Classification and retrieval of biomedical literatures: SNUMedinfo at CLEF QA track BioASQ 2014

Sungbin Choi, Jinwook Choi

Medical Informatics Laboratory, Seoul National University, Seoul, Republic of Korea

wakeup06@empas.com, jinchoi@snu.ac.kr

Abstract. This paper describes the participation of the SNUMedinfo team at the BioASQ Task 2a and Task 2b of CLEF 2014 Question Answering track. Task 2a was about biomedical semantic indexing. We trained SVM classifiers to automatically assign relevant MeSH descriptors to the MEDLINE article. Regarding Task 2b biomedical question answering, we participated at the document retrieval subtask in Phase A and the ideal answer generation subtask in Phase B. In the document retrieval task, we mostly experimented with semantic concept-enriched dependence model and sequential dependence model. Semantic concept-enriched dependence model showed significant improvement over baseline. In the ideal answer generation task, we reformulated task as, given relevant lists of passages, selecting the best ones to build the answer. We applied three heuristic methods.

Keywords: SVM, Text categorization, Information retrieval, Semantic concept-enriched dependence model, Sequential dependence model

1 Introduction

In this paper, we describe the participation of the SNUMedinfo team at the BioASQ Task 2a and Task 2b of CLEF 2014 [1]. Task 2a was about large-scale online biomedical semantic indexing; automatically annotating MEDLINE® document with the Medical Subject Headings (MeSH®) descriptor. Task 2b was about biomedical semantic question answering task, ranging from document retrieval subtask to the ideal answer generation subtask. For a detailed task introduction, please see the overview paper of CLEF Question Answering track BioASQ 2014'.

2 Methods

2.1 Task 2a

In the task 2a, we used Support Vector Machine (SVM) [2] with linear kernel type as a document classifier. We hypothesized computing resource-limited situation. If we increase the number of training documents, classification performance will be naturally

improved, but it will require more computing resources. In this study, we fixed the number of training documents to 50,000. We tried to draw better performance out of limited training document size.
We tried different training document selection strategies to select most useful 50,000 documents out of candidate 11 million MEDLINE documents to build the effective classifier per each MeSH descriptor.
The main directions of our experimentation can be summarized as the following two stages; Training and Classification.

Stage 1. Training SVM classifier
Stage 1 can be divided into the following 9 steps (step 0 to 8), as depicted in the Figure 1. We trained SVM classifiers per each MeSH descriptor individually.

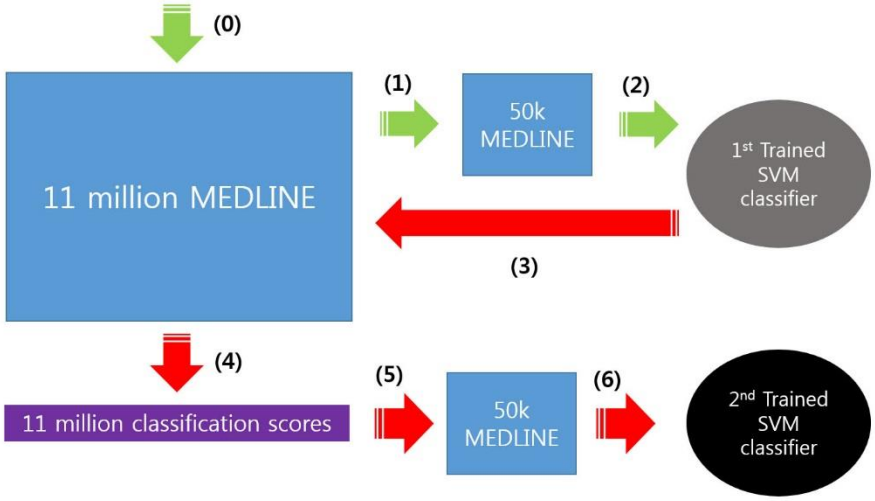

Fig. 1. General scheme about training SVM classifier

Step 0: Preparing potential training set 11 million MEDLINE document.

We leased 2014 MEDLINE/PubMed® Journal Citations [3] from the U.S. National Library of Medicine, composed of roughly 22 million MEDLINE citations. These files are compressed into 746 zip files (Numbered from 1 to 746). We used only 346 files (Numbered from 401 to 746), which contains roughly 11 million articles published within last 20 years[1].

Step 1: Randomly select 50,000 documents as a 1st training set .

[1] We didn't use training datasets distributed from the BioASQ homepage, because we were planning to use only very small (50,000) samples of documents as a training set.

Among 11 million MEDLINE documents, 50,000 documents are randomly selected. Half of them (25,000) were filled with target-positive document (which means that target MeSH descriptor is tagged in), and the other half were filled with target-negative document (which means that target MeSH descriptor is not tagged in). When total number of target-positive document is less than 25,000, number of target-negative document is increased to make total number of training set document 50,000 constant.

Per each document in training set, title and abstract text field were extracted, punctuation removed, case-folded, tokenized and stemmed using Porter stemmer. Per each unique stem, document stem frequency is used as feature value.

Step 2: Train SVM classifier

SVM classifier is trained on 50,000 documents selected from step 1.

Step 3 and 4: Applying trained svm classifier to the 11 million documents

SVM classifier trained in step 2 is applied on the 11 million documents. Per each document, document id (PMID) and classification score is stored.

Step 5: Selecting new training set

New training set document is selected from 11 million MEDLINE documents, based on classification scores calculated in step 4. We tried two different methods.

1. Selecting most difficult ones

Documents having most undesirable classification score is chosen. For target-positive document, documents having lowest classification score (so it could be wrongly classified as target-negative document) are selected. For target-negative document, documents having highest classification score (so it could be wrongly classified as target-positive document) are selected

2. Random selection among the most difficult ones

25,000 training documents are randomly selected from top 10% most undesirable documents of target-positive document. 25,000 training documents are randomly selected from top 10% most undesirable documents of target-negative document (If there are 10 million target-negative documents, top 1 million documents having highest classification score is selected first. Then, among this 1 million documents, 25,000 documents are randomly chosen).

Step 6: Train SVM classifier

SVM classifier is trained on 50,000 documents selected from step 5.

Step 7: Choose best classifier

After finishing step 6, we have three trained SVM classifiers (One SVM classifier from step 2, two SVM classifier from step 5). Now we choose single best classifier

based on Precision at *k* metric (*k* : number of target-positive documents among 11 million MEDLINE documents).

Step 8: Build the mapping table for translating SVM classification score into the expected precision, expected recall and expected F1 measure

After stage 7, SVM classifier is prepared per each MeSH descriptor. We calculated expected precision value as depicted in Figure 2. 11 million MEDLINE documents are sorted in descending order by the SVM classification score. Expected precision is calculated on each target-positive document.

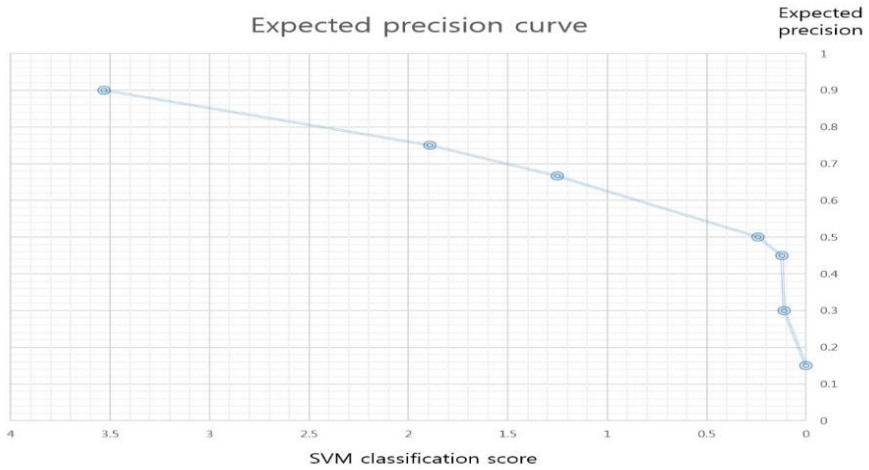

Fig. 2. Expected precision curve example: x-axis represents SVM classification score. y-axis represents expected precision value.

Now we can translate SVM classification score to the probability of target-positive. For example, with regard to the Figure 2, if we apply SVM classifier on the unseen MEDLINE document, and SVM classification score is calculated as 2.5, then the probability of this MEDLINE document being target-positive is estimated as 75%. If classification score is 1.0, then the probability of this MEDLINE document is target-positive is estimated as 50%.

In the same way, we can estimate expected recall and expected F1 measure too.

Stage 2. Classification

After stage 1, now we have trained SVM classifier, and corresponding mapping table for translating classification score to the expected precision, expected recall and expected F1 measure. When new test set document is given, per each MeSH descriptor, we apply SVM classifier to get the classification score, expected precision, expected recall and expected F1 measure.

When we decide output MeSH descriptor on the test set document, we tried following two different methods.

Method 1: Choose output MeSH descriptor by expected precision threshold

Two parameters are used; *p* for the precision threshold; *m* for the maximum number of MeSH descriptor that each document can have.

Per each test document given, among all candidate MeSH descriptors satisfying expected precision threshold *p*, top *m* MeSH descriptors are chosen.

Method 2: Choose output MeSH descriptor by F1-measure optimized threshold[2]

- Individual F1 measure optimized: Classification score threshold is decided based on maximum F1 measure point per each MeSH descriptor individually.
- Micro-F1 measure optimized: Per each MeSH descriptors, classification score threshold is initialized as the above mentioned Individual F1 measure optimized method. Then, all MeSH descriptors are sorted by the number of target-positive document in descending order. From MeSH descriptor having largest document frequency, we start finding micro-F1 measure optimal classification score threshold by changing classification score threshold on this specific MeSH descriptor while all other MeSH descriptor's score thresholds fixed. Per each cycle, we repeat this process for every MeSH descriptor. We repeat this cycle until there are no more overall Micro-F1 measure performance gain is observed.

2.2 Task 2b Phase A – Document retrieval

In Task 2b Phase A, we participated at the document retrieval subtask only. We used Indri search engine [4].

Indexing

We leased 2014 MEDLINE/PubMed Journal Citations [3] from the U.S. National Library of Medicine, composed of roughly 22 million MEDLINE citations. According to the task guideline, we filtered out articles published after March 14, 2013. Per each MEDLINE citation, article title, abstract, MeSH descriptor and publication type fields are extracted and indexed with Indri without stopword removal.[3]

Retrieval

The queries are stopped at the query time using the standard 418 INQUERY stopword list, case-folded, and stemmed using Porter stemmer. We used unigram language model with Dirichlet prior smoothing [5] as our baseline retrieval method (referred as QL: query likelihood model).

At this task, we tried to test various retrieval techniques and parameter settings. But most of our submitted runs use dependence models (SDM and SCDM). Our experimental methods can be summarized as the following three category.

---

[2] We didn't try optimizing hierarchical measure rather than this flat one (F1 measure), because we didn't have much time for preparation.

[3] We didn't use the web services provided by the competition, because we need to use Indri search engine to implement our retrieval methods.

Sequential dependence model (SDM).

In [6], Metzler and Croft proposed sequential dependence model which incorporates sequential query term dependence into the retrieval model. SDM assumes dependency between adjacent query terms. SDM showed better experimental performance on various TREC test collections [6-8] compared to the baseline query likelihood model (QL), or a full dependence model (FDM) which assumes that all query terms are dependent on each other.
SDM Indri query example for the original query 'What is the inheritance pattern of Emery-Dreifuss muscular dystrophy?' can be described as follows.

#weight (

λ_T #combine(inheritance pattern emery dreifuss muscular dystrophy)

λ_O #combine(#od1(inheritance pattern) #od1(pattern emery) #od1(emery dreifuss) #od1(dreifuss muscular) #od1(muscular dystrophy))

λ_U #combine(#uw8(inheritance pattern) #uw8(pattern emery) #uw8(emery dreifuss) #uw8(dreifuss muscular) #uw8(muscular dystrophy)))

λ_T, λ_O, λ_U are weight parameters for single terms, ordered phrases and unordered phrases, respectively.

Semantic concept-enriched dependence model (SCDM).

In [9], Choi et al. proposed incorporating semantic concept-based term dependence feature into a retrieval model. Standardized medical concept terms are assumed to have implicit term dependency within the same concept. Using MetaMap, all of the existing UMLS concepts in the original query text are identified.
We experimented with two different variants of SCDM. For detailed explanation about SCDM, please see [9]. SCDM Indri query example can be described as follows.

- SCDM type C (single + multi-term, all-in-one)

#weight(

λ_T #combine(inheritance pattern emery dreifuss muscular dystrophy)

λ_O #combine(#od1(inheritance pattern) #od1(pattern emery) #od1(emery dreifuss) #od1(dreifuss muscular) #od1(muscular dystrophy))

λ_U #combine(#uw8(inheritance pattern) #uw8(pattern emery) #uw8(emery dreifuss) #uw8(dreifuss muscular) #uw8(muscular dystrophy))

λ_{O_SC} #combine(#od1(inheritance pattern) #od1(emery dreifuss muscular dystrophy))

λ_{U_SC} #combine(#uw8(inheritance pattern) #uw16(emery dreifuss muscular dystrophy)))

λ_T, λ_O, λ_U, λ_{O_SC}, λ_{U_SC} are weight parameters for single terms, ordered phrases and unordered phrases of sequential query term pairs, ordered phrases and unordered phrases of semantic concepts, respectively.

- SCDM type D (single+multi-term, pairwise)

#weight(

λ_T #combine(inheritance pattern emery dreifuss muscular dystrophy)

λ_O #combine(#od1(inheritance pattern) #od1(pattern emery) #od1(emery dreifuss) #od1(dreifuss muscular) #od1(muscular dystrophy))

λ_U #combine(#uw8(inheritance pattern) #uw8(pattern emery) #uw8(emery dreifuss) #uw8(dreifuss muscular) #uw8(muscular dystrophy))

λ_{O_SC} #combine(#od1(inheritance pattern) #od1(emery dreifuss) #od1(dreifuss muscular) #od1(muscular dystrophy))

λ_{U_SC} #combine(#uw8(inheritance pattern) #uw8(emery dreifuss) #uw8(dreifuss muscular) #uw8(muscular dystrophy)))

Query expansion using top-k MEDLINE document's title field (TitleQE).

From the top *k* pseudo-relevant documents retrieved from 22 million documents, article title field is extracted and then added to the original query. Using large 22 million documents as reference [10], we expect top *k* documents are highly relevant to the original query, and their title field contains relevant terms to the original query. TitleQE Indri query example can be described as follows.

#weight (
(1-*w*) #combine(inheritance pattern emery dreifuss muscular dystrophy)
w #combine(cardiomyopathy atrioventricular block emery dreifuss muscular dystrophy case report emery dreifuss muscular dystrophy case report laminopathy saga))

w is a weight parameter for the expansion query part. Original query part is weighted by 1-*w*.

2.3 Task 2b Phase B – Ideal answer generation

In Task 2b Phase B, we participated only at the ideal answer generation subtask. We reformulated this task as, among relevant lists of passages given[4], selecting appropriate ones. We tried following three heuristic methods to select *m* passages and combine them to form the ideal answer.

Selecting shortest passages (Selecting shortest passages)

[4] We used gold relevant text snippets provided by the BioASQ.

In this method, we hypothesized that number of tokens in each passage represents the conciseness of content. If a passage has less tokens, it is assumed a good candidate for the ideal answer. We rank relevant passages by number of tokens in ascending order, and select top *m* passages as the ideal answer.

Identifying keyword terms and rank passages based on the number of unique keywords it contain (Selecting key passages)

In this method, parameter *minDF* represents minimum proportion of passages that keyword term should occur.

Firstly we tried to identify keyword terms. If there are 20 relevant passages given, and *minDF* is set to 0.5, then any terms occurring ≥ 10 passages are considered as keywords.

With identified keywords list, we rank passages based on the number of unique keywords each passage contains. Top *m* passages are selected as the ideal answer.

Selecting passages different from the previously chosen passage (Selecting complementary passages)

In this method, parameter *minUnseen* represents minimum proportion of new tokens that does not exist in the previously selected passages.

This methods builds upon the *Selecting key passages* method described above. Firstly, we rank passages in the way same as *Selecting key passages*. We select top-ranked passage. Then, regarding the second-ranked passage, we check proportion of tokens in the second passage that does not occur in the previously selected passages, and if it is ≥ *minUnseen* threshold, second-ranked passage is selected. If proportions of newly found tokens are below *minUnseen* threshold, that passage is abandoned, and we check next rank passage. This process is repeated until *m* passage is selected.

In this method, our intention was enhancing comprehensiveness of answer text by increasing the diversity of tokens.

3 Results & Discussion

At the moment of writing this paper, the evaluation result is not complete. So we analyzed results based on this tentative non-final version evaluations.

3.1 Task 2a

At this task, we participated from batch 2 week 2 to batch 3 week 5. When we start participating this task, our preparation was not complete. So we incrementally applied our methods described in section 2.1. We can separate our runs into three distinct periods.

- First period (batch 2 week 2 ~ batch 2 week 5): Training(Prepared up to Stage 1 step 4), Classification (Method 1)
- Second period (batch 3 week 1): Training (Complete), Classification (Method 1)

- Third period(batch 3 week 2 ~ batch 3 week 5): Training (Complete), Classification (Method 2)

For our first period, only one SVM classifier is trained from randomly selected 50k documents (Because, preparation for the Stage 1 step 5 was incomplete). For our second and third period, best SVM classifier is chosen from three distinct SVM classifier trained using different training document selection methodologies respectively.

For our first and second period, arbitrary expected precision threshold (Method 1) is used for the classification. For our third period, classification threshold is optimized on the expected micro F1 measure (Method 2), which is target evaluation metric for this task.

Roughly, in our first period, micro F1 measures are estimated as 0.45~0.47. But in our second period, it is improved to 0.49. In our third period, again it is improved to 0.48~0.52.

We have two lessons learned from this task.

1. We limited number of training documents to 50k. We used only simple SVM classifier without applying feature selection methods or using more sophisticated machine learning algorithms. But our methods showed certain level of performance, although it is clearly lower than the top performance team's (Top performing team's micro F1 measure was roughly 0.60).
2. As the competition goes on, performance of our methods gets improved increasingly. The performance of our second period was better than the first period, and the performance of our third period was better than the second period.

3.2 Task 2b Phase A – Document retrieval

There were five distinct batches in this task. The primary evaluation metric was gmap (geometric mean average precision) over top 100 documents. We tried to find best retrieval method and parameter setting. Our submitted runs used following parameter settings.

Batch1.
SNUMedinfo1: QL (mu=750)
SNUMedinfo2: TitleQE (mu=750, k=5, w=0.1)
SNUMedinfo3: TitleQE (mu=750, k=5, w=0.2)
SNUMedinfo4: TitleQE (mu=750, k=5, w=0.3)
SNUMedinfo5: QL (mu=750)

Batch2.
SNUMedinfo1: SCDM Type C ($mu=500, \lambda_T=0.85, \lambda_O=0.00, \lambda_U=0.00, \lambda_{O\_SC}=0.10, \lambda_{U\_SC}=0.05$)
SNUMedinfo2: SCDM Type C ($mu=500, \lambda_T=0.70, \lambda_O=0.00, \lambda_U=0.00, \lambda_{O\_SC}=0.20, \lambda_{U\_SC}=0.10$)
SNUMedinfo3: SCDM Type C ($mu=750, \lambda_T=0.85, \lambda_O=0.00, \lambda_U=0.00, \lambda_{O\_SC}=0.10, \lambda_{U\_SC}=0.05$)
SNUMedinfo4: SCDM Type C ($mu=750, \lambda_T=0.70, \lambda_O=0.00, \lambda_U=0.00, \lambda_{O\_SC}=0.20, \lambda_{U\_SC}=0.10$)

SNUMedinfo5: SCDM Type C (mu=1000, λ_T=0.85, λ_O=0.00, λ_U=0.00, λ_{O_SC}=0.10, λ_{U_SC}=0.05)

Batch3.

SNUMedinfo1: SCDM Type C (mu=500, λ_T=0.85, λ_O=0.00, λ_U=0.00, λ_{O_SC}=0.10, λ_{U_SC}=0.05)
SNUMedinfo2: SCDM Type C (mu=500, λ_T=0.70, λ_O=0.00, λ_U=0.00, λ_{O_SC}=0.20, λ_{U_SC}=0.10)
SNUMedinfo3: SCDM Type C (mu=750, λ_T=0.85, λ_O=0.00, λ_U=0.00, λ_{O_SC}=0.10, λ_{U_SC}=0.05)
SNUMedinfo4: SCDM Type C (mu=750, λ_T=0.70, λ_O=0.00, λ_U=0.00, λ_{O_SC}=0.20, λ_{U_SC}=0.10)
SNUMedinfo5: SCDM Type D (mu=500, λ_T=0.85, λ_O=0.00, λ_U=0.00, λ_{O_SC}=0.10, λ_{U_SC}=0.05)

Batch4.

SNUMedinfo1: SCDM Type C (mu=500, λ_T=0.85, λ_O=0.00, λ_U=0.00, λ_{O_SC}=0.10, λ_{U_SC}=0.05)
SNUMedinfo2: SCDM Type D (mu=500, λ_T=0.85, λ_O=0.00, λ_U=0.00, λ_{O_SC}=0.10, λ_{U_SC}=0.05)
SNUMedinfo3: SCDM Type C (mu=500, λ_T=0.70, λ_O=0.10, λ_U=0.05, λ_{O_SC}=0.10, λ_{U_SC}=0.05)
SNUMedinfo4: SCDM Type D (mu=500, λ_T=0.70, λ_O=0.10, λ_U=0.05, λ_{O_SC}=0.10, λ_{U_SC}=0.05)
SNUMedinfo5: SDM (mu=500, λ_T=0.85, λ_O=0.10, λ_U=0.05)

Batch5.

SNUMedinfo1: SCDM Type C (mu=500, λ_T=0.85, λ_O=0.00, λ_U=0.00, λ_{O_SC}=0.10, λ_{U_SC}=0.05)
SNUMedinfo2: SCDM Type C (mu=500, λ_T=0.70, λ_O=0.00, λ_U=0.00, λ_{O_SC}=0.20, λ_{U_SC}=0.10)
SNUMedinfo3: SCDM Type D (mu=500, λ_T=0.85, λ_O=0.00, λ_U=0.00, λ_{O_SC}=0.10, λ_{U_SC}=0.05)
SNUMedinfo4: SCDM Type C (mu=500, λ_T=0.70, λ_O=0.10, λ_U=0.05, λ_{O_SC}=0.10, λ_{U_SC}=0.05)
SNUMedinfo5: SDM (mu=500, λ_T=0.85, λ_O=0.10, λ_U=0.05)

In Batch 1, mostly we applied TitleQE method. Evaluation result for batch 1 is described in the following Table 1. We also report evaluation results from the training set, which has 307 queries[5]. We performed two-tailed paired t-test over the map (**: p-value < 0.01, *: p-value < 0.05).

Table 1. Batch 1 evaluation results

	Training set map (%)[6] significance gmap	**Batch1** map (%) significance gmap
QL (mu=750)	0.2256 0.0571	**0.2612** **0.0519**
TitleQE (mu=750, k=5, w=0.1)	0.2308 (+2.3%) ** **0.0578**	0.2587 (-1.0%) 0.0501

[5] We used training set provided by the BioASQ challenge. Originally, there are 310 queries in the training set. We removed queries having duplicate query id.

[6] Relative change compared to the baseline (QL) map performance

TitleQE (mu=750, k=5, w=0.2)	**0.2312** (+2.5%) 0.0565	0.2493 (-4.6%) 0.0468
TitleQE (mu=750, k=5, w=0.3)	0.2303 (+2.1%) 0.0532	0.2410 (-7.7%) 0.0449
QL (mu=1,000)	0.2225 0.0542	0.2547 0.0460

In Batch 1, TitleQE method failed to show significant performance improvement over QL. For Batch 2~5, we submitted runs using SDM and SCDM. Evaluation results are summarized in Table 2. Best performance result per each batch is highlighted in bold face.

Table 2. Evaluation results for the QL, SDM and SCDM methods

	Train-ing set	**Batch1**	**Batch2**	**Batch3**	**Batch4**	**Batch5**
QL (mu=500)	0.2289 0.0546	0.2614 0.0581	0.2806 0.0977	0.2931 0.0424	0.2551 0.0320	0.2555 0.0350
SDM (mu=500, λ_T=0.85, λ_O=0.10, λ_U=0.05)	0.2417 (+5.6%) * 0.0557	0.2735 (+4.6%) 0.0619	0.2867 (+2.2%) 0.1039	0.3059 (+4.4%) 0.0503	0.2692 (+5.5%) 0.0404	**0.2689** (+5.2%) **0.0427**
SCDM type C (mu=500, λ_T=0.85, λ_O=0.00, λ_U=0.00, λ_{O_SC}=0.10, λ_{U_SC}=0.05)	0.2392 (+4.5%) ** 0.0618	0.2808 (+7.4%) ** 0.0670	**0.2922** (+4.1%) 0.1042	0.3074 (+4.9%) * 0.0452	0.2748 (+7.7%) ** 0.0390	0.2665 (+4.3%) 0.0372
SCDM type C (mu=500, λ_T=0.70, λ_O=0.00, λ_U=0.00, λ_{O_SC}=0.20, λ_{U_SC}=0.10)	0.2412 (+5.4%) ** 0.0608	0.2870 (+9.8%) * **0.0672**	0.2903 (+3.5%) 0.1038	**0.3152** (+7.5%) * 0.0453	**0.2847** (+11.6%) ** **0.0417**	0.2634 (+3.1%) 0.0364
SCDM type C (mu=500, λ_T=0.70, λ_O=0.10, λ_U=0.05, λ_{O_SC}=0.10, λ_{U_SC}=0.05)	**0.2423** (+5.9%) ** **0.0625**	**0.2875** (+10.0%) * 0.0661	0.2869 (+2.2%) **0.1072**	0.3151 (+7.5%) * **0.0514**	0.2799 (+9.7%) * 0.0404	0.2686 (+5.1%) 0.0343

Generally, SDM and SCDM showed better performance compared to the QL. SCDM showed significant improvement over QL in most of the batches.

3.3 Task 2b Phase B – Ideal answer generation

Evaluation results are not available at the time of wring. Please check the overview paper or homepage of BioASQ for the later release of evaluation result.

4 Conclusion

In BioASQ 2014, we experimented with various classification and retrieval methods on the MEDLINE document. In Task 2a, we experimented with baseline SVM classifier with linear kernel type. We tried various training set document selection methodologies and target evaluation measure optimized threshold adjusting methods. In Task 2b Phase A document retrieval subtask, mainly we experimented with sequential dependence model and semantic concept-enriched dependence model. Semantic concept-enriched dependence model showed significant improvement over baseline. In Task 2b Phase B ideal answer generation subtask, we reformulated task as selecting appropriate passages among relevant list of passages. We tried three heuristic methods.

Evaluation results for our submitted runs were encouraging. We'll explore more effective methods in our future study.

Acknowledgements

This study was supported by a grant of the Korean Health Technology R&D Project, Ministry of Health & Welfare, Republic of Korea. (No. HI11C1947)

5 References

1. Cappellato, L., et al. *CLEF 2014 Labs and Workshops* 2014.
2. Joachims, T., *Svm-light support vector machine, 2002*. URL= http://svmlight. joachims. org, 2009.
3. *To Lease MEDLINE®/PubMed® and other NLM® Databases*. 2013 [cited 2014 July 12]; Available from: http://www.nlm.nih.gov/databases/license/license.html.
4. Strohman, T., et al. *Indri: A language model-based search engine for complex queries*. in *Proceedings of the International Conference on Intelligent Analysis*. 2005. McLean, VA.
5. Zhai, C. and J. Lafferty, *A study of smoothing methods for language models applied to Ad Hoc information retrieval*, in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*. 2001, ACM: New Orleans, Louisiana, USA. p. 334-342.

6. Metzler, D. 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*. 2005, ACM: Salvador, Brazil. p. 472-479.
7. Bendersky, M. and W.B. Croft, *Discovering key concepts in verbose queries*, in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. 2008, ACM: Singapore, Singapore. p. 491-498.
8. Bendersky, M., D. Metzler, and W.B. Croft, *Learning concept importance using a weighted dependence model*, in *Proceedings of the third ACM international conference on Web search and data mining*. 2010, ACM: New York, New York, USA. p. 31-40.
9. Choi, S., et al., *Semantic concept-enriched dependence model for medical information retrieval.* Journal of Biomedical Informatics, 2014. **47**(0): p. 18-27.
10. Choi, S., J. Lee, and J. Choi. *SNUMedinfo at ImageCLEF 2013: Medical retrieval task.* in *CLEF 2013 Evaluation Labs and Workshop, Online Working Notes*. 2013.