Concept-based Information Retrieval for Clinical Case Summaries

Team NU_UU_UNC

Jakob Stöber*[1], Bret S. E. Heale*, PhD[1], Heejun Kim*, MS[2], Kelley Fulghum, MD[1], Kalpana Raja, PhD[3], Guilherme Del Fiol, MD, PhD[1] and Siddhartha R. Jonnalagadda, PhD[3]

1 Department of Biomedical Informatics, University of Utah, Salt Lake City, UT, USA.
2 School of Information and Library Science, University of North Carolina, Chapel Hill, NC, USA.
3 Division of Health and Biomedical Informatics, Department of Preventive Medicine, Feinberg School of Medicine, Northwestern University, Chicago, IL, USA.

*The authors wish it be known that these authors contributed equally.

Abstract

Objective: Query representation is a classic information retrieval (IR) problem. Forming appropriate query representations from clinical free-text adds additional complexity. We examined if external search engine mediated conceptualization based on expert knowledge, concept representation of the abstract, and application of machine learning improve the process of clinical information retrieval.
Methods: Diagnosis concepts were derived through either using a Google Custom Search over a specific set of health-related websites or through manual, expert clinical diagnosis. We represented concepts both as text and UMLS concepts identified with MedTagger. Our approaches leverage Lucene indexing/searching of article full text, abstracts, titles and semantic representations. Additionally, we experimented with automatically generated diagnosis using Web search engines and the case summaries. Further, we utilized a variety of filters such as PubMed's Clinical Query filters, which retrieve articles with high scientific quality, and UMLS semantic type filters for search terms. In our final submission for the TREC 2015 CDS challenge, we focused on three approaches:

1. DFML/DFMLB: Combined ranking scores by data fusion and relevance probabilities derived by a machine learning method to offset ranking and classification errors.
2. HAKT/HMKTB: Used an iterative hierarchical search approach that progressively relaxed filters until we reached 1000 retrieved documents.
3. MDRUN/MDRUB: Manually added a diagnosis to each case and matched UMLS concepts by manual annotations with UMLS concepts in the case summaries.

Results: The concepts extracted from search results are similar to the diagnosis concepts extracted from manual annotation by clinicians, and similar to the extracted concepts from the given diagnosis in task B. Two out of our three approaches performed above the median performance by all participants for both Task A and B. Overall, the run by manual diagnosis

worked the best. The similarity between manual annotation by clinicians and given diagnosis in task B partially explains the performance of our algorithms. There was statistically significant difference in performance among our runs with two measures (R-prec and Prec@10) for Task A, but we could not find difference with other two measures (infNDCG and infAP) for Task A and all measures for Task B.
Discussion: Our concept based approach avoids the need to remove stop words or stemming and reduces the need to look for synonyms.
Conclusions: Overall, our major innovations are query transformation using diagnosis information inferred from Google searching of health resources, concept based query and document representation, and pruning of concepts based on semantic types and groups.

1. Introduction

Query representation is a classic information retrieval (IR) problem. However, forming appropriate query representations from clinical free-text introduces additional complexity to the problem. Finding and representing the diagnosis data-element from clinical summaries is critical. Identifying and representing diagnoses as concepts is an interesting solution to overcome the complexity caused by clinical free-text. The search engine of Google has been shown to be an effective diagnostic tool [1]. MedTagger [2] is an effective natural language processing (NLP) tool used to identify Unified Medical Language System metathesaurus (UMLS) [9] concepts in free-text. UMLS contains a near-comprehensive list of biomedical concepts arranged in a semantic network of types and groups. The UMLS semantic network can be leveraged to focus on a set of concepts relevant to diagnosis. Thus, we propose pairing Google and MedTagger to solve the challenge of clinical summary query representation. In addition to query representation, measuring the relevancy of conceptually matched documents can improve the precision of IR. Ranking documents with Data Fusion [6] and classifying relevance using Machine Learning are two main approaches typically used in isolation.

In addition to query representation, measuring the relevancy of conceptually matched documents can improve the precision of IR. Ranking documents with data fusion and classifying relevance using machine learning are two main approaches typically used in isolation. Combining data fusion and a machine learning based classification can potentially reduce both ranking and classification errors [3].

We sought to improve retrieval of documents relevant to a case summary through using a Google search of health resources, MedTagger, and tuning of a filter based on expert opinion of UMLS semantic groups and types. Additionally, we explored the use of a combination of Data Fusion and Machine learning to improve performance.

2. Background and Significance

2.1 TREC CDS challenge

To bring information closer to the point-of-care, the TREC CDS challenge investigates techniques for information retrieval of information relevant to clinical care. The 2015 challenge provided 30 case topics, which contain a description, a summary and type: diagnosis, test or treatment. A diagnosis was also given for task B. Additionally; a document set of PubMed Central (PMC) articles from the January 2014 PMC snapshot was included as the target of information retrieval. The goal of the challenge was to return the 1,000 most relevant documents per topic. We leveraged a set of publicly available resources to develop our approach. These are described briefed in the sections below.

2.2 Lucene

Apache Lucene is an open source indexing/search software, which is written and implementable in Java.[1] The software package is widely used in industry and academia as a basis of search engines. Lucene works based on an inverted index of documents, which can have several attributes such as title, author and contents. Each field can be searched, is sortable and can be ranked by a certain weight. Also in querying the index, Lucene supports boolean queries, queries for phrases, wildcards and ranges.

2.3 UMLS

The Unified Medical Language System metathesaurus (UMLS) is a near-comprehensive list of biomedical concepts. UMLS concepts have a preferred name, a unique identifier, at least one semantic type, and a higher level semantic group. The semantic types in UMLS are based on categories such as organisms and chemicals. Within UMLS, a semantic network exists that is composed of semantic types and semantic relationships between types. Semantic groups provide a higher-order grouping of semantic types.

2.4 MedTagger

MedTagger is an extension of the cTAKES NLP pipeline [5] and is used to identify semantically viable information from clinical documents. [2] This pipelined system combines rule-based and machine learning techniques to extract concepts for each sentence from the Unified Medical Language System (UMLS). MedTagger uses lexical normalization and dictionary-based concept extraction according to an Aho-Corasick string matching algorithm [8] using the NLM controlled vocabulary thesaurus MeSH (Medical Subject Headings) and the UMLS Metathesaurus.

[1] https://lucene.apache.org/core/index.html

The accuracy of MedTagger has been evaluated in former work in the context of the CLEF 2013 shared task [2]. The analysis yielded a precision and recall of 94% and 77% for the relaxed matching mode.

For instance, in the sentence “To learn about the molecular etiology of strabismus, we are studying the genetic basis of 'congenital fibrosis of the extraocular muscles' (CFEOM).”, MedTagger will find the following UMLS concepts:

C0038379 Strabismus, NOS
C0016059 Fibroses
C1268995 Extraocular muscle
C0026845 Muscles

2.5 Data Fusion and Machine Learning (DFML)

In determining ranking of retrieved documents, data fusion [6] and machine learning techniques called by learning to rank [3] have been widely employed. Data fusion considers IR as the task of the ranking algorithm and combining features from multiple systems (features, components, and so on). In machine learning, IR can be considered as classification problem of relevance. Several algorithms such as support vector machine, decision tree and logistic regression can be utilized for classification. Several features we generated could be used for training both ranking algorithm of data fusion and classification model for machine learning.

3. Methods

Regarding to the TREC CDS 2015 challenge, our retrieval process tries to match 733,138 Pubmed Central articles with 30 given case vignettes for each task. As shown in figure 1, we extracted the document titles, keywords, body and abstract text as single attributes (section 3.1). Also, we retrieved the relevant UMLS concepts (section 3.2.1), and incorporated manual or search-engine enabled diagnosis identification (section 3.2.2) based on case summaries. The document features were placed in a searchable Lucene Index (section 3.3), and the UMLS concept based representation of the case summary was used as a Lucene search query. For the TREC challenge, we used three main approaches: Machine learning to determine the best combination of features (section 3.4.1), Query transformation using a health-focused custom Google search to derive diagnosis concepts from a clinical case summary (section 3.4.2), and Manual determination of the case diagnosis from the case summary by clinicians (section 3.4.3).

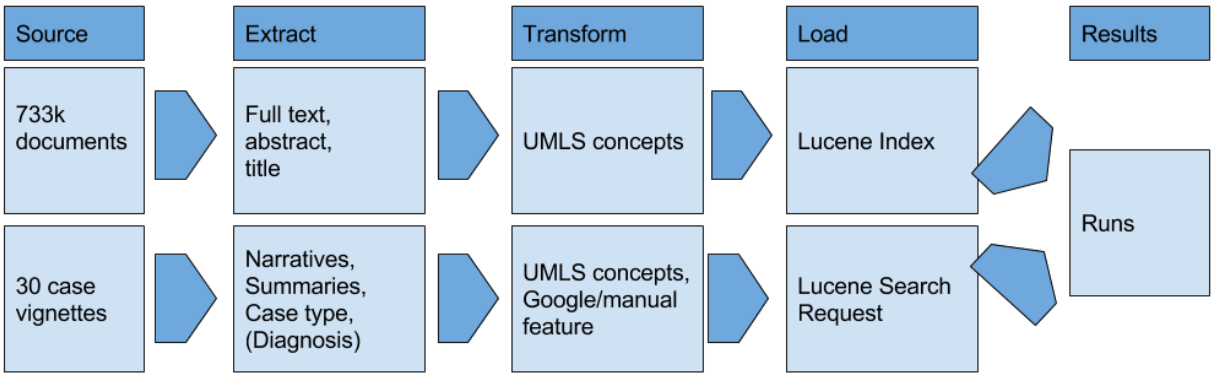

Figure 1. Overall architecture showing the data flows between inputs and result.

3.1 Feature extraction

Feature extraction is the initial step in our process. The articles' abstract, body, title and keyword text was extracted from the document XML, and then indexed using Lucene (version 5.10) as separate fields, with the exception that the abstract and body text were conflated into one field. After sentence splitting with SemRep version 2014 [4], UMLS concepts for abstracts and diagnoses were found using MedTagger. Additionally, the UMLS concepts derived from the abstract were indexed both as UMLS CUIs and as text derived from the UMLS concept name. The concepts were indexed in the same sentence order as they appear in the abstract.

In the search process, the matches between the query and Lucene fields were quantified using Lucene's basic scoring function. For machine learning, the individual Lucene calculated scores for each field were exported.

3.2 Innovative approaches

3.2.1 UMLS semantic filter

Using domain knowledge, we limited UMLS concepts from case narratives and articles to certain semantic types to reduce noise and focus on most relevant types of concepts. The following concept and semantic types were used:

- dsyn - Disease or Syndrome
- cgab - Congenital Abnormality
- neop - Neoplastic Process
- patf - Pathologic Function
- inpo - Injury or Poisoning
- mobd - Mental or Behavioral Dysfunction
- virs - Virus
- fngs - Fungus
- bact- Bacterium
- CUI = 'C0199176' (Preventive procedure)

Also, concepts from the semantic group CHEM were ignored in Part A. For Part B, we filtered out the CHEM group for the UMLS concept text but not from the UMLS CUIs.

3.2.2 Manual diagnosis / Google Feature

For the MDRUN and DFML runs we added a manual feature. We asked clinicians to propose a diagnosis for each of the cases based on information in the summaries, narratives and case type. The corresponding UMLS concepts were added as a case feature.

Because HAKT is designed as an automated run there is no manual input allowed. For query expansion using diagnosis terms, instead of manual annotation by a physician, we used the Google Custom Search Engine (CSE) to search for candidate diagnoses. The topic summaries served as the search input. To reduce noise the sources were limited to articles from health web sites: wikimedz.com, MedlinePlus, MayoClinic.org, WebMD.com and Wikipedia.org. The titles of the resulting hits were collected and then MedTagger was used to extract diagnosis concepts.

For example, for the case summary below, the hits on Figure 2 were found by the Google CSE:

A young woman in her second gestation presenting with anemia resistant to improvement by iron supplementation, elevated LDH, anisocytosis, poikilocytosis, hemosiderinuria and normal clotting screen.

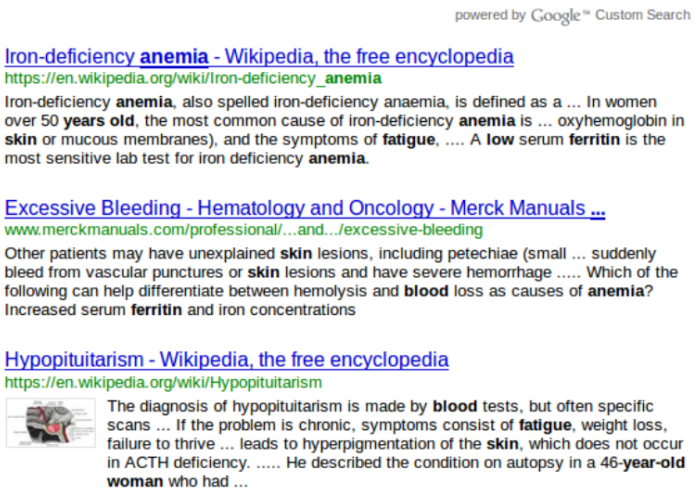

Figure 2. Results of the Google Custom Search Engine

The concepts extracted by MedTagger in these hits include:
C0240066 Iron deficiency, NOS
C0162316 Anemia, Iron-Deficiency
C0011155 Deficiency of
C0041782 Deficiency anemias
...

3.3 Lucene Index and Scoring

As noted in Table 1, our Lucene Index contains fields for UMLS concepts from the abstract. The concept UMLS CUI is in one field and the UMLS concept preferred name is in a second field. These fields are a string of text. Concepts are ordered by the appearance of the sentence containing the concept in the document text (abstract plus body). The keywords from the article body and title are also represented, both in a stem and non-stemmed field. The Lucene analyzer used for non-stemmed fields was the StandardAnalyzer class, and the analyzer used for stemming was the EnglishAnalyzer class.

Table 1. Index fields used for representing a document in a Lucene search

Field	Stop words removed	Stemmed	Type
Abstract UMLS Concepts	Yes	No	CUI
Abstract UMLS Concepts Preferred Name	Yes	No	Text
Document Title	Yes	No	Text
Document Title, Stemmed	Yes	Yes	Text
Document Keywords	Yes	No	Text
Document Keywords, Stemmed	Yes	Yes	Text
Document Text (abstract and body)	Yes	Yes	Text
Meets PubMed Clinical Query for Diagnosis	NA	NA	Boolean
Meets PubMed Clinical Query for Treatment	NA	NA	Boolean

In the search process, a filtered concept list derived from the Case Topic summary was used to search the fields noted in the table. We used the standard Lucene TF/IDF based scoring function to measure the match between Case Topic representation and document representation. The standard scoring function of Lucene uses a combination of the Vector Space Model (VSM) of Information Retrieval and Boolean model (BM) of Information Retrieval [7].

As input for machine learning, the Lucene similarity score was returned independently for each Topic summary and each field across the entire document set. Additionally, boolean values for filters were also returned.

3.4 Run configuration

Run summaries detailing the approaches used in each run can be found in Table 2.

Table 2. Summary of submitted runs

RunID	Query Version	Task Type	Run details
DFML	Summary	A	**Diagnosis, test, and treatment:** Data fusion + ordinary logistic regression (OLR) (1st: selecting 1,000 document by data fusion, 2nd: re-ranking by predicted probabilities of the OLR)
HAKT	Summary	A	**Diagnosis, test, and treatment:** Hierarchical adding of documents until collecting 1,000 documents by UMLS concept from Google search
MDRUN	Self-generated	A	**Diagnosis, test, and treatment:** UMLS concepts from manual queries created by physician with predefined template
DFMLB	Summary	B	**Diagnosis:** Data fusion + OLR without manual expansion (producing average of ranking scores by data fusion and predicted probabilities by OLR) **Test and treatment:** Data fusion + OLR with manual expansion for training data and given diagnosis by TREC for testing data (producing average of ranking scores by data fusion and predicted probabilities by OLR)
HMKTB	Summary	B	**Diagnosis:** Hierarchical adding of documents until collecting 1,000 documents by UMLS concept from Google search **Test and treatment:** Hierarchical adding of documents until collecting 1,000 documents by UMLS concept from given diagnosis and Google search
MDRUB	Self-generated	B	**Diagnosis;** UMLS concepts from manual queries created by physician with predefined template **Test and treatment:**

			Given diagnosis and corresponding UMLS concepts as queries

Details of each run configuration follows.

3.4.1 Data fusion and machine learning (run DFML / DFMLB)

The DFML run is a step-wise approach to combine data fusion to rank documents and machine learning to classify degree of relevance. As the TREC CDS challenge uses Normalized Discounted Cumulative Gain (NDCG) for measuring performance, the errors in the task are not only bounded by ranking errors, but also bounded by classification error. We made three variations for Tasks A and B: the way to produce final ranking; features used; and the level of classification.

For Task A, the linear combination method was used to select 1,000 relevant documents for the first step by fusing features, and the OLR was used to classify relevance of those selected documents for the second step to normalize those errors. TF-IDF similarity scores between title, keywords, or full-text and summary were main features for the machine learning process. Also TF-IDF similarity scores between the UMLS concepts from case summary and UMLS concepts from the original document text, Google expansion, or manual expansion were used. In terms of the level of relevance, we used multi-classification of three levels. The status of inclusion of related MeSH terms (diagnosis and therapy) in PubMed abstract was also utilized as one of features for both tasks.

For Task B, we used the TREC supplied diagnosis for test and treatment cases instead of our expert derived diagnosis. To prepare for Task B, UMLS concepts expanded by manual expansion were used in training.

We calculated the average of ranking scores by data fusion and predicted probabilities by OLR to offset the errors by two approaches and to produce final ranking for Task B. We also changed the multi-classification approach to binary classification to keep our approach simple so that some noises from multi-classification can be reduced.

For both tasks, the linear combination used the precision@200 by independent feature over the topic (diagnosis, treatments, and test) as its weight after conducting parameter sweeping. The class probabilities from OLR based on multi-classification were converted into ranking scores by assuming expected relevance score as sum of the class probabilities multiplied by graded relevance scale. The final ranks were produced according to the estimated relevance score. This approach leverages classical ranking algorithm and machine learning for classification to suit two major dimensions of information retrieval: ranking and classification.

3.4.2 Hierarchical + abstract + keywords +title (run HAKT/HMKTB)

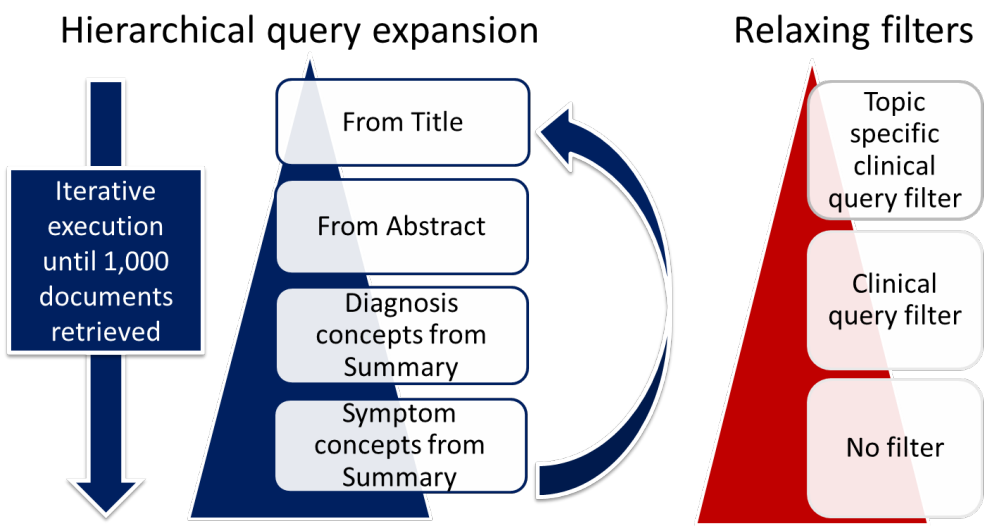

Figure 3: Description of the Semantic Hierarchical Iterative Approach. Blue cone represents the hierarchical search. First the title is used for matching to concepts in the query. Next, the abstract is used. Then diagnosis concepts from the summary are added to expand the query and the abstract and body is searched. Finally, symptoms are added to the query. The hierarchy is repeated with progressively relaxed filters based on MeSH article headings (red cone).

HAKT is a semantic hierarchical iterative approach using heuristically derived filters for diagnosis and symptoms. The algorithm uses a nested logic to prioritize different search strategies and is tuned by filtering. This hierarchical search strategy is enhanced by using a boolean query combination of a query from the hierarchy, a keyword search, a title search and a search with a term based on the case topic type. The two essential parts are summarized in Figure 3. The blue cone represents the hierarchical search. First the tile is used for matching to concepts in the query. Next, the abstract is used. Then diagnosis concepts from the summary are added to expand the query and the abstract and body is searched. Finally, symptoms are added to the query. The hierarchy is repeated with progressively relaxed filters based on MeSH article headings (red cone). The algorithm terminates when a thousand search results are found or both cones cannot be widened anymore.

Initially, UMLS concept preferred terms from the Google determined diagnosis are used in text based searches of the article's stemmed title and stemmed keywords. If there are no concepts for the Google-based diagnosis, then the stemmed text from the summary is used instead.

Next, the Hierarchical search is initiated. The hierarchical search makes use of the Lucene Boolean operator to join: a UMLS concept search, appropriate Topic type word search (e.g. a search with the word 'diagnosis' for cases with the 'diagnosis' type), stemmed title search and stemmed keyword search using the preferred terms of the UMLS concepts from the Google-diagnosis. The UMLS concept search follows the hierarchy of:

1. Google diagnosis UMLS concept CUIs vs. UMLS concept CUIs from the abstract,
2. Google diagnosis text vs. UMLS concept preferred names from document abstracts,

3. filtered UMLS concept CUIs from the case Summary vs. UMLS concept CUIs from the abstract,
4. UMLS concepts CUIs from the case Summary that are in the UMLS semantic group ‘DISO’ and the UMLS semantic type ‘sosy’ vs. UMLS concept CUIs from the abstract.

Additionally, a hierarchy of filters is applied in decreasing levels for each iteration of the Hierarchical search. The first filter utilizes topic specific PubMed Clinical Query indices. For the case topic of diagnosis the filter only allows documents satisfying the Clinical Query for diagnosis to be returned. For the cases with topics of treatment and test, returned documents must meet the Clinical Query for treatment. The next filter relaxes the Clinical Query constraint to allow any document in the union of PubMed Clinical Query results for diagnosis or test/treatment. The last filter iteration is a total relaxation of the filter to allow maximum recall, any article is allowed.

At each step through the Hierarchy, the filtered search results are appended in rank order to the final algorithm result. Each level of the Hierarchical search and filtering process is iteratively executed till 1000 documents have been outputted or the hierarchy has been exhausted. Then as a final effort to achieve 1000 returned documents, the stemmed summary text is searched against the stemmed document text.

3.4.3 Completely Manual (run MDRUN/MDRUB)

For this run, we attempted to manually model the cognitive task of determining the relevancy of a paper to a clinical case.

According to a predefined template,”[disease] + [type] + [any specific patient population constraints: age(child, adolescent, adult),…]”, a physician has created manual queries based on topic and task. The query is parsed into UMLS concepts using MedTagger, and the concepts are then used to search our Lucene index. Figure 4 illustrates the general pattern and an example.

Two Boolean queries are used for MDRUN. The first is more specific and the second is used to increase the number of documents returned.

The first query combines: 1) a UMLS concept CUI based search, using concepts from the manually identified disease, against UMLS concepts from article abstracts, 2) a second CUI based search using concepts derived from co-morbidities and additional constraints identified manually by a clinician against UMLS concepts from article abstracts, 3) three separate text based searches using the preferred names of the UMLS concepts used in 1) and 2) plus the Topic type against the stemmed text of the title, article body and abstract, or article keywords.

The second Boolean query is less specific and is used to add additional documents to achieve a return of 1000 documents. It is a text based search of the article body and abstract using the summary.

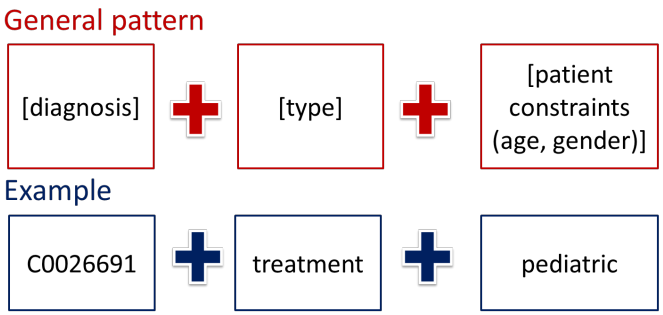

Figure 4. General Pattern and example of a query used for searching the Lucene index in MDRUN. The three elements used are the 'diagnosis' concepts, 'type' of the TREC topic (either diagnosis, treatment or test), patient population and constraints.

4. Results

4.1 Task A

Our best performing run for Task A was based on the combination of data fusion and machine learning approaches that utilized manual annotation of diagnosis by expert as one of main features (Table 3). The performance of our two best systems (DFML and MDRUN) was better than the median performance by all participants. The heuristic run of HAKT did not perform as well as other two approaches. The difference of infNDCG between our runs and median performance of all participants for Task A is summarized in Figure 5..

By applying one-way ANOVA analysis to our runs, we tested whether there were difference of performance among the different methods we used for Task A. With the one-way ANOVA test we could find that there was statistically significant difference in our runs with two measures (R-prec and Prec@10), but could not find significant difference with other two measures (infNDCG and infAP). For those two significant cases, we conducted additional pairwise comparisons using the Tukey method. In three pairwise comparisons among our runs, HAKT and DFML only had statistically significant difference in R-prec (p-value: 0.029) and in Prec@10 (p-value: 0.027). More details are summarized in Table 3.

Table 3. Task A evaluation results

Summary statistics				
Run ID	DFML	HAKT	MDRUN	Significance (p-value)
Processing type	Semi-Auto	Auto	Manual	NA
Number of topic	30	30	30	NA

Mean infAP	0.0776	0.0354	0.0725	0.15
Mean infNDCG	0.3019	0.1794	0.2925	0.087
Mean R-prec	**0.1889**	0.0936	0.1780	**0.020**
Mean Prec@10	**0.5133**	0.28	0.4867	**0.0184**

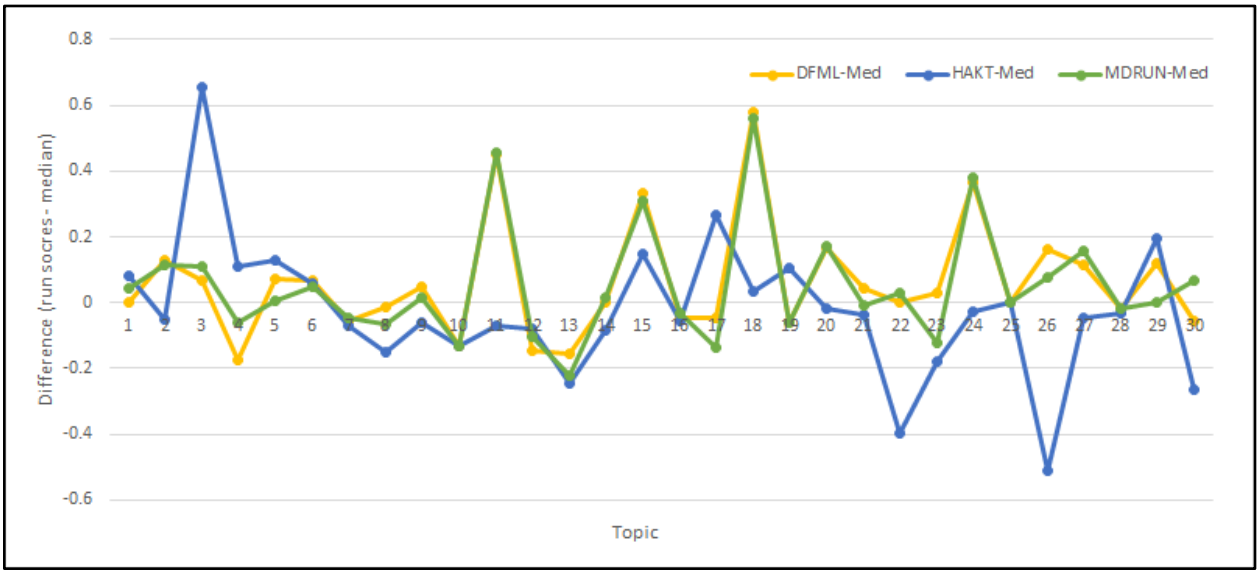

Figure 5. Difference from median infNDCG of all participants for Task A runs

4.2 Task B

Our best performing run for Task B was the MDRUB that is based on manual expansion of queries by expert (Table 4). The performance of our two best systems (MDRUB and HMKTB) was better than the median performance by all participants. The heuristic run of HAKT performed better than it did for Task A, and DFMLB, a combination of data fusion and machine learning approaches, did not perform as well as other two approaches. However, the best performance with different measures could be achieved by different runs. There was no one absolute outperforming run in Task B as we could observe in Task A. The difference of infNDCG between our runs and median performance of all participants for Task B is summarized in Figure 6. By applying one-way ANOVA analysis to our runs, we tested whether there was difference of performance among the different methods we used for Task B. We could not find that there was any evidence of difference in our runs. More details are summarized in Table 4.

Table 4. Task B evaluation results

Summary statistics

Run ID	DFMLB	HMKTB	MDRUB	Significance (p-value)
Processing type	Auto	Auto	Auto	NA
Number of topics	30	30	30	NA
Mean infAP	0.0485	0.06	0.0887	0.071
Mean infNDCG	0.2204	0.2796	0.3255	0.093
Mean R-prec	0.1889	0.0936	0.1780	0.0649
Mean Prec@10	0.38	0.5167	0.5133	0.182

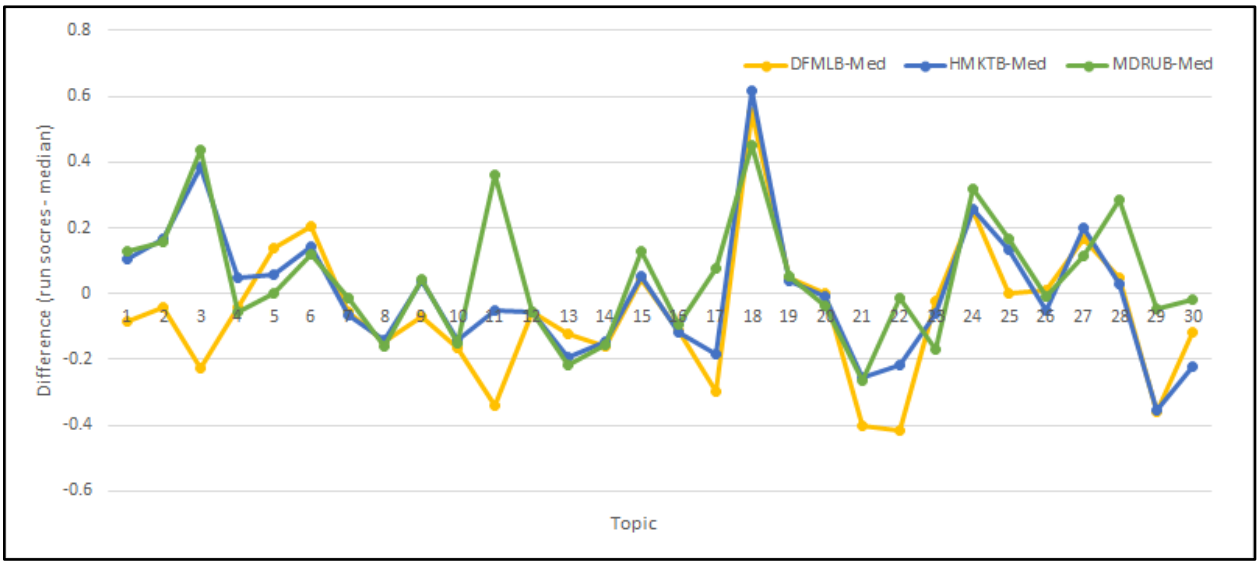

Figure 6. Difference from median infNDCG of all participants for Task B runs

5. Conclusions and Future Work

Our approach for the TREC CDS challenge 2015 utilized several innovative features including UMLS concepts, Google's custom search engine, and manual expansion by experts based on a predefined template. Query expansion by Google and by experts were submitted as independent runs and also utilized as a feature in the machine learning based run. The machine learning based run leveraged the ranking produced by data fusion and the predicted probabilities by machine learning to compensate errors caused by the two approaches. Two of our three runs outperformed the median performance measures such as infNDCG and infP@10 in average for both Task A and Task B.

In task B, HMKTB uses the same algorithm as HAKT, with the substitution of the TREC provided manual diagnosis for the google-automated diagnosis in the test and treatment topics. The performance improvement of HMKTB over HAKT indicates that manual diagnosis improves performance. This agrees with the observations that our algorithms that included concepts

derived from manual diagnosis in Task A, DMFL and MDRUN, perform better than HAKT. Together, these observations indicate that providing an accurate diagnosis representation is very important in representing a clinical case summary during information retrieval. As most electronic health records (EHR) include diagnosis by physician at point-of-care, this implicates how real world clinical information system should leverage this valuable clue while building a clinical information retrieval system. However, the overall performance of the runs in Task B was not high, suggesting that an accurate diagnosis alone is not sufficient for high performance.

In this shared task, we have applied the knowledge acquired from the 2014 dataset to the 2015 dataset, even though the cases for both years are different. As we now have an annotated dataset (qrel2015), we will test the generalizability of our approaches in applying what we learned from training data to new data. More experimental studies such as inclusion of article MeSH terms or weighting of UMLS concepts can help fine-tune our algorithm's performance. Also we would like to examine if cross-adoptive learning over different approaches (e.g., seed queries from manual expansion for refining Google CSE-based query expansion) can enhance clinical information retrieval.

6. Acknowledgements

This study was supported by the following National Library of Medicine (NLM) grants: 1R01LM011416-01, 5R00LM011389 and T15LM007124.

References

[1] Tang, H., & Ng, J. H. K. (2006). Googling for a diagnosis—use of Google as a diagnostic aid: internet based study. *Bmj*, *333*(7579), 1143-1145.

[2] Liu, H., Wagholikar, K., Jonnalagadda, S., & Sohn, S. (2013). Integrated cTAKES for concept mention detection and normalization. *Proceedings of the ShARe/CLEF Evaluation Lab*.

[3] Li, P., Wu, Q., & Burges, C. J. (2007). Mcrank: Learning to rank using multiple classification and gradient boosting. *Advances in neural information processing systems* (pp. 897-904).

[4] Rindflesch, T.C. and Fiszman, M. (2003). The interaction of domain knowledge and linguistic structure in natural language processing: interpreting hypernymic propositions in biomedical text. Journal of Biomedical Informatics, 36(6):462-477.

[5] Savova GK, Masanz JJ, Ogren PV, Zheng J, Sohn S, Kipper-Schuler KC, et al. Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. J Am Med Inform Assoc. 2010;17:507-13.

[6] Wu, S. (2012). *Data fusion in information retrieval* (Vol. 13). Springer Science & Business Media.

[7] Manning, C.D., Raghavan, P. and Schütze, H. (2008). Introduction to Information Retrieval, *Cambridge University Press*.

[8] Divita G, Browne AC, Rindflesch TC. Evaluating lexical variant generation to improve information retrieval. *Proceedings of the AMIA Symposium. 1998:775.*

[9] Bodenreider, O. (2004) The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Research.*