PRIS at TREC2011 Micro-blog Track

Yan Li, Zhenhua Zhang, Wenlong Lv, Qianlong Xie, Yuhang Lin, Rao Xu
Weiran Xu, Guang Chen, Jun Guo
School of Information and Communication Engineering,
Beijing University of Posts and Telecommunications
Beijing, P.R. China, 100876
buptly@yahoo.com.cn

Abstract

Our system to Micro-blog Track at TREC2011 is described in this paper, which includes data obtaining and preprocessing, index building and query expansion. There’re two methods of query expansion introduced in this report: Word Activation Force algorithm (WAF) and Electric Resistance Network. We also show the evaluation results for our team and the comparison with the best and median evaluations.

1. Introduction

The Micro-blog Track examines search tasks and evaluation methodologies for information seeking behaviors in micro-blogging environments. This year is the first year of the Micro-blog Track, which aims at addressing a search task whereby a user's information need is represented by a query at a specific time. In particular, it is a real-time search task, where the user wishes to see the most recent but relevant information to the query. Hence, the system should answer a query by providing a list of relevant tweets ordered chronologically. It is expected that when selecting tweets to include in the list, the "interesting" but "newer" relevant tweets should be paid more attention to. Interestingness is subjective, but the issuer of a query might interpret it as providing somehow added value with respect to the query topic. For this year, the "novelty" between tweets is not considered.

2. Dataset and Preprocessing

In Micro-blog Track at TREC 2011, theTweets2011 corpus is provided officially, and we also downloaded the web pages linked from the tweets as extra corpus.

2.1 The Tweet2011 Corpus

The Tweet2011 corpus is obtainable with the official downloader, provided that the network environment guarantees a stable access to Twitter.com. When a tweet is finished, a status code is generated as a result. There’re five types of codes in the corpus, 200, 302, 403, 404 and null, which means ok, found, forbidden, not found and nothing respectively. In practice, the number and status of the tweets differs according to the network environment, downloading time and other possible reasons. In our case, the statistics of the corpus fetched via the corpus downloader are shown as follow:

Table 1. The statistics of our team's corpus

200	302	403	404	Null	total
13,979,849	1,114,483	239,935	700,435	1,006,050	16,034,705

2.2 Web Pages Linked from the Tweets

Due to the limited length of the tweet text, it fails to provide adequate information. We downloaded the URL links extracted from tweets to obtain extended content. The total number of tweets with one or more URLs was 2,768,878, in which 65109 were non-English. Even though there were considerable numbers of tweets that share the same URLs, especially in the case of "re-tweet", we reserved the duplicate links considering that it might indicate the popularity or other properties of relate tweets. Eventually, 1,659,097 web documents were successfully crawled from the internet.

2.3 Tweets Pre-Processing

Two pre-processing tasks were performed in our system: RT tweets removal and non-English tweets removal.

We first extracted user remarks and remove the label "RT". The "new style" re-tweets to which the HTTP crawler returned 302 were all removed.

When URLs and punctuation were removed, each tweet was judged to be English or non-English with the help of an English vocabulary word list, Alan Beale's Core Vocabulary. A tweet with more than a half English words would be left and used for retrieval task while tweets with any non-English content were rejected in the query expansion task.

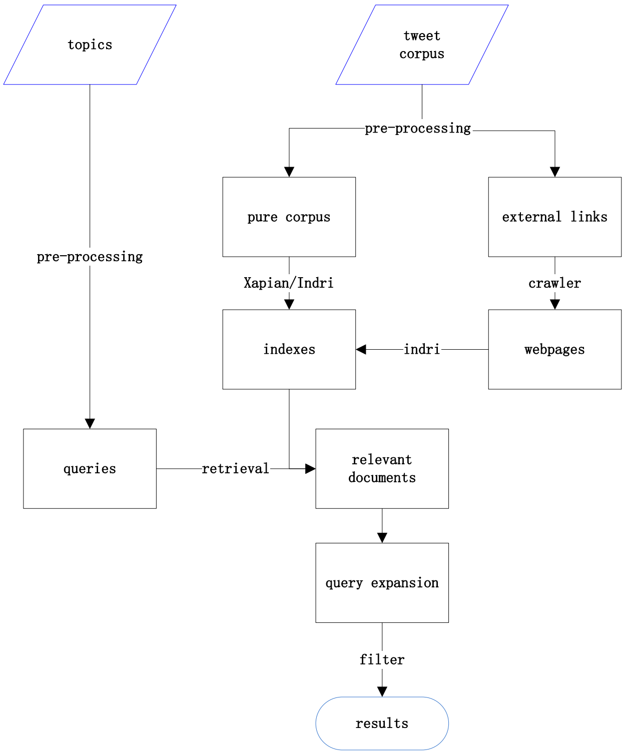


Figure 1. The structure of our system

3. Retrieval Model

3.1 System Introduction

To build a real-time system for this year's Micro-blog Track search task, the system we implemented was designed based on the structure of an information retrieval system as the Fig.1 shows.

As shown in Fig.1, the corpus of tweets was downloaded by official crawler while another dataset of web pages, whose links are provided in tweets, are fetched by a self-designed crawler. Then we extracted the relevant tweets according to the relevance score between tweets and the queries calculated based on the title of the topics. In the third part of our system, query expansion were applied for every topic. Finally, we re-evaluated the relevance between the tweets retrieved in the second step and the expanded topics, filtering out the tweets under the relevance threshold, and then re-sorted the tweets in chronological order.

3.2 Baseline: Relevant Documents Retrieval

As the tweets corpus contains almost all the tweets posted from Jan 24 to Feb 8, most of the tweets are not relevant to the topics.

Firstly, we filtered out the re-tweets from the original tweet corpus, creating a new corpus NO302-Dataset, and building indexes for both the NO302-Dataset and the original corpus With302-Dataset.

Secondly, we manually extracted query words of each topic by filtering out stop words and expanded them using their synonyms. For example, the word US was expanded by USA and America.

Finally, the Xapian and Indri Toolsets were applied as our indexing toolsets. Tweets in No302-Dataset containing any of the keywords of a topic and posted before the query timestamp were treated relevant to the topic and would be retrieved for re-ranking later. While tweets in With302-Dataset containing all of the keywords of a topic were treated highly relevant to it and would be retrieved for expansion later.

In practice, we found that tweets ranked after 1000 in the ranking list were irrelevant to the topics though containing some keywords, so we made 1000 as the threshold of the number of results.

As to the webpage dataset, the process was similar and the difference was that only Indri Toolset was used and the threshold was set to 500.

We regarded the retrieval results as our baselines of the system.

4. Query Expansion

In this stage, we are expected to mine the words that have strong connection with a given topic so as to improve document retrieval performance with more adequate information. Two algorithms were applied in this stage: the Word Activation Force algorithm and Term Similarity Metric method.

4.1 Word Activation Force Algorithm

The Word Activation Force algorithm (WAF) is based on the assumption that there's a special force in documents helping human brains activate associates of a word, such as ‘hospital’ activates strongly ‘doctor’ or ‘nurse’. It believes that there are latent structures of word network in documents. The WAF proposes an effective approach mapping syntactical and semantic information into sparse directed networks, comprehensively highlighting the features of individual word. Based on the directed networks, sensible word clusters and hierarchies can be efficiently discovered.

For the Micro-blog TREC, WAF was applied to unearth extra keywords of a given topic to improve outcomes. To begin with, the top 500 most relevant web documents in the baseline were selected for each topic. Note that, the set of ranked documents for each topic is independent and we name it basic set.

The documents were stemmed and turned into lower-case letters in the first place. Then words occurrence and co-occurrence were calculated in the basic set. We use the follow annotations:

- f_i ,the frequency of word i in the basic set;
- f_{ij} ,the co-occurrence of word i to word j in the basic set, which indicates the frequencies of pairs (i,j) where i precedes j by up to L words(L =4 in our study);
- d_{ij}, the average word distance between word i and word j.

Then the word activation force of word i to word j, or waf_{ij}, can be calculated as follows:

$$\mathrm{waf}_{ij} = \frac{f_{ij}\, f_{ji}}{f_i\, f_j} \tag{1}$$

It is obvious that all the element values in the WAF matrix is between 0 and 1. Zero means that word i is never followed by word j within our word window in the basic set, while one means that word i and j are always adjacent like a compound(f_{ij}=f_j=f_i, d_{ij}=1)

With the WAF Matrix above, we can calculate the closeness of word i and j, namely affinity, as follows:

$$A_{ij}^{waf} = [\frac{1}{|K_{ij}|} \sum_{k \in K_{ij}} \mathrm{OR}(\mathrm{waf}_{ki}, \mathrm{waf}_{kj}) \cdot \frac{1}{|L_{ij}|} \sum_{l \in L_{ij}} \mathrm{OR}(\mathrm{waf}_{il}, \mathrm{waf}_{jl})]^{1/2} \tag{2}$$

where K_{ij} = {k|waf_{ki}>0 or waf_{kj}>0} and Lij = {l|waf_{il}>0 or waf_{jl}>0}. And OR(x,y) = min(x,y)/max(x,y). The Affinity Matrix enables us to discover the association between words in the basic set.

We define Q, W, S as the set of all words in basic set, the set of keywords, and the set of stop words respectively, where stop words were not taken into account in the first two sets. For each word i in Q, and word j in W but not in S, we selected valuable expansion keywords by the Aijwaf measure, assuming that high relevant words would have larger affinity value.

Apart from the affinity measure, we implied “topic frequency” to eliminate bad expansion words based on the assumption that words with high topic frequency is usually less discriminating.

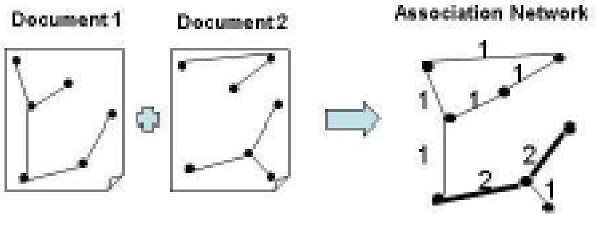


Figure 2. Illustration of association network

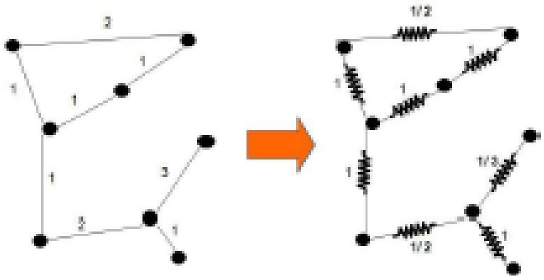


Figure 3. Illustration of resistance network

4.2 Term Similarity Metric Based on Electric Resistance Network

Besides WAF, the term similarity metric derived from the electric resistance network was also applied to query expansion. The distances between vertices were calculated through an undirected weighted graph. And we used the tweets retrieved from Indri as the corpus.

Step 1: Building an association network from related tweets

We built an undirected weighed graph G = (V;E;w) where nodes V represent terms in tweets, edges E represent the associated pair, and weight w on the edges measures the strength of association between two connected nodes.

If two terms co–occur in one tweet, a direct link is built between them. If these two terms co–occur in n (n>0) different tweets, the weight w is n, as is shown in Fig.2.

The association between any two terms is calculated considering all association paths and cumulative weights connecting them.

Step 2: Calculating the effective resistances

The electric resistance network can be used to calculate the effective resistances between any two nodes in the association network built in step 2. The weight w_{jk} between node j and node k was calculated according to the electric conductance c_{jk} defined in the original weighted graph: $r_{jk} = 1/c_{jk} = 1/w_{jk}$.

Fig.3 illustrates the resistance network obtained from the weighted association network.

For all possible pairs, we calculated the resistances with the help of Laplacian Graph L and L = A-D, where A is the adjacency matrix and D is the degree matrix of the graph. Then the effective resistance between node v_j and node v_k can be calculated as follows:

$$r_{jk} = L^{+}_{jj} + L^{+}_{kk} - L^{+}_{jk} - L^{+}_{kj} \tag{3}$$

where L^{+} represents the pseudo-inverse of L.

Step 3: Query expansion with the distance metric

We extend the Effective Resistances calculation to the term space to define the distance in between a target term t and a set of terms S.

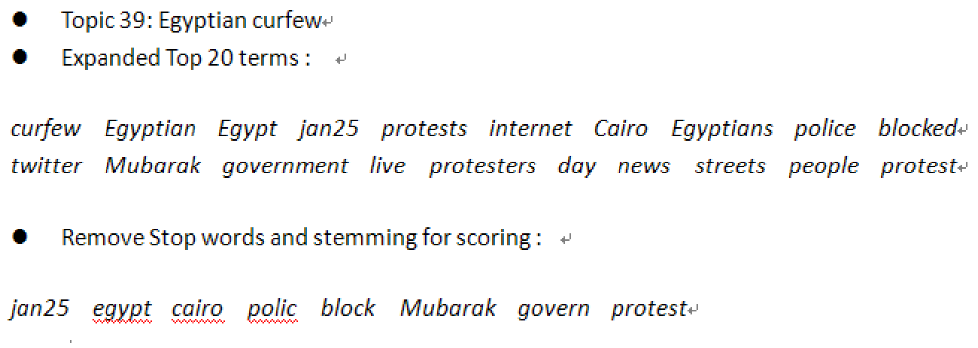
The definition of the distance between a target term t and a term set S is as follows:

$$r_{S,t} = \frac{1}{|S|}\sum_{s_i \in S} r_{s_i,t} \tag{4}$$

where r_{ij} is the effective resistance of node i and node j.

As for the query, we define Q and X as query term set and corpus term set respectively. And for a target term x, its normalized distance to Q can be calculated as follows:

$$r^{\mathrm{norm}}_{Q,x} = \frac{r_{Q,x}}{\frac{1}{|X-Q|}\sum_{y \in X-Q} r_{x,y}} \tag{5}$$



- Topic 39: Egyptian curfew
- Expanded Top 20 terms :

curfew Egyptian Egypt jan25 protests internet Cairo Egyptians police blocked
twitter Mubarak government live protesters day news streets people protest

- Remove Stop words and stemming for scoring :

jan25 egypt cairo polic block Mubarak govern protest

Figure 4. An example of query expansion

With the above metric, all relevant terms to the original queries can be found. And we selected the top 20 terms as highly relevant expansion terms for the next scoring step. Fig.4 shows an example of our query expansion result.

5. Scoring and Ranking

The relevance of a tweet to a certain topic is evaluated separately. For each topic, the score of a tweet t can be calculated as follows:

$$\mathrm{score_t} = \mathrm{a} \cdot \frac{\mathrm{N_q}}{\mathrm{L_q}} + \mathrm{b} \cdot \frac{\mathrm{N_e}}{\mathrm{L_{eq}}} + \mathrm{c} \cdot \frac{\mathrm{N_s}}{\mathrm{L}} + \mathrm{d} \cdot \frac{\mathrm{L}}{\mathrm{L_{max}}} + \frac{\mathrm{HasURL}}{\mathrm{N_s}} \qquad (6)$$

where N_q is the number of words which are contained both in the topic and t, L_q is the number of key words in topic. Similarly, N_e is the number of extension words that contained in t, L_{eq} is the number of words in the expanded terms. L and N_s represent the number of words in t with and without stop words respectively. Lmax is the maximum number of words of tweets for the current topic, and HasURL is a boolean variable indicating whether t contains URLs (used only in our run1 and run3). In addition, a, b, c and d are parameters, which were set to 3, 1, 0.3 and 0.5 respectively.

Obviously, a tweet containing more keywords in the topic is more likely a relevant tweet and should be given a higher score. The score of Ns/L and L/Lmax shows how informative a tweet is, while HasURL/Ns represents the potential information.

Then we ranked tweets for each topic according to the score in the descending order. Finally, we chose top n tweets in the ranking list as the relevant tweets.

6. Evaluation Results

In this year's TREC Micro-blog Track, we submitted 4 versions of runs. And each run is different from another in three aspects as shown in Tab.2.

Table 2. Four runs of our team

Run Id	Future Evidence	External Evidence	Threshold Selection
PRISrun1	no	no	manually
PRISrun2	no	yes	automatically
PRISrun3	yes	no	manually
PRISrun4	yes	yes	automatically

Table 3. Evaluation results of our runs

		MAP	R-Prec	bpref	P@30
allrel	**PRISrun1**	0.3350	0.4019	0.3739	**0.4388**
	PRISrun2	**0.4000**	**0.4940**	**0.4600**	0.4347
	PRISrun3	0.2621	0.3240	0.3002	0.3612
	PRISrun4	0.2914	0.3570	0.3272	0.3721
	baseline	0.1411	0.1486	0.1827	0.0986
highrel	**PRISrun1**	0.2690	0.2827	0.5506	0.1455
	PRISrun2	**0.3131**	**0.3055**	**0.7156**	**0.1677**
	PRISrun3	0.2145	0.2333	0.5465	0.1323
	PRISrun4	0.2379	0.2389	0.6620	0.1384

For each run, the using of future and external evidence affected the range of mining corpus and the expanded words. All retrieved tweets were evaluated by relevance scores and a threshold was set to select highly relevant tweets. We can fix the threshold value manually or automatically. The manual way means that we set the cut-off value by observation, while for the automatic way, the value are worked out by some parameters that were configured automatically according to the features of expanded words. In addition, the runs using external evidences combined scored relevant tweets with tweets that offer highly relevant web pages even if the tweet itself may be irrelevant to the query. All the returned tweets for each run were sorted chronologically.

Tab.3 shows the evaluation results of the four runs. The topic 50 was dropped from the evaluation for it did not have any relevant tweets. The ‘allrel’ is the evaluation for the remaining 49 topics, while the ‘highrel’ is for the 33 topics having highly relevant tweets. We also list the baseline result provided by TREC, and it is obvious that all our four runs outweigh over the baseline significantly. It can also be concluded that the PRISrun2 is our best run.

We also compare our results of PRISrun2 with the best and median results of the track in Fig.5 and Fig.6. In both ‘allrel’ set and ‘highrel’ set, it is obvious that our results outperform the median almost on every topic and even reach the best on some topics.

7. References

[1] http://sites.google.com/site/trecmicroblogtrack/

[2] http://www.manythings.org/vocabulary/lists/l/

[3] Guo, J., Guo, H. & Wang, Z. An Activation Force-based Affinity Measure for Analyzing Complex Networks. Sci. Rep. 1, 113; DOI:10.1038/srep00113 (2011).

[4] Shuguang Wang and Milos Hauskrecht. Effective Query Expansion with the Resistance Distance Based Term Similarity Metric. In proceedings of the 33th ACM SIGIR conference, pages 715-716. ACM, 2010.

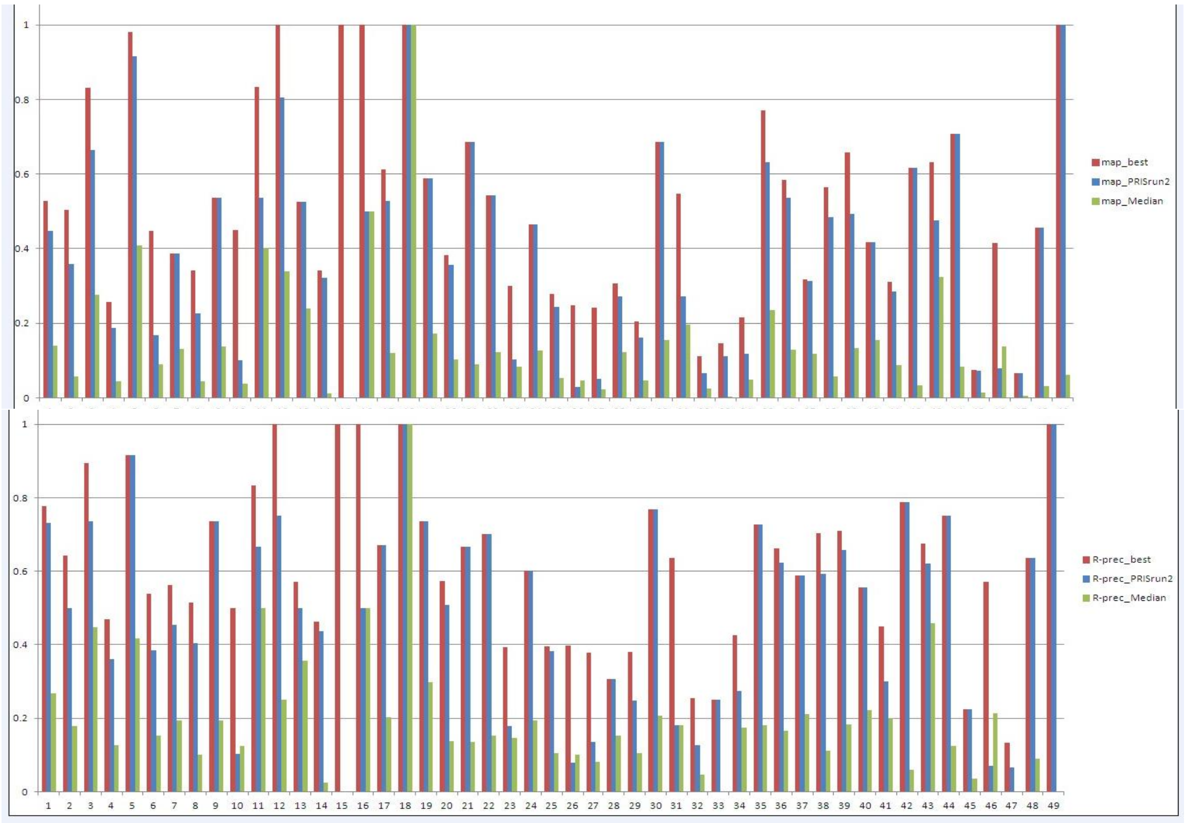


Figure 5. Comparison with the best and median on MAP and R-Prec for PRISrun2.allrel

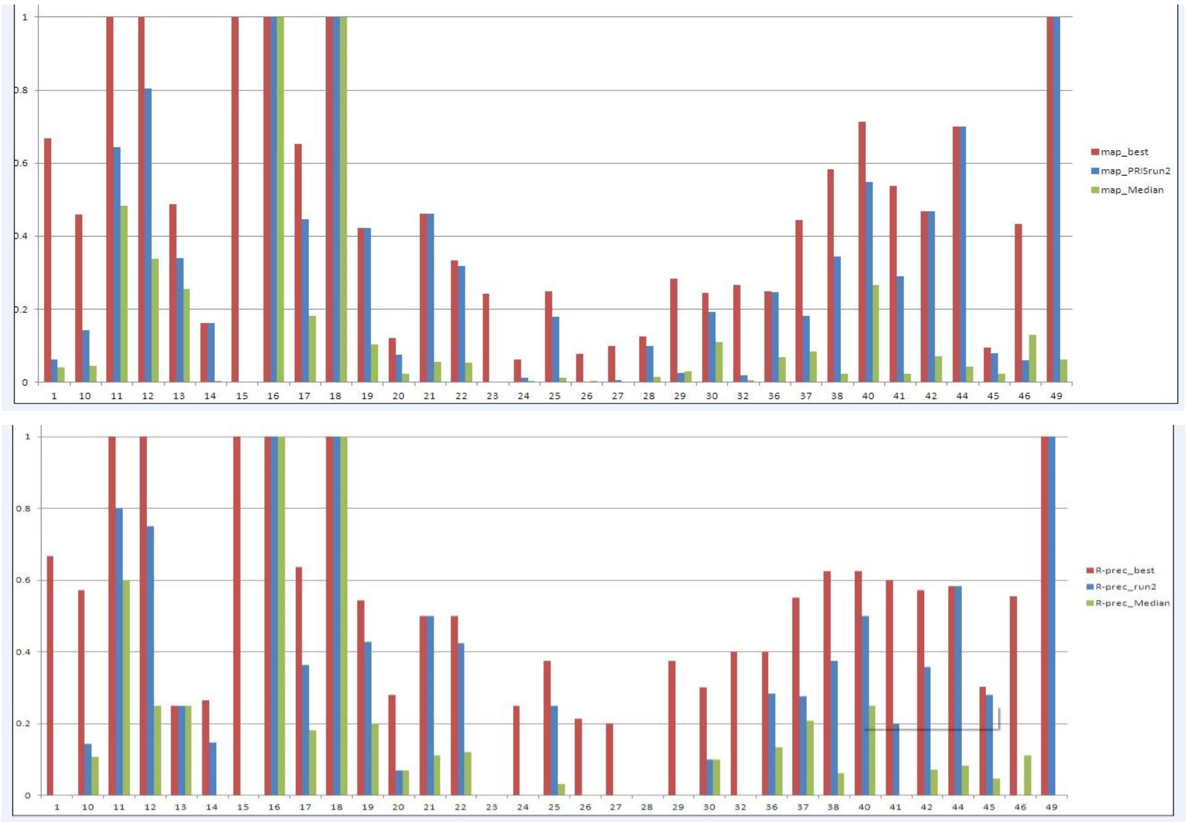


Figure 6. Comparison with the best and median on MAP and R-Prec for PRISrun2.highrel