UCliDSS : *An Unsupervised Clinical Decision Support System for Text (Demo Paper)*

Tahir Dar
International Institute of Information Technology
Bangalore, 560100
Karnataka, India
tahir.dar@iiitb.org

Sumant Kulkarni
International Institute of Information Technology
Bangalore, 560100
Karnataka, India
sumant.k@iiitb.org

Srinath Srinivasa
International Institute of Information Technology
Bangalore, 560100
Karnataka, India
sri@iiitb.ac.in

Ullas Nambiar
EMC Corporation
Bangalore, 560048
Karnataka, India
Ullas.Nambiar@emc.com

ABSTRACT

We present our tool UCliDSS, an Unsupervised Clinical Decision Support System for textual data. UCliDSS is aimed at building a retrieval engine to retrieve documents from a bio-medical data-set to provide decision support for medical professionals. In this work, we use a term co-occurrence graph (TCG) based approach augmented with Solr to build the document retrieval engine (UCliDSS). The TCG is built from the documents of a given bio-medical data-set and is used for query expansion using a variant of random walk algorithm. The expanded query is given to Solr as input to retrieve relevant documents.

Keywords : Unsupervised Clinical Decision Support, Bio-medical document retrieval, Information retrieval

1. INTRODUCTION

Medical professionals (predominantly doctors) usually look for decision support systems to help them in different task of their daily routine like diagnosis, treatment, prescription of relevant tests and prescription of relevant effective medication. There can be different types of text documents like earlier diagnosis reports, discharge summaries, conference and journal papers which help the physician and specialist doctors make decisions about the case at hand. However, due to the sheer number of these documents, it has become almost impossible for a medical professional to manually search the relevant documents for a given problem. It becomes even more difficult problem to filter the document based on the aspects like diagnosis, treatment, test and so on. This puts forth an immediate need for building a information retrieval system for medical data. *UCliDSS* is such a system developed to serve the purpose of clinical decision support.

The motive of UCliDSS is to retrieve relevant medical documents from a large dataset, which help in answering generic clinical questions about medical conditions of patients. UCliDSS also filters the retrieved medical documents according to particular aspect (such as diagnosis, treatment, symptoms). We assume that UCliDSS works on a large corpus of medical text documents that contains information related to various medical and clinical cases. The input query to UCliDSS is a medical text scenario about the case in hand. The input query may narrate medical condition of a patients, may describe the symptoms. Additionally, it also mentions the aspect at which the user is interested. The medical documents to be retrieved from the corpus should be relevant to the input query in that aspect of the case.

For the information needs of physicians, the input queries can be put according to the many common generic clinical questions type. Some input query types are as in Table 1. The type determines the kind of question we would ask about the given case at hand.

Table 1: Some types of input queries.

Type	Generic Clinical Question
Symptoms	Symptoms of the disease ?
Diagnosis	Diagnosis of a disease ?
Prescription	Medicines prescribed to a disease ?

The input query is expected to be free text. It may consists of a current case report, summary investigation, or history of patients condition. It's *type* (aspect) is one of the generic clinical question (like mentioned in table 1). We expect the relevant text documents retrieved from the corpus.

In this work, we describe our approach in section 3. The section 4 describes the tool *UCliDSS*. Further, in section 5, we discuss an use case of the tool, where we participated in

a clinical data challenge using UCliDSS. Finally the future scope of work for UCliDSS is discussed in section 6.

2. RELATED WORK

There have been multiple approaches to information retrieval in medical domain. The most predominate ones are listed below for the sake of completeness. Collen and Flagle [1] proposed a primitive command line based medical information system which is based on a comprehensive database. However, the most predominant systems appeared post 2000. Mao and Chu [7] used phrase based vector space model to index and retrieve the medical documents. This problem suffers with the same issue of scale and issues with dimensionality reduction. Liu and Chu [6] proposed a medical IR system which used domain knowledge based query expansion. Even though it looked similar to our approach, they used human constructed thesaurus. It is a tedious task to build such domain knowledge and thesaurus. Zuccon et.al [12] came up with approach which exploited medical hierarchies for information retrieval. However, this approach relies on subsumption hierarchies which are very difficult to build. Holzinger et.al [2] presented a survey on the biomedical text mining approaches.

Even though our approach also requires background knowledge for medical information retrieval, the process of building the background knowledge is completely unsupervised. Our approach depends on the co-occurrence graph based text mining approaches [9, 5, 3, 8, 4].

3. THE METHOD

All the nouns are extracted from the given biomedical text document and a term co-occurrence graph (TCG) is built from these terms. The term co-occurrence graph represents the knowledge of the system. The TCG is treated as the background knowledge of the systems and is used for query expansion of the input query.

The text in the biomedical documents is used to create the term co-occurrence graph. We extract nouns from each text document using statistical noun phrase extraction techniques. For each paragraph having k unique nouns, we create a clique of size k with the nouns as nodes and the edge weight being one. These cliques are merged with each other to create a single large TCG. The detailed explanation of the TCG creation are in [9, 5]. We then convert it into a generatability graph (as discussed in [9, 5]).

We use the TCG for *query expansion* of the input query. Here, we expand the terms in the given input query to get more relevant terms. From the text of the input query, the nouns are extracted which are called *input query terms*. To incorporate the aspect (like diagnosis, test, symptoms) of the input query, we treat the term representing the *type* also as a part of the input query terms.

For each of the *input query term* we get their degree in TCG. We assume that a term with more neighbors is a common term and hence has lesser importance. Hence we calculated the importance score for term t as $I(t)$ as the inverse of the number of neighbours of t. The TCG is queried for the *semantic context of closure* (SCC) of the given *input query term*. The SCC of a terms on TCG returns the induced sub-graph of all the first hop neighbors of the given set of input terms. We run a variant of random walk algorithm [10] on SCC. The random walk algorithm is explained in algorithm 1. This algorithm, on stationary distribution, leaves each node in SCC with some amount of cash. Once we finish running the random-walk for all the terms in *input query terms*, we add the cash for each unique term and choose the top 20% nodes with most cash sum as the query expansion. We assume that top twenty percent of the terms represent eighty percent of the content importance. This selection of smaller number of query terms also helps in reducing noisy, unwanted query terms. To make sure that the keywords in the *aspect* are not missed out due to lesser cash sum, we append all the nouns extracted from the aspect to this list of top 20% terms. We call these terms as *expanded query* set.

Data: Generatability graph G_l, seed terms t, its importance score s, a bound on the maximum cash difference between two consecutive iterations in the random walk max_cash_dif
Result: Cash history H of all the terms in G_l
for *all nodes* $i \in G_l$ **do**
  $H[i] \longleftarrow 0$; $c[i] \longleftarrow 0$;
end
$c[t] \longleftarrow s$;
while *there exists a term* $\boldsymbol{m}$ *for which* $abs(P[m] - H[m]) \geq max\_cash\_dif$ **do**
  $historysum \longleftarrow 0$;
  $P \longleftarrow H$;
  for *each node i picked at random from* G_l **do**
    $H[i] \longleftarrow H[i] + c[i]$;
    for *each node j in N (i) and* G_l **do**
      $c[j] \longleftarrow c[j] + (\Gamma_{i \to j} \times c[i])$;
    end
    $historysum \longleftarrow historysum + H[i]$;
    $c[i] \longleftarrow 0$;
  end
  for *each node i in* G_l **do**
    $H[i] \longleftarrow \frac{H[i]}{historysum}$;
  end
end
// Stationary distribution reached
return H;

Algorithm 1: The Random walk algorithm runs random walk starting from the given node t with seed cash s on the given graph and produces the cash history H as the output.

To retrieve files from the dataset using the terms retrieved from term co-occurrence graph after query expansion, the approach is augmented with Solr a search engine[1]. Here, the given text corpus is parsed and indexed into Solr. The *expanded query* is given to Solr and relevant documents are retrieved. The topmost documents are found to be more relevant to the input query terms.

4. THE TOOL

UCliDSS is developed in ruby language and is a desktop application. The tool contains a parser which can extract text from html, xml or single column pdf file. The terms(nouns) are extracted from the unstructured text documents using an algorithm which has extended features from the *rb-brill-tagger*[2]. UCliDSS is integrated with Solr as dis-

[1]http://lucene.apache.org/solr
[2]https://github.com/taf2/rb-brill-tagger.git

cussed in section 3.

The term co-occurrence graph is stored in Agama a graph database[3] where each term is a node and each directed edge has a weight of generatability of the target term from source term.

We have a command line interface where we can perform basic operations. The co-occurrence graph can be created from a given corpus as below.

```
$uclidss --tcg <folder-having-documents>
```

UCliDSS also takes the input query and the aspect to retrieve the set of relevant documents.

```
$uclidss --query <query> --aspect <aspect>
```

5. TREC CLINICAL DATA CHALLENGE WITH UCLIDSS

We found that this framework was suitable to be applied on TREC Clinical Decision Support Track (Trec-cds-2015 challenge[4]). Here the task was to retrieve documents which can aid the medical experts in their daily routine tasks.

The dataset in Trec-cds contained seven lakh thirty three thousand documents (journal articles) in nxml format. Each document(journal article) describes a medical outcome. Many of these articles are relevant to the medical experts in certain aspects like *diagnosis*, *test prescription* and *treatment*. The task was to retrieve the most relevant of these journal articles for a given input query (with aspect).

There were thirty input queries in an xml file. Each input query (also known as topic) in Trec-cds challenge is of one of three types (aspects) – Diagnosis, Treatment and Test.

As the document corpus was huge, it was practically impossible to create a TCG of all the journal articles. We however assumed that a TCG created using a significant number of randomly sampled journal articles would be a good representative of the complete corpus. We choose around 13000 random documents and generated the term co-occurrence graph from it. This was the *background knowledge* for the UCliDSS.

UCliDSS is augmented with Solr for retrieving the journal articles. All journal articles were parsed and indexed into Solr a search engine. Solr is used after performing query expansion using random walk algorithm.

The experiments of UCliDSS were based on several factors so that we can retrieve and compare between multiple results. The experiments were done on an i7 4th generation intel machine with 30 gb ram. The execution time to get results for given 30 input queries (topics) was some 1.5 hours. The following experiments were performed.

1. The Topic as shown below consists of topic Number, type, description and summary.

   ```
   <topic number="12" type="test">
   <description>
   A 44-year-old man was recently in an
   automobile accident where he sustained
   a skull fracture.  In the emergency room,
   he noted clear fluid dripping from his
   nose.  The following day he started
   complaining of severe headache and fever.
   Nuchal rigidity was found on physical
   examination.
   </description>
   <summary>
   A 44-year-old man complains of severe
   headache and fever.  Nuchal rigidity was
   found on physical examination.
   </summary>
   </topic>
   ```

   In the first experiment textual information in both "description" and "summary" and extracted input query terms from the same and also appended the type of the topic . The sementic context of closure (SCC) for the input query terms was itself a large induced sub graph and it was heavy in terms of execution time. Hence, we decided to choose only certain percentage C of neighboring nodes with higher generatability (from the given *input query term)* to build the SCC. Even though there was no rational way of identifying what would be the right percentage of nodes C to generate SCC, we experimented with three values for C, 1%, 5% and 10%. Through the manual evaluation we realized that results with $C = 5\%$ and $C = 10\%$ were almost the same and were better than the results of $C = 1\%$. Hence, we chose $C = 5\%$ as the cutoff to generate SCC.

2. In the second experiment we extracted terms only from "description" only (and not the "summary") and performed the similar retrieval of documents as explained in the first experiment. We chose $C = 1\%$, $C = 5\%$ and $C = 10\%$ to genrate the sementic context of closures (SSC). Once again we found out that $C = 5\%$ was better due to its better quality of results even with lower number of nodes in SCC.

3. In third experiment the similar experiment as second was repeated extracted terms only from "summary" only (and not the "description") with value $C = 5\%$.

4. The above was the task A of the Trec-CDS 2015. For, Task B we were given an extra field called diagnosis. It looked like,

   ```
   <diagnosis>Bacterial Meningitis</diagnosis>
   ```

   Here, In Fourth experiment in addition to the summary filed, we also appended the data in the diagnosis filed and then preformed the query expansion. The rest of the execution remained the same.

To perform the following experiments, we had to write a small wrapper which understood the different fields in the test cases given. The results of the experiment 3 and 4 were submitted for the competition. The results have been evaluated by medical experts.

5.1 Results of TREC-CDS Challenge

The evaluation results of TREC-CDS is done based on the paper [11] which describes two methods of large scale retrieval evaluation, infAP(inferred Average Precision) and infAPNDCG(inferred Average Precision Normalised Discounted Cumulative Gain). In addition for our results R-Precision

[3]https://github.com/arrac/agama.git
[4]http://www.trec-cds.org/2015.html

Table 2: tab: Evaluation Results of Trec-cds

Experiment Number	Input Query	Mean infAP	Mean infNDCG	Mean R-precision	Mean Precision@10
1	Summary + Description	0.0402	0.1929	0.1592	0.3000
2	Description	0.0322	0.1772	0.1445	0.2733
3	Summary	0.0277	0.1534	0.1169	0.2500
4	Summary with diagnosis	0.0450	0.2145	0.1577	0.3233

and Precision@10 is also calculated which are known measures of information retrieval evalvation. The table 2 shows the measures calculated for our results on TREC-CDS data and input queries. All the above measures for the information retrieval are based on a ranked relevancy file which contains ranked relevant document ids for each input query. The relavancy file for TREC-CDS data has been provided by NIST. Based on the relavancy file the measures in table 2 have been calculated.

6. FUTURE WORK

The future work is to develop an optimized algorithm in terms of execution time to create a term co-occurrence of all the documents in a large corpora. The document-id of the document will be made to co-occur with all the terms/concepts of the document in the co-occurrence graph. The cash leaking random walk on the induced sub-graph of the neighbours of the input terms will be applied. Once the stationary distribution is achieved, system will identify the journal-ids which have accumulated the most cash history and declares them as the relevant documents for input query. Thus the role of Solr a search engine will be reduced. We also intend to implement a distributed version of the algorithm.

7. REFERENCES

[1] M. F. Collen and C. D. Flagle. Full-text medical literature retrieval by computer: a pilot test. *JAMA*, 254(19):2768–2774, 1985.

[2] A. Holzinger, J. Schantl, M. Schroettner, C. Seifert, and K. Verspoor. Biomedical text mining: State-of-the-art, open problems and future challenges. In *Interactive Knowledge Discovery and Data Mining in Biomedical Informatics*, pages 271–300. Springer, 2014.

[3] S. Kulkarni and S. Srinivasa. Sortinghat: a deep matching framework to match labeled concepts. In *Proceedings of the 20th International Conference on Management of Data*, pages 134–137. Computer Society of India, 2014.

[4] S. Kulkarni, S. Srinivasa, and R. Arora. Topic expansion using a term co-occurrence graph. Technical report.

[5] S. Kulkarni, S. Srinivasa, J. N. Khasnabish, K. Nagal, and S. G. Kurdagi. Sortinghat: A framework for deep matching between classes of entities. In *Data Engineering Workshops (ICDEW), 2014 IEEE 30th International Conference on*, pages 90–93. IEEE, 2014.

[6] Z. Liu and W. W. Chu. Knowledge-based query expansion to support scenario-specific retrieval of medical free text. *Information Retrieval*, 10(2):173–202, 2007.

[7] W. Mao and W. W. Chu. The phrase-based vector space model for automatic retrieval of free-text medical documents. *Data & Knowledge Engineering*, 61(1):76–92, 2007.

[8] A. R. Rachakonda, S. Srinivasa, S. Kulkarni, and M. Srinivasan. Mining analytic semantics from unstructured text. Technical report.

[9] A. R. Rachakonda, S. Srinivasa, S. Kulkarni, and M. Srinivasan. A generic framework and methodology for extracting semantics from co-occurrences. *Data & Knowledge Engineering*, 2014.

[10] M. Yazdani and A. Popescu-Belis. A random walk framework to compute textual semantic similarity: a unified model for three benchmark tasks. In *Semantic Computing (ICSC), 2010 IEEE Fourth International Conference on*, pages 424–429. IEEE, 2010.

[11] E. Yilmaz, E. Kanoulas, and J. A. Aslam. A simple and efficient sampling method for estimating ap and ndcg. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 603–610, New York, NY, USA, 2008. ACM.

[12] G. Zuccon, B. Koopman, A. Nguyen, D. Vickers, and L. Butt. Exploiting medical hierarchies for concept-based information retrieval. In *Proceedings of the Seventeenth Australasian Document Computing Symposium*, pages 111–114. ACM, 2012.