**Report | Borgalinov Timur | BS3-DS2**

**Note: All results can be seen in HW2(IR).ipynb file | To run it read readme.docx**

**Part 1:**

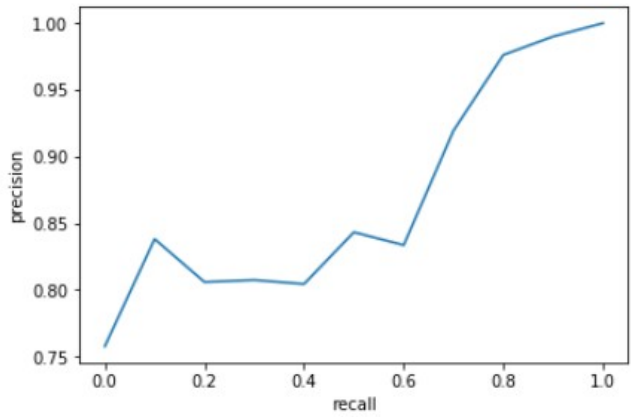
In this part I implemented Rocchio algorithm from book and pseudo-relevance feedback. I used cranfield dataset.

**Results :**

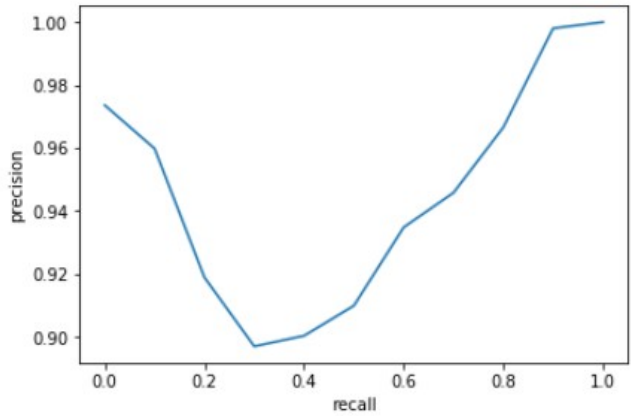
**Rocchio algorithm**
Average cosine similarity between initial and modified by
Rocchio algorithm vectors: 0.939715788987004
**Mean average precision:**
before = 0.6631061042959226
after = 0.8910033404042107
Improvement = 0.22789723610828805
**NDCG score:**
before = 0.4520374153246157
after = 0.6930290561340531
Improvement = 0.24099164080943736

**11-Point Interpolated Average**

Before Rocchio




After Rocchio




```
[ 0.2157958   0.12162776  0.11295362  0.08958729  0.09586451  0.0666206
  0.10119686  0.02633146 -0.00961699  0.00803922  0.        ]  = improvement in each region
```

---

## Pseudo relevance feedback

Average cosine similarity between initial and modified by pseudo algorithm vectors : 0.9476209907692038

**Mean average precision:**

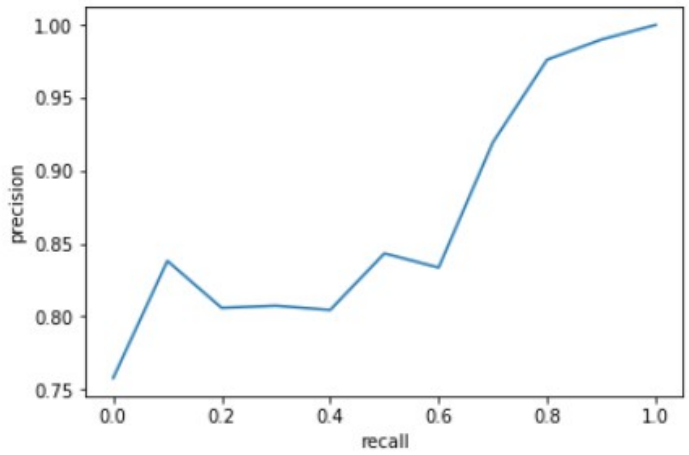
before = 0.6631061042959226
after = 0.6667813237570793
Improvement = 0.0036752194611566757

**NDCG score:**

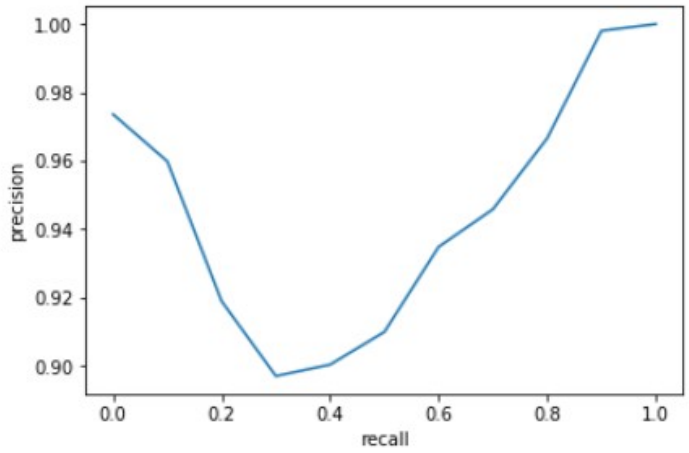
before = 0.4520374153246157
after = 0.4518320111052615
Improvement = -0.0002054042193542216

## 11-Point Interpolated Average

**Before Pseudo Relevant**




**After Pseudo Relevant**




```
[ 0.2157958   0.12162776  0.11295362  0.08958729  0.09586451  0.0666206
  0.10119686  0.02633146 -0.00961699  0.00803922  0.        ]  = improvement in each region
```

**Global method**

Average cosine similarity between initial and modified by pseudo algorithm vectors : 0.9476209907692038

**Mean average precision:**

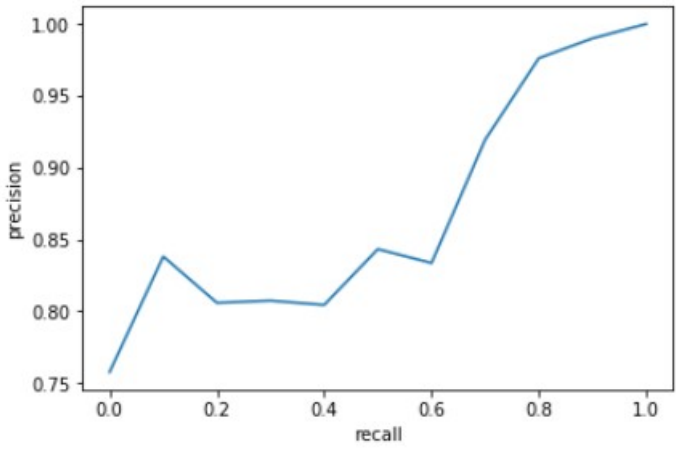
before = 0.6631061042959226
after = 0.637714249038816
Improvement = -0.025391855257106632

**NDCG score:**

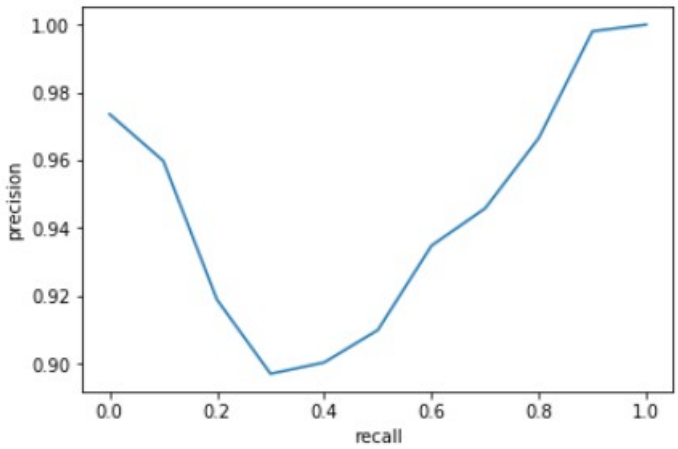
before = 0.4520374153246157
after = 0.4278542345290069
Improvement = -0.024183180795608827

**11-Point Interpolated Average**

**Before Global**




After Global




```
[ 0.2157958   0.12162776  0.11295362  0.08958729  0.09586451  0.0666206
  0.10119686  0.02633146 -0.00961699  0.00803922  0.        ]  = improvement in each region
```

**Part 2**

I implemented two single-document algorithms for summary

Pipeline of the first:

Pipeline of the second:

1) Get document
2) Translate it to sentences
3) Translate every sentence to vector
4) Build cosine similarity matrix between sentences
5) Run page rank algorithm on that matrix
6) Get top_k score sentences

**Results:**

Same document by different methods:

doc id = 184 in Cranfield dataset

**First method:**
automatic programmed control of the tunnel would appear to be necessary . experimental and analytical work is required to check on the validity of these assumptions . it is concluded that complete similarity obtains only when aircraft and model are identical in all respects, including size . an investigation is made of the parameters to be satisfied for thermo-aeroelastic similarity . scale models for thermo-aeroelastic research .

**Second method:**
by limiting consideration to conduction effects, by assuming the major load carrying parts of the structure are in regions where the flow is either entirely laminar, or entirely turbulent, and by assuming a specific relationship between reynolds number and nusselt number, an approach to similarity can be achieved for small scale models .. it is concluded that complete similarity obtains only when aircraft and model are identical in all respects, including size .. it appears that existing hot wind tunnels will not be completely adequate for thermo-aeroelastic work, and accordingly a possible layout for the type of tunnel required is described .. scale models for thermo-aeroelastic research . an investigation is made of the parameters to be satisfied for thermo-aeroelastic similarity .

**Conclusion:**

**1st part**

As you can see there is an improvement in performance near of 0.2 when we are using Rocchio algorithm and small improvement with pseudo relevant, even a little disprovement on NDCG score.
Also there is a disprovement in global method expanding queries by Wordnet by adding synonyms for every token after removing stop words, and lemmatizing. This disprovement comes from generated synonym sentences and their result. We cant compare this two methods because it's obious that global method in general case can only advice some similar queries but not run them automatically. Because before doing it automatically we need to understand the semantic of users query which is pretty complex task to complete.

**2nd part**

In second part both algorithms were too simple to give us good summary about documents. They extract knowledge using only 1 document which restricts it's abilities to represent best info. There are plenty alternatives that use machine learning and deep learning methods to create more useful summary that gives attention to global corpus not only to our document which we want to summarize.