

The queries with non-technical words tend to show dissatisfying results as this form of retrieval system only depends on word structure and not synonyms.

1.17 Shortcomings of Vector Space Model

The biggest shortcoming of the Vector Space model is the fact that it only considers term frequency. This in turn translates into the following challenges:

- **Synonymy and Polysemy:** Since this method only depends on word structure, it may not identify synonyms. For example, it may not recognise that car and automobile are the same thing. Similarly, it may not differentiate words with the same structure but different meanings(for example bank may mean money bank or the river bank).
- **Context not accounted for:** Since context is not accounted for, the results obtained will not be accurate. Following the polysemy example, we can illustrate this shortcoming.

Additionally, the tf matrix has a lot of zero entries where redundant space is occupied.

1.18 Document Title

In the IR system, for representation of the title of the document we can use it as a seperate vector like the rest of the sentences in the document. However, the words in the title will be weighed differently.

If the title is weighed 3 times the body, the words that occur in the title are calculated as occurance of 3 in tf-idf. This in turn increases probability of that document retrieved in case such a word is present in the query.

1.19 Bigram indices

While bigrams go beyond first order word comparison by accounting for relation between words, it increases the data required for storage and also the computation time. Thus, overall this method proves to be disadvantageous.

1.20 Relevance feedback

Relevance feedback is a method used search systems across companies in order to keep the system updated to latest searches. Implicit relevance feedback includes getting feedback from user in a nonintrusive manner without the user's apparent knowledge. A simple example of this would be to record the number of times a retrieved document is referred to, or by recording the time spent reading that document. [3]

Note: apart from the reference mentioned, this topic was discussed in the course EE4371 2 years ago.

2 References

1. "sklearn.feature_extraction.text.TfidfTransformer." Sci-kit learn.
2. K. Pradeep Reddy." Impact of Similarity Measures in Information Retrieval." International Journal of Computational Engineering Research (IJCER), vol. 08, no. 06, 2018, pp. 54-59.
3. Indirect relevance feedback. (n.d.). https://nlp.stanford.edu/IR-book/html/htmledition/indirect-relevance-feedback-1.html