Jordan Rodak
Pallavi Singh

CS276-PA4 Report

Pointwise Approach and Linear Regression(Task 1)

For each query document pair, we have created a five dimensional vector representation of tf-idf scores of url, title, header, body and anchor of a document.We have used Stemming and length normalization to calculate term frequency vector of document field.

TF-IDF vector:

```
Query-idf-vector * Doc-tf-vector(Stemmed + Length normalized)
```

Query-Doc vector:

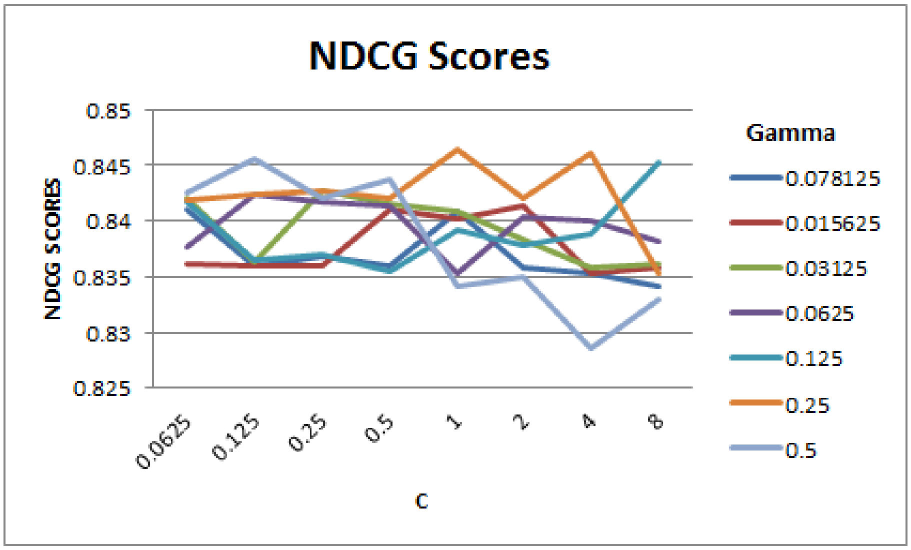
```
[url_Score  title_Score  header_Score  body-score_Score  anchor_Score]
```

Each dimension of vector represents an extracted feature of the train data with relevance score of a document as an added target feature. An instance is created from the features of an document and added to the dataset which is trained for linear regression model. The trained model is then used to learn feature weights and give relevance score on test data which is used to rank documents on test data. Below are the NDCG result scores on training and test data.

	NDCG_training	NDCG_development_test
Task 1- Pointwise	0.87048	0.83972

Pairwise Approach and Ranking SVM(Task 2)

Using cosine similarity as the features of our instance vectors, NDCG score for linear kernel is **0.83573** and **0.84543** for non-linear kernel prior to grid search over values for C and Gamma. The results of the grid search are shown below. The scores reported on are on the development data set.



We can see from this chart that there are a few combinations of values for C and Gamma which result in NDCG scores above 0.845. The highest score achieved was **0.84636** with C = 1.0 and gamma = 0.25.

Our feature vectors are cosine similarities between the document and query for each of the zones of the document. We use the following formula to compute tf-idf for each document : $tf \cdot idf(t,d) = 0.5 + \frac{tf(t,d) * 0.5}{max - tf(d)} \times idf$. We also tried logarithmically scaled tf-idf but this resulted in slightly worse NDCG scores. The dataset is normalized prior to computing difference vectors for training.

More features and Error Analysis(Task 3)

Non-Linear SVM with C = 1.0 and gamma = 0.25, gave us best score on dev data and we tried to improve this score by adding more features to our model. Given are the features that we tried to add and corresponding NDCG results for different combinations are mentioned in the table.

Feature Addition:

1. *BM25F*: We added a feature of BM25F score for a query-doc pair to train. We considered the formula without pagerank since it was giving better score(0.8597) than the one with pagerank(0.8513).
2. *Smallest Window*: We calculated the boost for the smallest window of query words in doc. This gave us a better score
3. *Page Rank* of documents.
4. *URL Relevancy*: The number of query words present in url.

Ablation Test: We performed an ablation test to have an idea which features are useful and we observed that the URL relevancy feature is not very useful (removal increased the score). The removal of other features decreased the score and hence are useful. Hence, we tried different combinations of BM25F, Smallest Window and Pagerank. The progressive results are enlisted in the table below:

	BM25F	**Smallest Window**	**Pagerank**	**URL Relevancy**	**NDCG Score**
Test 1	✓				0.8456
Test 2		✓			0.8457
Test 3			✓		0.8501
Test 4		✓	✓		0.8518
Test 5	✓	✓			0.8567
Test 6	✓	✓	✓	✓	0.8573
Test 7	✓		✓		0.8581
Test 8	**✓**	**✓**	**✓**		**0.8603**

Jordan Rodak
Pallavi Singh

Error Analysis:

1. *Url-PDF document*: As suggested in handout, we observed that some pdf urls are not ranked as expected, Hence we added a binary feature: 1 if url is linked to pdf, 0 otherwise.
   Result:YES useful - A good percentage of pdf documents are ranked correctly and our NDCG score improved by +0.01

2. *Tilde(~) in URL*: A tilde in a URL is usually present in case of a specific document related to a user profile etc. The ranking of these documents were mostly different than optimal ranking. Hence we added a binary feature: 1 if ~ present in URL, 0 otherwise.
   Result: YES useful -This feature helped in correcting the ranking of many profile specific documents (having ~ in url). NDCG score increased by +0.001

3. *URL length*: We observed that many of ranking followed a pattern of longer URLs at top and shorter urls near bottom. Hence, we thought of using the url length as a feature.
   Result:NOT useful - The NDCG score was decreased by -0.01 and hence this feature was not much help.

4. *Title Length*: We applied the same logic of URL length to Title length, because we observed that the documents having larger title length was giving wrong ranking. This was mainly because longer title will have more query words even if it is not much relevant. Hence, we tried to use the title length as a feature to correct this.
   Result: YES useful -This feature increases the score on training data but did not affected the dev score.

5. *URL with ID*: Url text normally consists of an id of a page/profile/document which is indicated by “id=....”. We observed that these documents are not ranked as expected in our ranking. An obvious reason is that these are mapped pages or documents which do not have much information in url. We decided to try this as boolean feature with value 1, if string “id=” is present in a url, 0 otherwise.
   Result: YES useful -This feature helped improve our ranking a little and our NDCG score was improved by a small amount (+0.001).

6. *Exact Query Match*: We noticed that the title and headers which are exact query text are normally up in optimal ranking but ranked differently in our results. So we tried to have a boolean feature, 1 if header/title is exact query match, 0 otherwise.
   Result: NOT useful - It decreased the score by -0.01.

Jordan Rodak
Pallavi Singh

NDCG scores for different error analysis is listed in table. We got the best score with the combination of error features:

`URL-PDF-linked + Tilde-in-URL + Title-length + URL with ID ->` **`0.8718`**

	Url-PDF linked	Tilde in url	URL length	Title length	URL with ID	Query match	NDCG Score
Test 1	✓						**0.8701**
Test 2		✓					**0.8615**
Test 3			✓				0.8525
Test 4				✓			**0.8603**
Test 5					✓		**0.8607**
Test 6						✓	0.8572
Test 7	✓	✓		✓	✓		**0.8718**