Combining Multiple Evidence from Different Types of Thesaurus for Query Expansion

Sneha Shankar Narayan

December 7, 2014

CS646 - Information Retrieval

Contents

Chapter 1

Background

Query expansion (QE) is the process of reformulating a seed query to improve retrieval performance in information retrieval operations. [1] The given input is evaluated and additional search terms are added to in order to retrieve more relevant documents.

The approach being explored in this project is combining different types of thesaurus for query expansion. Thesauri have frequently been incorporated in information retrieval systems as a device for the recognition of synonymous expressions and linguistic entities that are semantically similar but superficially distinct. [2] This is based off of the idea presented in the paper "Combining multiple evidence from different types of Thesaurus for query expansion" by Rila Mandala, Takenobu Tokunaga, Hozumi Tanaka [2].

The basic idea is to get "related" terms to the query terms from the different thesauri individually and combining them and evaluate how the expanded queries perform as opposed to the original queries.

The various thesauri being used in this project are:

Wordnet based thesaurus

WordNet is a hand-crafted thesaurus developed by Princeton University. In WordNet, words are organized into taxonomies where each node is a set of synonyms (a synset) representing a single sense. There are four different taxonomies based on different parts of speech and also there are many relationships defined among them. [2] The paper uses the noun taxonomy only.

The similarity between words a and b can be defined as the shortest path from each sense of a to each sense of b

$$sim_{path}(a, b) = max[-\log(\frac{N_p}{2D})]$$

where N_p is the maximum number of nodes in path p from a to b and D is the maximum depth of the taxonomy.

Co-occurence based thesaurus

This method is based on the assumption that a pair of words that occur frequently together in the same document are related to the same subject. Therefore word co-occurrence information can be used to identify semantic relationships between words.[2]

The text is subdivided into pseudo-sentences of word-size 3, and each word in the query is compared against the words in the documents to determine co-occurrence. Mutual-information is used as a tool for computing similarity between words. Mutual information compares the probability of the co-occurrence of words a and b with the independent probabilities of occurrence of a and b:

$$I(a,b) = \log \frac{P(a,b)}{P(a)P(b)}$$

where the probabilities of $P(a)$ and $P(b)$ are estimated by counting the number of occurrences of a and b in the document. The joint probability is estimated by counting the number of times that word a co-occurs with b

Combined approach

The similarities from the above two approaches are averaged to calculate the similarities between two words a and b in the combined approach

$$sim_{combined}(a,b) = \frac{sim_{wordnet} + sim_{co-occurence}}{2}$$

Since the similarities can be of an range they are normalized before being plugged into the above formula. For all similarities that are computed in the wordnet and co-occurence approaches the following formula is applied

$$sim_{new} = \frac{sim_{old} - sim_{min}}{sim_{max} - sim_{min}}$$

Chapter 2

Evaluation Methodology

The original queries are expanded using the terms determined by each of the three approaches. The queries are run on Galago [**?**] and the new ranked lists are obtained.

The determined ranked lists are checked against the baseline (the experiments performed as a part of P2) and the following measures of interest are obtained

- Average precision and Mean average precision
- NDCG @ 10
- Precision @ 10

The queries in the books-medium collection, the queries for the robust collection from the class, and the TRECRobust collection were used to perform the experiments.

Chapter 3

Implementation

The queries that were obtained from the various collections had to be expanded and the implementation of the query expansion in each type of thesaurus was done using Python.

In order to help with the implementation, all the word-similarity computation was done with the query words and the words in the top 50 ranked documents of the original query. Thus query expansion was basically done on the fly.

Wordnet based thesaurus

The wordnet thesaurus was made available to python using textblob [3] a text processing library for python. First, the top 50 documents were retrieved for the original query and the wordnet path similarity between every query word and the other words were computed.

The similarity threshold of 0.5 was applied to determine the similar words and for each query word the top 3 similar words were added to the query as long as the similarity threshold was met.

Co-occurence thesaurus

The top 50 documents for the original query was retrieved. Counters are managed for the number of occurences of each of the query words in the retrieved documents. Hashes are managed to get the most co-occuring words with the query words in the documents revtrived. Mutual information values are computed for the co-occuring words.

The similarity threshold (mutual information values) of 0.5 was applied to determine the similar words and for each query word the top 2 similar words were added to the query as long as the similarity threshold was met.

Combined approach

The similarities obtained from the above two methods were averaged and the set of words that were determined from both the approaches were added to the original query in order to expand it.

Process

The query expansion and evaluation on each of the approaches had the following steps

1. Parse the queries.xml file in all the datasets and send each query to the script (of the method being evaluated) which does the query expansion.
2. The obtained query along with the query number which is got from the XML file is output to a json file.
3. Run the expanded queries on Galago and get the ranked list.
4. Evaluate the expanded queries against the relevance judgements using the evaluation scripts that were written for P2.

Chapter 4

Results and Analysis

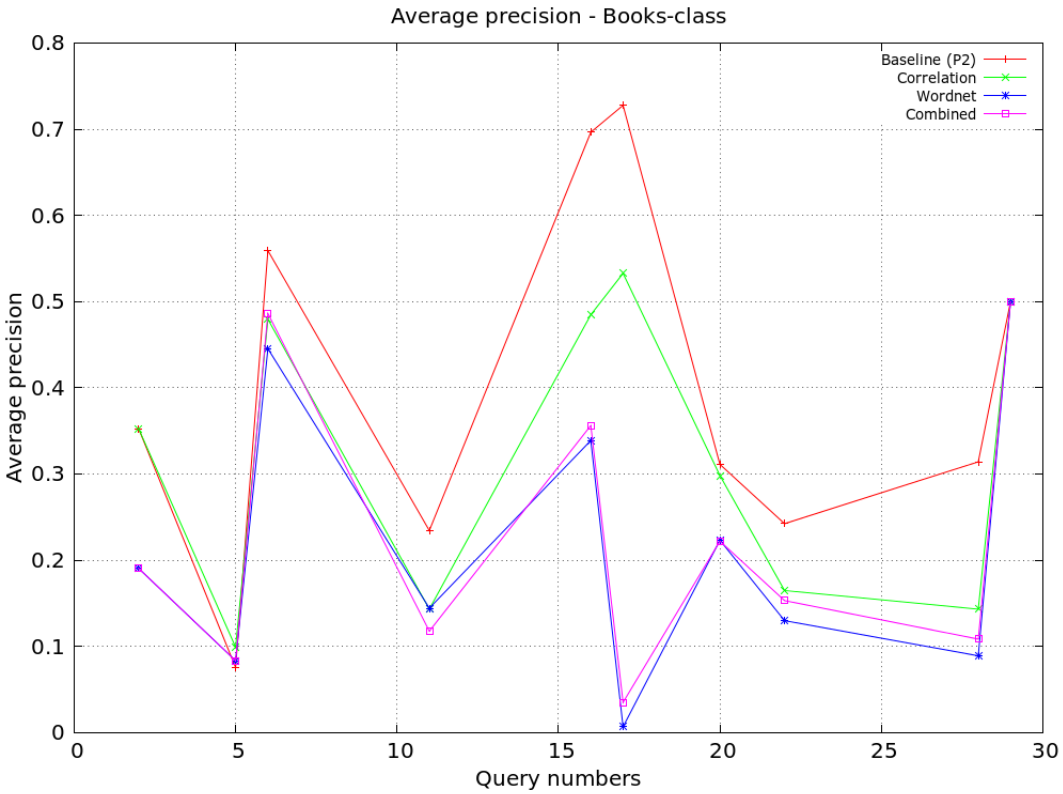
The evaluation measures computed are the following:

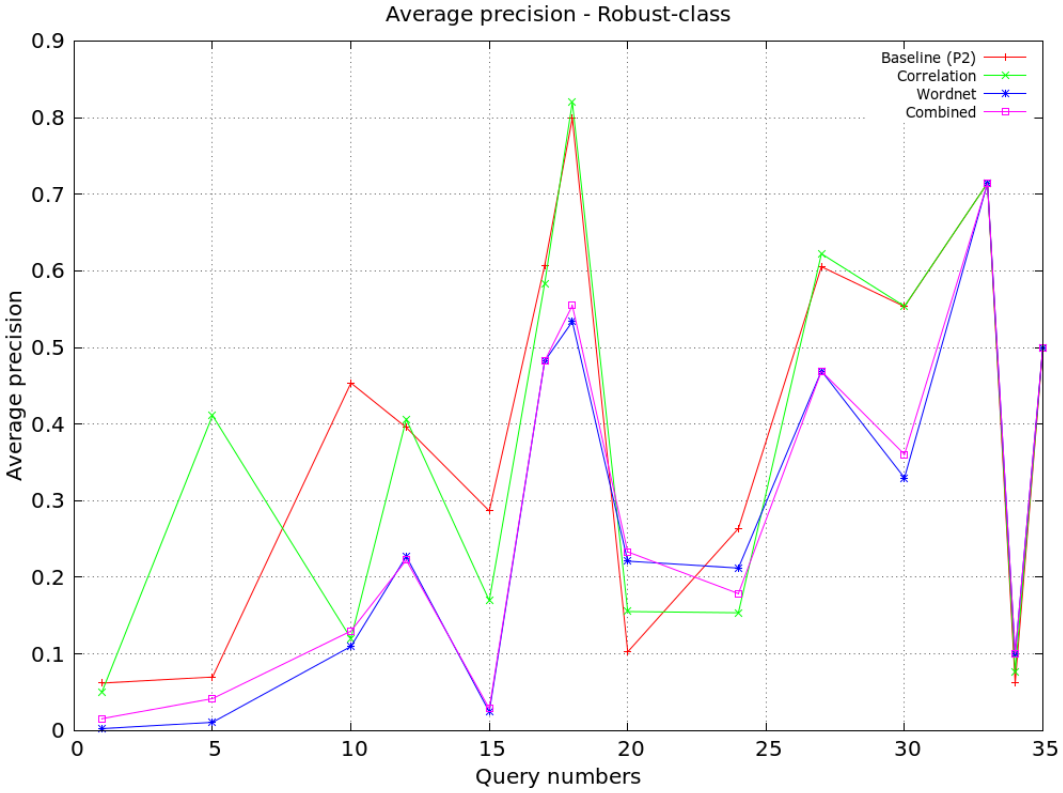
Mean average precision

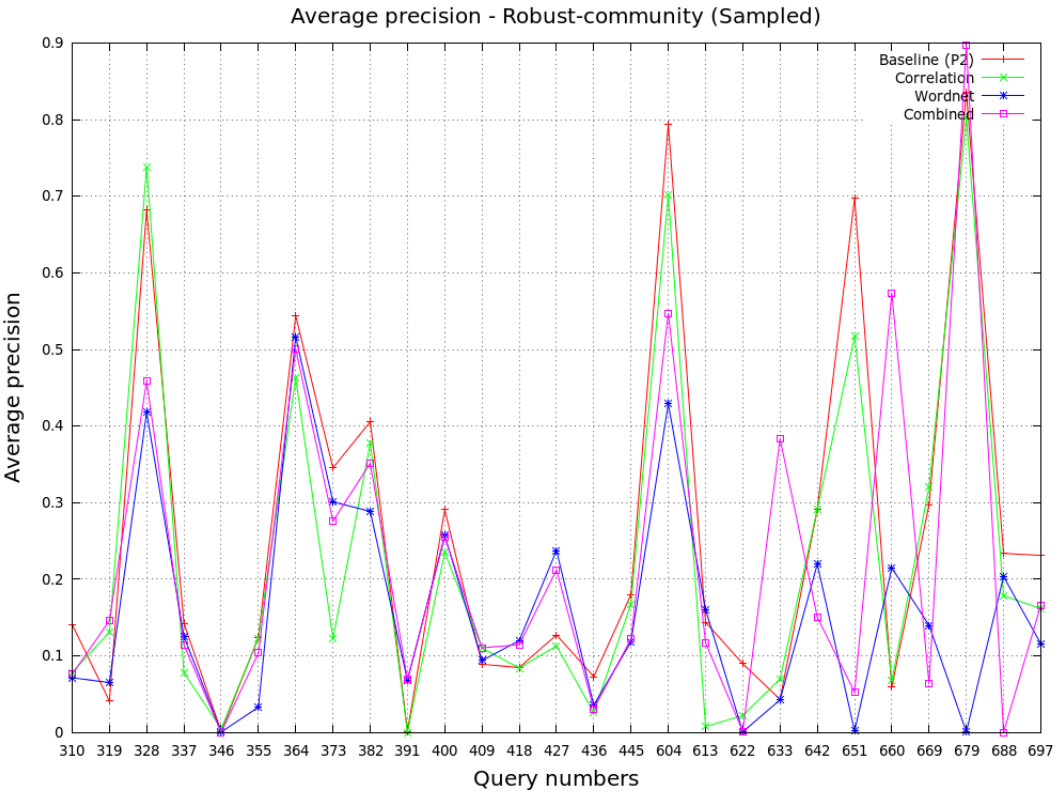
	Baseline (P2)	Wordnet Thesaurus	Co-occurence Thesaurus	Combined Approach
Books	0.3647	0.1954 (-46%)	0.2906(-20.3%)	0.2047 (-43.84%)
Robust-class	0.3910	0.2812 (-28%)	0.3811 (-2 %)	0.2879 (-26.3%)
Robust-community	0.2295	0.1549 (-32%)	0.1843 (-19.69 %)	0.1582 (-31.06%)

Graphs

For easier analysis of the obtained results, it's interesting to see how the average precision works across the data sets:


Average precision - Books-class
Baseline (P2)
Correlation
Wordnet
Combined
Average precision
Query numbers
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0
5
10
15
20
25
30


Average precision - Robust-class
Baseline (P2)
Correlation
Wordnet
Combined
Average precision
Query numbers
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
0
5
10
15
20
25
30
35


Average precision - Robust-community (Sampled)
Average precision
Query numbers
Baseline (P2)
Correlation
Wordnet
Combined
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
310 319 328 337 346 355 364 373 382 391 400 409 418 427 436 445 604 613 622 633 642 651 660 669 679 688 697

Chapter 5

Conclusion

Bibliography

[1] http://en.wikipedia.org/wiki/Query_expansion

[2] http://www.iro.umontreal.ca/~nie/IFT6255/mandala99combining.pdf

[3] http://textblob.readthedocs.org/en/dev/