The University of Iowa at CLEF 2014: eHealth Task 3

Chao Yang, Sanmitra Bhattacharya, and Padmini Srinivasan

Department of Computer Science,
University of Iowa, Iowa City, IA, USA
{chao-yang,sanmitra-bhattacharya,padmini-srinivasan}@uiowa.edu

Abstract. The task 3 of CLEF eHealth Evaluation lab aims to help laypeople get more accurate information from health related documents. In this task, we did several experiments and tried different technologies to improve the retrieval performance. We tried to clean the original dataset and did sentence level retrieval. We explored different parameter settings for pseudo relevance feedback. Description and Narrative was utilized to expand the query as well. We also modified Markov Random Field (MRF) model to expand the query using medical phrase only. In our training set (2013 test set), using those methods can significantly improve the retrieval performance by 8-15% from baseline. We submitted 4 runs. Results on 2014 test set suggest that the technologies we used except MRF have the potential to improve the performance for the top 5 retrieved results.

Keywords: Information Retrieval; Query Expansion; Pseudo Relevance Feedback; Markov Random Field

1 Introduction

The ShARe/CLEF eHealth Evaluation Lab[3] is part of CLEF 2014 Conference and Labs of the Evaluation Forum[1]. It aims to help laypeople understand health related documents better. We participated in Task 3: User-centred health information retrieval[2]. Its goal is to develop a more accurate retrieval strategy for health related documents. Specifically, participants were required to submit a list of relevant health related document ids for each query (topic). In 2014, Task 3 includes a monolingual IR task (Task 3a) and a multilingual IR task (Task 3b). We participated in Task 3a only.

In particular we asked questions like:

1) Does sentence splitting on documents help improve retrieval performance?

2) How does one optimize the parameters for pseudo relevance feedback?

3) Is query expansion using descriptions and narratives more effective than using titles only?

3) Can we include medical phrase detection to make a better Markov random field (MRF) model?

[1] http://clef2014.clef-initiative.eu

2 Dataset

The dataset for Task 3 is provided by Khresmoi project[2]. It has a set of medical-related documents in HTML format. The documents are from well-known health and medical sites and databases. The size of dataset is about 41G (uncompressed), it has 1,103,450 documents.

2.1 HTML to Text

Since the format of the documents is HTML, it has a lot of HTML tags and other noises which may affect the retrieval performance if we index them directly. We employed Lynx[3], a command line browser to convert the HTML files to text only. The size of the text only dataset decreased to 8.6G. Then we replaced frequent UTF-8 broken characters[4]. We named this text only dataset "All_Text".

2.2 Content Cleaning

The ideal text we extract should be the main article from the webpage. However, there are different sections in a typical webpage. The sections could be the structure information about the website, contact information, headlines, even advertisement. The example in Figure 1 shows the beginning of one text output from Lynx. Except for the last two lines, all the information is unrelated to the main article.

```
* Home
* About
* Ask A Question
* Attract CME

Attract
NPHS Logo
Search Clinical Questions Enter search details Search
A total of 1713 clinical questions available
Quick Guide to ATTRACT

What is the evidence for betamethasone cream versus circumcision in phimosis?
Associated tags:child health, men's health, circumcision, phimosis, treatment, corticos-
teroid
...
```

Fig. 1. Example Output From Lynx

[2] http://clefehealth2014.dcu.ie/task-3/dataset
[3] http://lynx.browser.org
[4] http://www.i18nqa.com/debug/utf8-debug.html

However, to remove all those irrelevant information is not trivial. In order to keep the main article only, we tried to use simple rules to remove the headlines, titles. In particular, we removed all the lines which have less and equal than 3 tokens. We also removed all the lines which start with either '*', '+', '-', 'o', '#', and '@'. Those are the headline start symbols from Lynx.

After the data cleaning mentioned above, the dataset we have is about 5.4G. We name this collection "Text_Clean".

2.3 Sentence Splitting

Besides indexing whole documents, we also explored sentence level retrieval. We used GENIA Sentence Splitter (GeniaSS) [6] to split sentences of each text document from "All_Text". This sentence splitter is optimized for the biomedical documents and has good performance. Keeping track of the original text document id we created 3 sentence level datasets: "Sent_1", "Sent_2", and "Sent_3".

"Sent_1" has only single sentences. (In other words, we treat each sentence as a logical 'document'.)

"Sent_2" has pairs of adjacent sentences.

"Sent_3" has sequences of 3 adjacent sentences.

2.4 Training Topic Set

We did not use the training topics provided in CLEF eHealth 2014 because there were only 5 topics and the coverage of qrels file is small. Therefore, we used CLEF eHealth 2013 test topics as our training topics. The 2013 test set has 50 topics. Figure 2.4 shows an example training topic.

```
<query>
<id>qtest1</id>
<discharge_summary>00098-016139-DISCHARGE_SUMMARY.txt
</discharge_summary>
<title>Hypothyreoidism</title>
<desc>What is hypothyreoidism</desc>
<narr>description of what type of disease hypothyreoidism is</narr>
<profile>A forty year old woman, who seeks information about her condition</profile>
</query>
```

Fig. 2. Example Training Topic

3 Baseline

To find out our baseline strategy we created separate indexes from different datasets ("All_Text", "Text_Clean", "Sent_1", "Sent_2" and "Sent_3") using Indri [7]. We filtered out stopwords during indexing and in the queries. We ran Indri's Query Likelihood model used title only as query to retrieve documents from different indexes and the one with best performance is our baseline. For instance, the query for the example in Section 2.4 is "#combine(Hypothyreoidism)"

The evaluation focused on P@5, P@10, NDCG@5, and NDCG@10. These results including MAP are shown in Table 1. We also include the baselines and the best performing runs in 2013. Scores bolded are the best for that measure in the table.

Table 1. Baselines and 2013 Best Runs on 2013 Test Set (Our Training Set)

Run ID	P@5	P@10	NDCG@5	NDCG@10	MAP
Title_All_Text	0.4840	0.4760	**0.4764**	**0.4811**	0.2350
Title_Text_Clean	0.4520	0.4560	0.4583	0.4680	0.2028
Title_Sent_1	0.2040	0.1960	0.2219	0.2114	0.2040
Title_Sent_2	0.2040	0.2000	0.2178	0.2108	0.0964
Title_Sent_3	0.1880	0.1740	0.1921	0.1818	0.0863
BM25	0.4520	0.4700	0.3979	0.4169	0.3043
BM25_FB	0.4840	0.4860	0.4205	0.4328	0.2945
Mayo2	0.4960	**0.5180**	0.4391	0.4665	**0.3108**
Mayo3	**0.5280**	0.4880	0.4742	0.4584	0.2900

Again, Title_All_Text is the retrieval strategy using title as query and All_Text as index which mentioned before. BM25 and BM25_FB (with Pseudo Relevance Feedback) are the official baselines in 2013. The two official baselines only use title as query. (The same strategy with Title_All_Text.) Mayo2 and Mayo3 are the best 2 runs last year from Zhu et al. at Mayo Clinic[8]. Our Title_All_Text is better than BM25 in all the measures, it could have benefited from using Lynx to output text format. It even outperforms Mayo2 and Mayo3 in terms of NDCG@5 and NDCG@10 (but not in P@5, P@10 or MAP). However, using title only to retrieve from Text_Clean and Sent_1/2/3 indexes did not improve the performance. Especially for using Sent_1/2/3, the performance for all the measures dropped significantly.

Therefore, we use Title_All_Text as the baseline for the later experiments. We drop the Text_Clean and the three sentence level datasets since these do not improve retrieval performance.

4 Optimize Pseudo Relevance Feedback

Pseudo Relevance Feedback is a popular and successful method for expanding queries. We can see in Table 1, the official baseline BM25_FB outperforms

BM25 in almost all of the measures. We tried to improve on our baseline results with Title_All_Text by optimizing the parameters of Pseudo Relevance Feedback (Lavrenko's relevance models [4]) using Indri. There are 3 parameters that need to be set. The first is the weight of original query (Weight). The weight for the expanded query is 1-Weight. The number of documents used for pseudo relevance feedback. The number of terms selected for the feedback query.

One important notice is that in the later experiments, if a retrieved document which ranked in top 10 is not in the 2013 test qrels (since 2013 test topics are our training topics) provided, we judge it by ourselves and add it to the 2013 test qrels. When judging the documents, we always tried to refer how the documents were labeled in the official qrels (Actually, a lot of documents are almost identical, but only some of them were labeled because of pooling). In the end of our experiments, we added total of 310 documents in the qrels. (80 relevant and 230 non-relevent documents.) It is true adding the qrels might make the later comparison against the 2013 official submitted runs and 2013 baselines unfair. But it would be also impossible to improve our retrieval strategies if we don't label the unjudged top 10 retrieved documents.

4.1 Weight of Original Query

We experimented with Weight from 0.1 to 0.9. We set the initial value of # terms and # docs to 20 and 5 respectively. Result is shown in Table 2.

Table 2. Pseudo Relevance Feedback Results Varying Weight (# Docs: 5, # Terms: 20)

weight	P@5	P@10	NDCG@5	NDCG@10	MAP
0.1	0.4520	0.3860	0.4516	0.4099	0.1652
0.2	0.4640	0.4080	0.4644	0.4295	0.1800
0.3	0.4720	0.4260	0.4671	0.4412	0.1924
0.4	0.4800	0.4400	0.4679	0.4483	0.2003
0.5	**0.4880**	0.4480	0.4725	0.4540	0.2075
0.6	**0.4880**	0.4520	0.4748	0.4587	0.2161
0.7	0.4840	**0.4620**	0.4710	0.4662	0.2238
0.8	0.4840	0.4600	0.4749	**0.4675**	0.2294
0.9	0.4840	0.4560	**0.4751**	0.4642	**0.2327**

Weight between 0.6 and 0.9 seem strong across the measures. We favor 0.6 and 0.7 in terms of emphasizing precision at high ranks.

4.2 Number of Documents

We explored different values for number of documents from 5 to 50. We tried both 0.6 and 0.7 for Weight, which is the optimal values from the last experiment.

Table 3. Pseudo Relevance Feedback Results Varying Number of Documents (Weight: 0.6, # Terms: 20)

# Docs	P@5	P@10	NDCG@5	NDCG@10	MAP
5	0.4880	0.4520	0.4748	0.4587	0.2161
10	**0.5000**	**0.4660**	**0.4983**	**0.4788**	**0.2216**
20	0.4680	0.4260	0.4563	0.4341	0.2100
30	**0.5000**	0.4400	0.4868	0.4532	0.2126
40	0.4880	0.4380	0.4804	0.4519	0.2158
50	0.4800	0.4340	0.4739	0.4479	0.2149

Again, the initial value for number of terms is set to 20. Table 3 shows the result for Weight=0.6, as it performs better than 0.7 in the experiment.

The optimal value for number of documents is 10 (both for Weight = 0.6 and 0.7).

4.3 Number of Terms

Next we explored values of number of terms from 5 to 50. We set Weight and # Docs to 0.6 and 10 respectively based on the previous experiments. Table 4 shows the result. We also show the baseline results (without the benefit of pseudo relevance feedback).

Table 4. Experiment of # Terms for Pseudo Relevance Feedback (Weight: 0.6, # Docs: 10)

# Terms	P@5	P@10	NDCG@5	NDCG@10	MAP
5	0.4840	0.4420	0.4935	0.4650	0.2180
10	0.4960	0.4420	0.4958	0.4630	0.2232
15	0.5080	0.4620	0.5015	0.4755	0.2222
20	0.5000	0.4660	0.4983	0.4788	0.2216
25	0.5080	0.4600	0.5063	0.4771	0.2232
30	0.5080	0.4580	0.5040	0.4743	0.2247
35	0.5080	0.4600	0.5069	0.4776	0.2258
40	0.5160	0.4720	0.5128	**0.4880**	0.2290
45	**0.5240**	0.4680	**0.5173**	0.4851	0.2284
50	0.5080	0.4580	0.5040	0.4743	0.2247
Title_All_Text	0.4840	**0.4760**	0.4764	0.4811	**0.2350**

Both 40 and 45 are good values for # Terms. We choose 45 for the later experiment since we would like to focus more on top 5 performance (In the later official evaluation, top 10 was used in the primary measures). Finally our parameters for pseudo relevance feedback, Weight, number of Docs, number of Terms are 0.6, 10, and 45 respectively.

5 Expanding the Query Using Description & Narrative

From the topic example in Section 2.4, we know the title only contains the minimum information for the topic. In order to better describe the information needs of the user, we could expand the query using description or narrative field of the topic.

We explored linear combinations of title and description, title and narrative to improve retrieval performance. Specifically we weight the title by WeightT and weight for description or narrative by 1-WeightT. (We also filtered out stopwords for description or narrative fields.)

The results of linear combination of title and description, title and narrative are shown in Table 5 and Table 6 respectively. We can see that for both Table 5 and Table 6, when the weightT increases, performance also increases. But even the weightT=0.9, it is still not as good as the baseline. Therefore, using description or narrative fields did not significantly improve retrieval performance. These fields may require more sophisticated methods to extract keywords and combine them with the title.

Table 5. Results with Linear Combinations of Title & Description

# WeightT	P@5	P@10	NDCG@5	NDCG@10	MAP
0.1	0.1400	0.1480	0.1353	0.1429	0.0699
0.2	0.1560	0.1620	0.1513	0.1575	0.0731
0.3	0.1800	0.1780	0.1758	0.1759	0.0782
0.4	0.2040	0.2040	0.2084	0.2092	0.0887
0.5	0.2600	0.2360	0.2609	0.2481	0.1093
0.6	0.2960	0.3120	0.3060	0.3179	0.1428
0.7	0.3960	0.3800	0.3939	0.3897	0.1844
0.8	0.4520	0.4320	0.4474	0.4446	0.2132
0.9	0.4800	**0.4780**	0.4690	0.4809	0.2331
Title_All_Text	**0.4840**	0.4760	**0.4764**	**0.4811**	**0.2350**

6 Markov Random Field Model

Inspired by Zhu et al. [8], we explored Markov Random Field (MRF) model [5] as well. Zhu et al. used the parameters settings described in [5]. For example if the topic title is "Coronary artery disease", the expanded Indri query using MRF model should be:

```
#weight( 0.8 #combine(coronary artery disease) 0.1 #combine( #1(coronary artery) #1(artery disease) ) 0.1 #combine( #uw8(coronary artery) #uw8(artery disease) ) )
```

In this section, we describe how we modified the MRF model and explored the

Table 6. Results with Linear Combinations of Title & Narrative

# WeightT	P@5	P@10	NDCG@5	NDCG@10	MAP
0.1	0.2720	0.2580	0.2760	0.2694	0.1261
0.2	0.2800	0.2760	0.2854	0.2855	0.1316
0.3	0.3160	0.3040	0.3181	0.3138	0.1467
0.4	0.3360	0.3140	0.3390	0.3274	0.1576
0.5	0.3600	0.3300	0.3625	0.3457	0.1697
0.6	0.4000	0.3800	0.4029	0.3929	0.1868
0.7	0.4240	0.4040	0.4236	0.4153	0.1978
0.8	0.4360	0.4200	0.4362	0.4310	0.2128
0.9	0.4520	0.4600	0.4487	0.4619	0.2230
Title_All_Text	**0.4840**	**0.4760**	**0.4764**	**0.4811**	**0.2350**

parameters. In order to distinguish the original MRF from our modified version, we call the original MRF, MRF_Bigram since it expands the query using bigrams in the query. And we call our modified version, MRF_MedPhrase.

6.1 MRF_Bigram

There are 3 parameters for MRF_Bigram model: weight of the title (WeightT) (weights for #1 part and uw8 part are both equal to (1-WeightT)/2), Window Type (uw or od: uw/od means unordered/ordered window for the terms), and Window Size (e.g uw8 means unordered window size 8 in Indri). We began with the experiment for the WeightT. The initial value for Window Type & Size are set to uw and 8 respectively. The result is shown in Table 7.

Table 7. Results from Varying WeightT for MRF_Bigram (Window Type & Size: uw8)

# Weight	P@5	P@10	NDCG@5	NDCG@10	MAP
0.7	0.4560	0.4300	0.4553	0.4477	0.2189
0.8	**0.4960**	0.4760	**0.5051**	0.4968	0.2411
0.9	0.4760	**0.4900**	0.4860	**0.5028**	**0.2525**
Title_All_Text	0.4840	0.4760	0.4764	0.4811	0.2350

MRF_Bigram model does improve retrieval performance compared to our baseline (Title_All_Text). The optimal value for the WeightT is 0.8 or 0.9. We choose 0.8 since we focused on the top 5 performance more (Again, the official evaluation later focuses on the top 10).

Next, we would like to find if changing Window Type & Size would affect the retrieval performance. Results exploring Window Type & Size are shown in Table 8.

Therefore, WeightT 0.8, uw5 are our optimal parameters for MRF_Bigram model.

Table 8. Results from Varying Window Type & Size for MRF_Bigram (WeightT: 0.8)

# Weight	P@5	P@10	NDCG@5	NDCG@10	MAP
UW5	**0.5000**	**0.4820**	**0.5045**	**0.4988**	**0.2402**
UW10	0.4840	0.4580	0.4912	0.4801	0.2383
UW15	0.4960	0.4620	0.4997	0.4851	0.2394
OD5	0.4920	0.4740	0.4909	0.4873	0.2358
OD10	0.4840	0.4680	0.4879	0.4842	0.2375
OD15	0.4840	0.4680	0.4894	0.4859	0.2354

6.2 MRF_MedPhrase

MRF_Bigram does improve the retrieval performance, but using bigram does not always make sense. For example, ideally topic "facial cuts and scar tissue" should be interpreted as phrases "facial cuts" and "scar tissue". Bigram "cuts scar" (ignore stopwords) does not make sense. Therefore, we modified the original MRF model and only use medical phrases to expand the query. Using the same example in Section 2.4, MRF_MedPhrase model should generate the query like:

```
#weight( 0.8 #combine(coronary artery disease) 0.1 #combine( #1(coronary artery disease)) 0.1 #combine( #uw5(coronary artery disease) ) )
```

Because coronary artery disease is a medical phrase. Using another topic example: "shortness breath swelling". The query using MRF_MedPhrase model should generate the query like:

```
#weight( 0.8 #combine(shortness breath swelling) 0.1 #combine( #1(shortness breath) swelling ) 0.1 #combine( #uw5(shortness breath) swelling ) )
```

To identify the medical phrases, we use MetaMap [1] to parse the title of topic. Similar with the MRF_Bigram, we found the optimal parameter value for WeightT is 0.8, the Window Type & Size should be set as uw5 as well.

To make the extraction of medical phrases correct, we need to also enabled spell checking (SC) for MRF models. Table 9 shows the comparison for MRF_Bigram and MRF_MedPhrase. In the comparison, we combined MRF with Pseudo Relevance Feedback (RF) as well.

Table 9. Comparision for MRF_Bigram & MRF_MedPhrase

Runs	P@5	P@10	NDCG@5	NDCG@10	MAP
Title_All_Text_MRF_Bigram_RF_SC	0.5320	**0.5060**	0.5254	0.5168	0.2528
Title_All_Text_MRF_MedPhrase_RF_SC	**0.5400**	**0.5060**	**0.5372**	**0.5227**	**0.2651**

Supporting our intuition, MRF_MedPhrase model outperforms MRF_Bigram for all the measures.

7 Expand Medical Abbreviation

Our best run using MRF_MedPhrase with spell checking and pseudo relevance feedback is significantly better than the best runs last year. But there is one more important thing to do. There are several abbreviations in the medical topics, which would be very helpful if we can expand them. However, to expand medial abbreviation is also not trivial. We tried several medical abbreviation lists and found the one from Wikipedia[5] might be the most appropriate one for our task. However, there are still some abbreviations missed. In the 2014 test data, we found "L" could mean "left" which our method cannot expand.

The result is shown in Table 10. Using medical abbreviation expansion does help achieve higher performance.

Table 10. Expanding Medical Abbreviations

Runs	P@5	P@10	NDCG@5	NDCG@10	MAP
Title_All_Text_MRF_MedPhrase_RF_SC	0.5400	0.5060	0.5372	0.5227	**0.2651**
Title_All_Text_MRF_MedPhrase_RF_SC_Abbr	**0.5520**	**0.5120**	**0.5498**	**0.5257**	0.2625

So far, we did several experiments including cleaning the web text, sentence level retrieval, pseudo relevance feedback, linear combination of title and description/narrative, MRF model, spell checking and abbreviation expansion. The comparison between our best strategy and our baseline is shown in Table 11. Our best strategy improved about 15% for the measures on top 5 retrieved results. It also improved about 8-9% for the measures on top 10 retrieved results from baseline.

Table 11. Comparison Between Best Strategy And Baseline

Runs	P@5	P@10	NDCG@5	NDCG@10	MAP
Title_All_Text (Baseline)	0.4840	0.4760	0.4764	0.4811	0.2350
Title_All_Text_ MRF_MedPhrase_ RF_SC_Abbr	**0.5520** (14.05%↑)	**0.5120** (7.56%↑)	**0.5498** (15.41%↑)	**0.5257** (9.27%↑)	**0.2625** (11.7%↑)

8 Submitted Runs And Results

Because the discharge summary is very noisy, we didn't develop retrieval strategies utilizing it. We submitted 4 runs in our final submission. (The baseline is

[5] `http://en.wikipedia.org/wiki/List_of_medical_abbreviations:_A`

run 1, the experiments without discharge summaries should be Runs 5-7. 5 is the highest priority while 7 is the lowest.) Table 12 shows our runs and the technologies used.

Table 12. Submittion and the technologies used

Runs	Pseudo Relevance Feedback	MRF_MedPhrase	Spell Checking	Abbr. Expansion
Run 1				
Run 5	X	X	X	X
Run 6		X	X	X
Run 7	X		X	X

Run 1 is our baseline, which only uses title to retrieve medical documents. Run 5 is our best run, it uses Markov Random Field (MRF) model which expands queries using only medical phrases, it also utilizes abbreviations expansion, pseudo relevance feedback and spell checking. Run 6 is the same as Run 5, but without pseudo relevance feedback. Run 7 is the same as Run 5, but without MRF model.

Table 13 shows the final performance from the official evaluation. Unfortunately, the runs do not significantly differ from each other. Our Run 7 has better scores for P@5 and NDCG@5 which is our original focus. It shows that pseudo relevance feedback has the ability to achieve high accuracy retrieval especially for the top 5 results. (In the final judgement, run 7 submission was not in the judged pool. Therefore, the real performance for run 7 could be even higher.) But our baseline (Run 1) has better performance for P@10 and NDCG@10 which are the primary official measures. The MRF model we trained using 2013 test data does not improve retrieval performance using 2014 test dataset. The reason could be that we overfitted the model though we attempted to avoid that pitfall.

Table 13. Performance Of Submitted Runs

Runs	P@5	P@10	NDCG@5	NDCG@10	MAP
Run 1	0.6880	**0.6900**	0.6705	**0.6784**	**0.3589**
Run 5	0.6840	0.6600	0.6579	0.6509	0.3226
Run 6	0.6760	0.6820	0.6380	0.6520	0.3259
Run 7	**0.7000**	0.6760	**0.6777**	0.6716	0.3452

Figure 3 shows our Run 1 (since it has the best top 10 performance in our runs) against the median and best performance (p@10) across all systems submitted to CLEF for each query topic. Topics 8, 13, 15, 28, 34, 44, and 50 are easily handled by Run 1, but topics 7, 11, 22, 32, 38, 40, 47 are difficult for it.

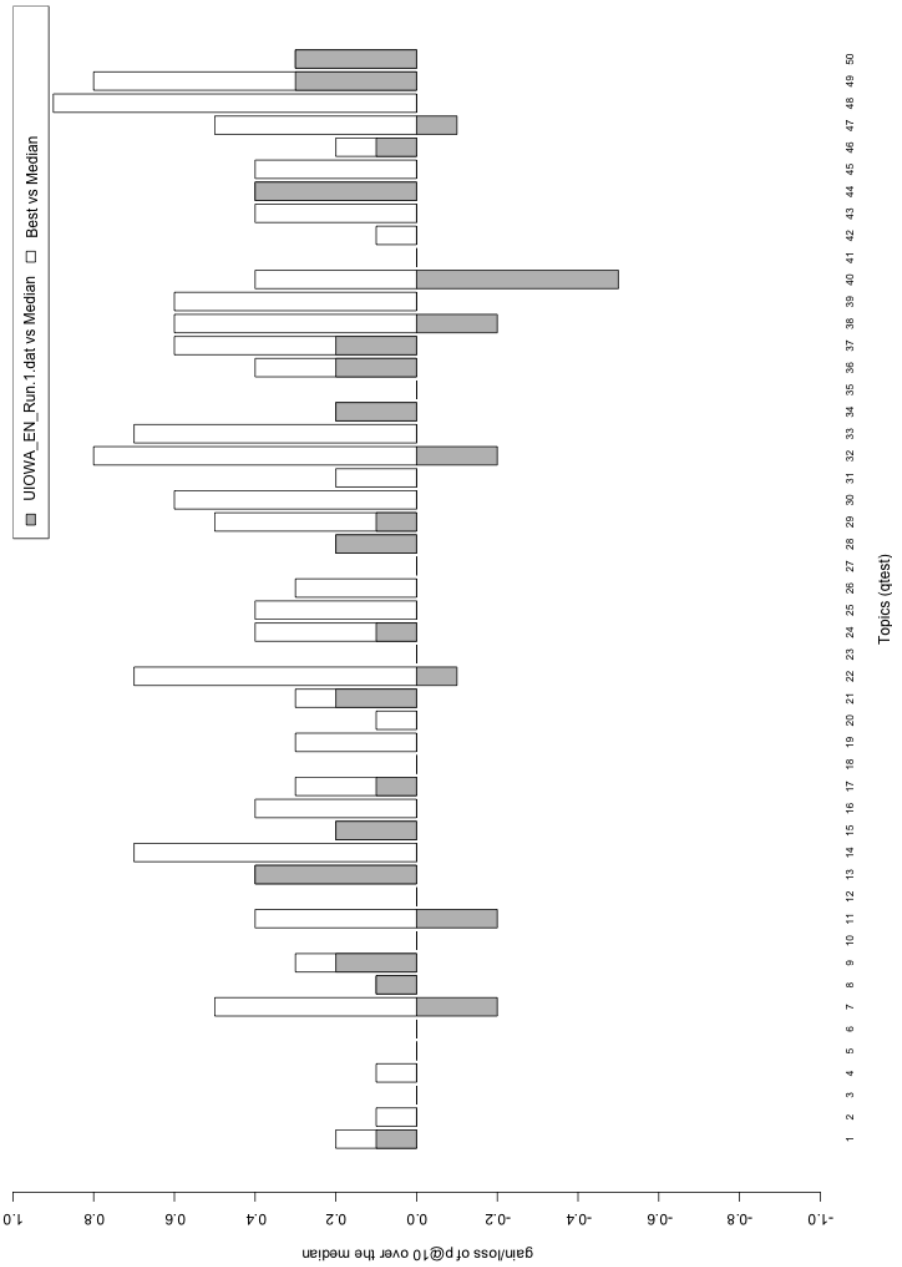

Fig. 3. Run 1 Performance For Each Topics

9 Conclusion

We explored cleaning of the dataset and sentence level retrieval. We showed that retrieval performance did not improve by utilizing the two methods. We also tried linear combinations of title and description/narrative, it seems it is a non trivial task. We did experiments to find out the optimal parameters for pseudo relevance feedback, showed that it can achieve higher performance for top 5 retrieved items. We modified the Markov Random Field model by using the medical phrases to expand the query. This method shows the ability to achieve higher performance on the 2013 queries but fails using the 2014 test dataset. Future work planned includes a more sophisticated method to combine the title and description/narrative/discharge summary, and avoiding the overfitting of the MRF model.

References

1. A. R. Aronson and F.-M. Lang. An overview of metamap: historical perspective and recent advances. *Journal of the American Medical Informatics Association*, 17(3):229–236, 2010.
2. L. Goeuriot, L. Kelly, W. Li, J. Palotti, P. Pecina, G. Zuccon, A. Hanbury, G. Jones, and H. Mueller. Share/clef ehealth evaluation lab 2014, task 3: User-centred health information retrieval. In *Proceedings of CLEF 2014*, 2014.
3. L. Kelly, L. Goeuriot, H. Suominen, T. Schrek, G. Leroy, D. L. Mowery, S. Velupillai, W. W. Chapman, D. Martinez, G. Zuccon, and J. Palotti. Overview of the share/clef ehealth evaluation lab 2014. In *Proceedings of CLEF 2014*, Lecture Notes in Computer Science (LNCS). Springer, 2014.
4. V. Lavrenko and W. B. Croft. Relevance based language models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 120–127. ACM, 2001.
5. D. Metzler and W. B. Croft. A markov random field model for term dependencies. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 472–479. ACM, 2005.
6. R. Saetre, K. Yoshida, A. Yakushiji, Y. Miyao, Y. Matsubayashi, and T. Ohta. Akane system: protein-protein interaction pairs in biocreative2 challenge, ppi-ips subtask. In *Proceedings of the Second BioCreative Challenge Workshop*, pages 209–212, 2007.
7. T. Strohman, D. Metzler, H. Turtle, and W. B. Croft. Indri: A language model-based search engine for complex queries. In *Proceedings of the International Conference on Intelligent Analysis*, volume 2, pages 2–6. Citeseer, 2005.
8. D. Zhu, S. Wu, M. James, B. Carterette, and H. Liu. Using discharge summaries to improve information retrieval in clinical domain. *Proceedings of the ShARe/-CLEF eHealth Evaluation Lab*, 2013.