Unsupervised Mapping of Sentences to Biomedical Concepts based on Integrated Information Retrieval Model and Clustering

Mi-Young Kim, Qing Dou, Osmar R. Zaiane, Randy Goebel
Department of Computing Science, University of Alberta, Canada
{miyoung2, qdou, zaiane, goebel}@cs.ualberta.ca

ABSTRACT

Structured information revealed by manual annotation of disease descriptions with UMLS meta-thesaurus concepts, can provide high-quality reliable data sources for the research community. While progress in both extent and annotation has been made, only a limited scope of diseases has been annotated, largely because of the required human resources. Since annotating text is time consuming and the variation of disease descriptions makes the annotation task difficult, it is useful to develop systems for automatic mapping of biomedical sentences into an ontology. Our goal is to automatically map biomedical sentences into UMLS disease concepts. Previous methods including statistical methods, are still weaker than dictionary-based simple matching methods. To consider an alternative to both, we demonstrate how the mapping problem can be viewed as a document retrieval problem: under this perspective, the mapping integrates information based on a language model, document frequency, and distance measures. Our improvements are based on a three-step method using information retrieval and clustering. In the first step, we retrieve the top-10 ranked relevant UMLS concept entries using an integrated information retrieval model. In the second step, we cluster the retrieved concept entries according to shared words. In the final step, we select one answer for each cluster using a threshold. Our experiments are promising, and on typical data show a precision of 73.28%, recall of 77.51%, and F-measure of 75.34% significantly outperforming previous methods based on statistics, dictionaries, and the MetaMap by 6.95 to 9.95 percent.

Categories and Subject Descriptors

I.2.1 [**Artificial Intelligence**]: Application and Expert Systems – *Natural language interfaces, Medicine and science.*

General Terms

Algorithms, Experimentation, Languages.

ACM-BCB 2010, Niagara Falls, NY, USA

Keywords

Text mining, mapping of biomedical terms, information retrieval, bioinformatics

1. INTRODUCTION

Text mining in molecular biology or medicine – defined as the automatic extraction of information about genes, proteins, diseases, treatments and their functional relationships from text documents – has emerged as a hybrid discipline at the intersection of the fields of information science, bioinformatics and computational linguistics [1]. Literature mining in bioinformatics broadly consists of two parts. One part is the identification of biological or medical entities, and the other is mining for interactions and relations amongst those entities [2-7]. For the literature mining of medical records, the medical ontology known as Unified Medical Language System (UMLS) [8, 9] enables physicians to classify signs, symptoms, and diseases using accepted medical concepts. The UMLS integrates over 2 million names for some 900,000 concepts from more than 60 families of biomedical vocabularies, and includes 12 million relations among these concepts. Vocabularies integrated in the UMLS Metathesaurus include the National Center for Biotechnology Information (NCBI) taxonomy, Gene Ontology, the Medical Subject Headings (MeSH), Online Mendelian Inheritance in Man (OMIM) and the Digital Anatomist Symbolic Knowledge Base. Our hypothesis is that, combined with an integrated information retrieval method, the UMLS is a powerful and appropriate tool for automatically mapping disease names with variant forms into one concept.

Even though manual annotation provides highly reliable data sets, it is time consuming for biologists/physicians to annotate a large volume of biomedical sentences. Therefore we need automatic methods to detect biomedical terms in sentences, without requiring a large annotated corpus. There is some previous work to detect genes/proteins in sentences using annotated data [2-4]. However, for the detection of diseases, a large volume of annotated corpora is not currently available in the open domain. In order to improve the general situation with the annotation of disease descriptions to a common vocabulary, our goal here is to automatically and in an unsupervised way, map biomedical sentences into UMLS disease concept IDs according to the corresponding disease terms. Previous research either requires a large volume of annotated corpora as training data, or uses extensive linguistic knowledge [10]. Even though there is some previous work that provides their own custom statistical methods, they perform even more poorly than alternative

dictionary-based simple matching methods. In view of the seriously limited volume of training data in the domain of disease name recognition, we develop an information retrieval technique to find disease names in the UMLS metathesaurus, based on identifying those that are most related to a given zone (usually a sentence) in free text. We view the problem as determining likelihood of a sentence based on a list of candidate disease names. In the end, only the most likely disease names are chosen as final answers.

The originality of our method is two-fold. First, to our knowledge, this is the first time that the problem is modeled as a *generation process* based on information retrieval techniques using a language model. Second, the method does *not* require any preprocessing of sentences into noun phrases or pre-identification of disease terms, but jointly identifies and maps disease terms mentioned in a sentence to the UMLS metathesaurus. To accomplish this, we develop an integrated model by combining the conventional IR model, document frequency information, and a heuristic distance measure.

To improve performance in mapping sentences into UMLS concepts, we propose a three-step method based on an information retrieval technique and clustering. In the first step, we retrieve the top-10 ranked relevant UMLS concept entries using an integrated model by combining information retrieval technique based on a language model, distance information between words, and document frequency measures. In the second step, we cluster retrieved concept entries into group concepts based on the common words that they share. In the final step, we select one concept for each cluster based on our ranking score and a heuristic cut-off threshold.

Through experiments, we show that our proposed method significantly outperforms existing methods, and provide an analysis of how each step contributes to that overall performance.

The remainder of the paper is organized as follows. Section 2 presents previous work on identifying and mapping biomedical terms into an ontology. Section 3 explains the details of our three-step method. Section 4 describes the data used for our experiments and presents experimental results which demonstrate the performance of our three-step method. Finally, we provide our conclusions.

2. Previous Work

Automatic annotation methods for biomedical terms have been studied extensively in recent years, including the mapping of text-phrases to UMLS concepts [10,11]. Most of these approaches are focused on automatic indexing of biomedical literature, and have proved inadequate for processing annotations of high-throughput datasets [12,13]. It has also been shown that for the task of identifying concepts from annotations of high-throughput datasets, simple methods perform as well or better than MetaMap [12,13,14,15]. In previous work, some approaches of S. Gaudan et al. [16,17] are based on the identification of weighted words that compose terms denoting ontology concepts. They integrate two new aspects in their scoring method: the proximity between words in text and the amount of information carried by each individual word. Their method is a statistical method based on specificity, evidence, and proximity. Specificity and evidence are based on the frequency of a word in a corpus, and proximity is based on the distance between words in a corpus. They adopt TFxIDF used in information retrieval to measure the evidence and specificity, but the performance is worse than dictionary-based simple matching methods, and they do not use any threshold methods to choose relevant concepts among the ranked concepts.

F. Mougin et al. [18] mapped gene terms into UMLS based on normalization of words using UMLSKS API and exact match. A.Mottaz et al. [19] tried to map disease names into MeSH terminology. They used manually curated disease annotations to extract disease names, and applied exact and partial match to map the disease names into MeSH. To consider the information content of each word, they applied a TFxIDF weighting schema.

MetaMap is viewed as the state of art in mapping biomedical terms in free text to UMLS concepts. It first extracts noun phrases from a sentence based on syntax analysis. Then, synonym lexicons generated by experts are applied to generate variations of the extracted noun phrases. In the end, candidates are found from UMLS ranked by their matching score. The rules for generating those scores are purely heuristic.

J. Hakenberg et al. [20] introduced a method based on context models. In processing texts related to genomic analysis, they consider functions, processes, locations and tissue specificities as gene context. When trying to disambiguate a specific gene mention in free text, those kinds of information are compared with contexts that co-occur with the focus gene in the text. It is possible that different names are used to describe the same gene, as a result, to normalize variations in gene names, previous work has applied some heuristic rules [21]. Inspired by such work, Y. Tsuruoka et al. [22] tried to learn such rules automatically by minimizing ambiguity and variability of a lexicon. Their method shows significant improvement in matching rate for gene names. However, their learned rules are not general enough for handling disease names.

Some other machine learning approaches have also been investigated. H.W. Chun et al. [23] used a maximum entropy-based method to filter candidate disease names found by dictionary-based methods. Various features are selected, e.g., context words, part of speech tag, word affix, etc. M. Bundschus et al. [28] tried cascaded CRF using various features based on contexts, dictionary and orthogonal form to detect disease terms and the functional relations between them, and they need annotated data set for training. In the methods of A. Neveol et al. [24], a priority model [25] was used to find noun phrases that are possibly disease names. However, the mapping process is still done with the MetaMap program.

We conclude that most previous work uses simple exact or partial matching based on a dictionary, and performs deep preprocessing such as noun phrase detection and normalization of variant words. Some statistical methods try TFxIDF to measure the information content of a word as used in the information retrieval field, but show poorer performance than the dictionary-based matching methods.

We approach the mapping problem of sentences into an ontology as an information retrieval problem and show that the performance can be greatly improved by applying a clustering technique. We regard a biomedical sentence as a query, and a UMLS ontology entry as a document, and try to apply a language

modeling-based information retrieval method as currently used in the document retrieval field.

The language modeling approach to information retrieval [26] directly implements the following idea: a document is a good match to a query if the document model is likely to generate the query, which will in turn happen if the document often contains the query words. This model shows better performance than those which use TFxIDF and BM25 weights.

Our approach is to infer a language model for each concept entry and to estimate the probability of generating the query according to the model. We then rank the concepts according to these probabilities. Based on the IR model and clustering, this paper proposes a three-step based method for mapping sentences into UMLS. In the next section, we explain our method in detail.

3. Method

Our method is summarized as follows. To deal with the variation between plural and singular forms, we simply truncate nouns ending with '-(e)s' by trimming the '-(e)s' suffix (stemming). Our method does not require other natural language processing strategies such as normalization or noun-phrase recognition.

We then map sentences to UMLS concepts in three steps. In the first step, we apply our integrated model which combines a language model-based information retrieval with a distance measure and document frequency measure to generate the top 10 candidate concepts. In the second step, we cluster the retrieved concepts according to the common disease-related words. Finally, we select one concept from each cluster based on ranking and a cut-off point. Details of each step will be presented in the following subsections.

3.1 Step 1: Retrieval of Relevant UMLS Disease Concepts

We consider each input sentence as a query and UMLS entries as documents. We infer a language model for each UMLS concept entry, and rank each related entry according to how likely it generates the input sentence based on its language model.

However, our model is different from traditional IR models in three ways. First, the characteristics of a query and a document in our experiments are different from those in the traditional information retrieval field, since a query is longer than a document in our model, and the frequency of most words in a document is uniform.

Second, our purpose is to retrieve all and only matched disease concepts for a query sentence, while the purpose of traditional document retrieval is to retrieve a list of documents ranked by their relevance to a query. In traditional IR formulation, for a word *w* that is included in a document but not included in a query, one assigns a penalty by computing the probability that the language model does not generate *w* based on the term frequency. There is some risk of assigning a penalty based on only term frequency in a document for the words that have different information content, as measured by document frequency. To assign a small penalty to the words that are common to the domain but do not have a large amount of information content (e.g. the words 'disease', 'disorder', and 'symptom'), we add a document frequency measure. The document frequency of a word helps determine its information content: the smaller the document frequency of a word is, the bigger information content it has.

Finally, our purpose is to detect medical terms in a sentence. Traditional IR does not consider distance information. In term detection, the distance between words of a term *t* provides a clue on the likelihood that those words belong to a common term. Therefore, we add a distance measure. In the following subsections, we will explain our method in detail.

3.1.1 Information Retrieval Based on a Language Model

We consider each input sentence as a query and UMLS entries as documents. Then, we would like to estimate $\hat{p}(Q \mid M_d)$, the probability of the query given the language model of document *d* as follows.

$$\hat{p}(Q \mid M_d) = \prod_{t \in Q} \hat{p}(t \mid M_d) \times \prod_{t \notin Q} 1.0 - \hat{p}(t \mid M_d)$$

The first term is the probability of generating words in the query and the second term is the probability of not generating other terms. The detailed probabilities for $\hat{p}(Q \mid M_d)$ are defined as follows:

$$\hat{P}(t \mid M_d) = \begin{cases} p_{ml}(t,d)^{(1.0-\hat{R}_{t,d})} \times p_{avg}(t)^{\hat{R}_{t,d}} & \text{if } f_{(t,d)} > 0 \\ \dfrac{cf_t}{cs} & \text{otherwise} \end{cases},$$

$$\hat{R}_{t,d} = \left(\frac{1.0}{(1.0+\bar{f}_t)}\right) \times \left(\frac{\bar{f}_t}{(1.0+\bar{f}_t)}\right)^{tf_{(t,d)}},$$

$$\hat{p}_{ml}(t,d) = \frac{tf_{(t,d)}}{dl_d},$$

$$\hat{p}_{avg}(t) = \frac{\left(\sum_{d_{(t \in d)}} p_{ml}(t,d)\right)}{df_t}.$$

$\hat{p}_{ml}(t \mid M_d)$ shows the maximum likelihood estimate of the probability of term *t* under the term distribution of document *d*, where $tf_{(t,d)}$ is the raw term frequency of term *t* in document *d* and dl_d is the total number of tokens in document *d*. *cft/cs* is the background probability for the document that is missing one or more of the query terms, since we do not want to assign 0 for $\hat{p}(t \mid M_d)$ of this document, where *cft* is the raw count of term *t* in the collection and *cs* is the total number of tokens in the collection. $\hat{p}_{avg}(t)$ is the estimate of the probability of the word *t* from a larger volume of data. $\hat{R}_{t,d}$ is a risk function based on a geometric distribution, selected to benefit from the robustness of the estimator $\hat{p}_{avg}(t)$ and to minimize the risk of using the estimator. For more details on each probability, refer to Ponte and Croft [26].

<original sentence>

Using dna of a patient with piebaldism mental retardation and multiple congenital anomalies associated with a 46,, xy, del(4 karyotype we carried out quantitative southern blot hybridization analyses of the kit gene and the adjacent pdgfra genes.

<top-10 ranked results>

RANK	SCORE	CONCEPT_ID	CONCEPT_ENTRY
<1>	5.944104e-91	C0000772	anomalies congenital multiple
<2>	1.130237e-91	C0080024	piebaldism
<3>	1.074931e-91	C0025362	mental retardation
<4>	4.237612e-93	C0000768	congenital anomalies
<5>	4.126886e-102	C0004936	disorder mental
<6>	5.574557e-103	C0242354	congenital disorder
<7>	1.603987e-107	C0494422	other mental retardation
<8>	6.323268e-109	C0158795	other congenital anomalies
<9>	6.210680e-113	C0275544	congenital infection
<10>	3.192410e-113	C0037268	congenital skin anomalies

Figure 1. Example of concept retrieval

3.1.2 Document Frequency Information

In the IR formulation of subsection 3.1.1, for the word *t* that is included in a document but not included in a query, we assign a penalty by computing the probability that M_d does not generate *t* based on the term frequency. However, in our task, the term frequency of most terms in a document is uniform. So, there is some risk of assigning a probability measured by only term frequency for words with different information content. To assign a small penalty to the domain-specific common words, we add a document frequency measure which is:

$$DF(Q) = \prod_{t \notin Q} \frac{df(t)}{|D|},$$

where *df(t)* is the frequency of documents that contain *t*, and $|D|$ is the number of all documents.

3.1.3 Distance information

We need to consider distance among words to detect whether the words in a query indicate one common term. We modify the distance measure of S. Gaudan et al. [16] , as follows:

$$dist(d,Q) = \frac{current\_dist(d,Q)}{min\_dist(d,Q)}$$

Let *W* be the set of words of a document *d* found in a query sentence *Q*. Then W is:

$$W = tok(Q) \cap tok(d),$$

where *n* is the number of words in *W*. Then,

$$min\_dist(d,Q) = \sum_{i=0}^{n-1}\sum_{j=0}^{n-1}|i-j|,$$

$$current\_dist(d,Q) = \sum_{w_i \in W,\ w_j \in W} |position(w_i,Q) - position(W_j,Q)|$$

,

where $position(w_i, Q)$ is the position index of w_i in the query sentence *Q*.

The p(Q|Md)(probability of generating Q based on the language model Md), DF(Q)(document frequency measure) , and dist(d,Q)(distance measure) are three factors that are combined to score the mention of d in a query Q. The three criteria may be of various importance and must be weighted accordingly. Finally, the three criteria are combined by the product of the functions, and the integrated formula is:

$$score(Q,d) = \hat{p}(Q \mid M_d) \times DF(Q)^{\theta 1} \times \left\{\frac{1}{dist(d,Q)}\right\}^{\theta 2},$$

where document frequency and distance measure are weighted by the parameter $\theta 1$, $\theta 2$, and how to estimate each parameter is explained in the next subsection.

3.1.4 Parameter estimation

Since our approach is unsupervised, we need to set a loss function *f(θ)* from the retrieved results, and we estimate parameter *θ* through iterative scaling that minimizes the difference between loss functions $|f_{t-1}(\theta) - f_t(\theta)|$. We set *f(θ)* as the sum of frequencies of the words that are included in the relevant top-10 ranked documents but not included in the query. The intuition is that the smaller *f(θ)* becomes, the better performance we have. In other words, *f(θ)* is:

$$f(\theta) = \sum_{m}^{|U|}\sum_{i}^{K}\sum_{t \notin U_m} tf(t, doc(\theta, i, U_m)),$$

where $tf(t, doc(\theta, i, U_m))$ is term frequency of *t* in the document $doc(\theta, i, U_m)$ which is retrieved with the *i*th rank for the m[th] query U_m using parameter θ. In our experiments, K is 10 and $|U|$ is the number of input query sentences. We update the parameter using

$$\theta_{n+1,i} = \theta_{n,i} + \eta_{n,i}\{f_{n-1}(\theta) - f_n(\theta)\},$$

where $f_0(\theta) = \sum_{m}^{|U|}\sum_{i}^{K}\sum_{t \notin U_m} 1$, and $\{\theta_{0,1}, \theta_{0,2}\} = \{1, 1\}$.

The initial value $f_0(\theta)$ means that the frequency of the words that are not included in the query in each document is one, and we assign 1 for the initial parameter values.

For the update of parameter value θ_1 of the document frequency measure, we set $\eta_{n,1}$ to be proportional to $(avg\_df_n)/(avg\_df_{n-1})$. avg_df_n is the average document frequency of a word that is included in the top-10 ranked documents but are not included in the query at the n^{th} iteration. We perform an update of θ_1 using the following $\eta_{n,1}$:

$$\eta_{n,1} = \alpha \times \frac{avg\_df_n}{avg\_df_{n-1}},$$

$$avg\_df_n = \frac{\sum_{m}^{|U|}\sum_{i}^{K}\sum_{t \in d_i} df_n(t)}{\sum_{m}^{|U|}\sum_{i}^{K}\sum_{t \in d_i} 1}.$$

Similarly, for the update of parameter value θ_2 of the distance measure, we set $\eta_{n,2}$ to be proportional to $(avg\_dist_{n-1})/(avg\_dist_n)$. avg_dist_n is the average distance of words per document in n^{th} iteration. We perform an update of θ_2 using the following $\eta_{n,2}$:

$$\eta_{n,2} = \beta \times \frac{avg\_dist_{n-1}}{avg\_dist_n},$$

$$avg\_dist_n = \frac{\sum_{m}^{|U|}\sum_{i}^{K} dist_n(d_i, U_m)}{\left\{\sum_{m}^{|U|}\sum_{i}^{K} 1\right\}}.$$

The α and ß are constants that control the convergence speed of the iterations. In our experiments, we set α and β to 0.01, and the initial values of $\{\eta_{n,1}\ \eta_{n,2}\}$ to {0.01,0.01}. We stop iterating when $\left|f_{n-1}(\theta) - f_n(\theta)\right| \le 10$.

We obtain ranked relevant concepts according to the integrated IR model. Figure 1 shows one retrieval example using this model.

3.2 Step 2: Clustering retrieved concepts

We assume there is only one concept ID corresponding to a disease term. Since it is typical that more than one concept is retrieved for each disease term mentioned in a sentence, we cluster the concepts to group them according to the shared words.

We then apply the Hierarchical Agglomerative Clustering (HAC) algorithm which is the most commonly used method for document clustering [27]. It does not require a prespecified number of clusters. This algorithm begins with each document as a cluster of its own(lines 1-4 in Figure 2), and iterates by merging the two most similar clusters (lines 6-7), and terminates when there are no more non-overlapping sets to merge (lines 8-9). The HAC algorithm requires the definition of a similarity function between documents and between sets of documents. Each document (UMLS concept entry) is represented as an attribute vector, with each word in the input sentence being an attribute in this vector. If a word in the input sentence occurs in a concept entry, the corresponding attribute value of a vector is '1'. Otherwise, it is '0'.

The similarity of two documents is often taken as a normalized function of the dot product of their attribute vectors.

The HAC algorithm that we use is shown in Figure 2. This algorithm groups the most similar two clusters at each iteration, and recalculates the similarity between clusters. It terminates if the iteration is performed N (number of documents)-1 times or the maximum similarity between clusters becomes 0. We use the cosine similarity between vectors as a similarity measure. We assume that there is a hierarchical relation between concept entries 'A' and 'B' only if all the common words between 'A' and the input sentence occur in the entry 'B'. When A and B are represented as vectors, this property can be described as following: there is a hierarchical relation between concept entries

```
HAC(d_1, ...., d_N)
1   for n ← 1 to N
2     do for i ← 1 to N
3        do C[n][i] ← SIM(d_n, d_i)
4     I[n] ← 1  (keeps track of active clusters)
5   A ← [ ]  (assembles clustering as a sequence of merges)
6   for k ← 1 to N-1
7     do <i,m> ← argmax_{<i,m>:i≠m ∧ I[i]=1 ∧ I[m]=1} C[i][m]
8     if C[i][m] = 0
9       return A
10    A.APPEND(<i,m>)  (store merge)
11    I[m] ← 0  (deactivate cluster)
12  return A
```

Figure 2. Modified HAC algorithm for our task

$$\text{if } ((A \cdot B) = \{\min(\|A\|, \|B\|)\}^2)$$
$$\text{SIM}(A,B) = \text{cosine\_similarity}(A,B);$$
$$\text{otherwise, SIM}(A,B) = 0;$$

Figure 3. Similarity between concept entries

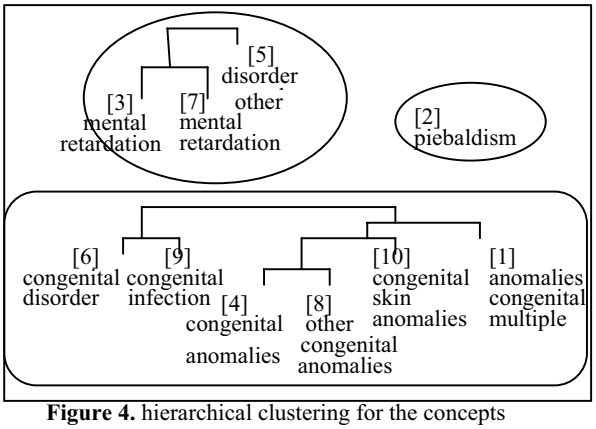

Figure 4. hierarchical clustering for the concepts in Fig.1

'A' and 'B' only if $(A \cdot B)$ equals to $\{\min(\|A\|, \|B\|)\}^2$). Finally, we assign the similarity value as shown in Figure 3.

The final result of clustering is a set of tree(s), and we make one cluster for each tree. A clustering example using the top 10 ranked list is shown in Figure 4.

3.3 Step 3: Choosing answer concepts from clusters

Once the clustering has been completed, we select a concept that shows the highest rank in each cluster and is within the threshold. We select the threshold dynamically based on the ranking score distribution, specifically choosing the point at which there is a significant drop in ranking scores which means the ratio of score[i]/score[i+1] is biggest.

From the three clusters in Figure 5, concept IDs C0000772, C0080024, and C0025362 are chosen because they show highest rank in each cluster. Then, you can see that the ratio of score[4]/score[5] is biggest from Figure 1. We remove the concepts with the 5th to 10th rank from the answer list. The final

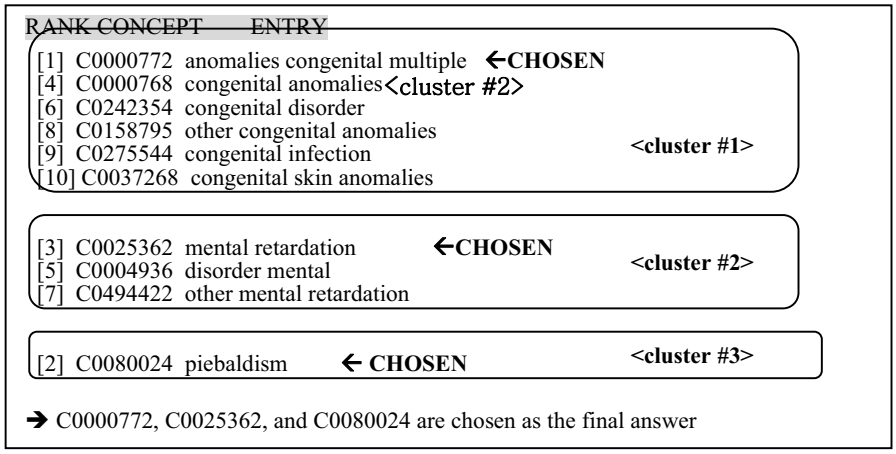

Figure 5. The final concepts that are retrieved as answers

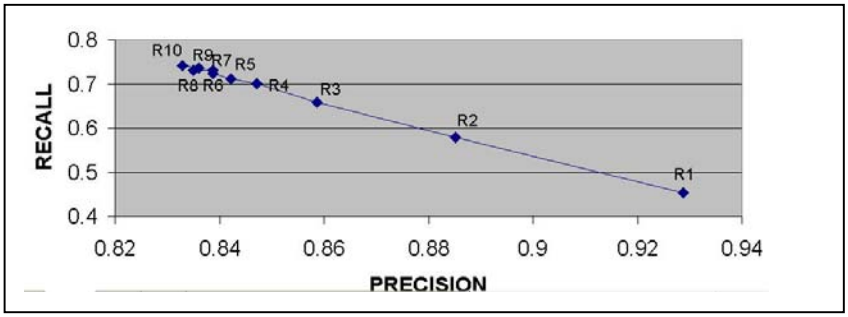

Figure 6. Precision/recall when top-R ranked concept entries are retrieved in Step 1

Table 1 Performances of our system and other previous systems

	Precision (%)	Recall (%)	f-measure (%)
Our performance	**73.28**	**77.51**	**75.34**
Performance of EBI's statistical method [17]	66.17	67.10	66.63
Performance of EBI's Dictionary lookup method [17]	79.40	60.06	68.39
MetaMap	83.90	53.57	65.39

selection of mapped disease concepts is C0000772, C0080024, and C0025362.

4. Experimental Evaluation

4.1 Performance of our three-step method

We construct a disease lexicon by extracting disease-related concepts from UMLS according to J. Antonio et al. [17], and use their evaluation data. The data consists of 600 sentences and 924 disease terms.

In the experiments, we obtained the following results.

1. Our proposed method achieved an F-measure of 75.34%. (see Table 1)
2. Our method significantly outperformed the statistical method and dictionary-based method of EBI (European Bioinformatics Institute) by 6.95 to 8.71%, and MetaMap by 9.95 %. (see Table 1)
3. When we did not perform the clustering of Step 2, the precision dropped significantly. (see Table 2)
4. When we did not use distance or document frequency measures of Step 1, the precision was better, but the recall was reduced. (see Table 2)
5. We can see the convergence of the precision/recall points for higher ranks (see Figure 6)

As shown in Table 1, among the systems using the same evaluation data, our system outperformed all previous reported systems with a precision of 73.28%, recall of 77.51%, and F-measure of 75.34%. In the Experiments, the EBI's experiment and MetaMap program trained/tested the same set of data.

Figure 6 shows each precision/recall of our system when we retrieve top R ranked concept entries in Step 1, with R=1 to 10. We can note the convergence of the precision/recall points for higher ranks, and we see that R=10 is enough for the convergence of precision/recall. (Figure 6).

The Precision, recall, and F-measure represent proportions of populations. In trying to determine the difference in performance of two systems, we therefore employ the z-test on two proportions. We test the significance of the differences in F-measures between 3 kinds of system pairs: {our system, EBI's statistical system}, {our system, EBI's dictionary-based system}, and {our system, MetaMap}. Given two system outputs, the null hypothesis is that

there is no difference between the two proportions, i.e., H0: p1 = p2. The alternative hypothesis states that there is a difference between the two proportions, i.e., p1 =/ p2. A z-statistic of ±2.58 means that the difference between the two proportions is significant at a = 0.01. Z-values in all three significance tests are bigger than 2.58, and shows that the null hypothesis of no difference in the two proportions is rejected.

4.2 Significance of each step in our method

We now summarize the significance of each step introduced in Section 3. As shown in Table 2, each step has a significant impact on the system's performance.

Without using the document frequency and distance measures for the model in Step 1, the recall is significantly reduced, since the retrieved concept entries can include words that have a high information content but are not related to the query.

Without the clustering of Step 2, the precision is significantly reduced, since more than one concept is chosen for each disease-related term in the input sentence. The performance without clustering shows reduced performance, compared to the method with clustering.

In a similar way, without the cut-off threshold restriction in Step 3, the recall improves, which we attribute to the selection of more concepts. However, the precision is reduced.

One might consider the addition of noun phrase detection, and then use each noun phrase as an input, rather than the whole sentence. However, the experiment after noun phrase detection shows reduced performance compared with the method using a whole sentence as an input. That indicates that, in many cases, a disease term is not embedded in one noun phrase.

In conclusion, the experiments demonstrate that the second and third steps contribute to the improvement of precision, and the first step to the improvement of recall. We conclude that all three steps are important for mapping of sentences into an ontology.

In the experiments, we see that word variation results in some errors. e.g., 'hemolysis' vs. 'haemolysis', 'norrie' vs. 'norrie's' , and 'cirrhotic' vs. 'cirrhosis'. In future work, such variants can be treated by aligning names under the same concept in UMLS.

5. Conclusion

To improve the mapping performance of biomedical sentences into an ontology, we propose applying an integrated information retrieval technique which combines a simple language model, document frequency, and distance measure, followed by a clustering of answer candidates. We regard a biomedical sentence as a query and a UMLS concept entry as a document. The proposed method does not require any preprocessing except part-of-speech tagging, and we do not perform stemming, noun phrase detection, or normalization of words. A whole sentence is used as an input without using any n-gram window.

In our proposed three-step method, the first step adopts an integrated information retrieval model for our mapping problem, and retrieves the top-10 answer candidates using the model. In the second step, we cluster the top-10 ranked concepts according to the disease-related words in common. In the final step, we choose one concept from each cluster based on a cut-off threshold. The experimental results show that our three-step method performs significantly better than previous methods by 6.95 to 9.95 percent.

Table 2 Change in performance when one phase is removed

		Precision (%)	Recall (%)	f-measure (%)
Using all steps		**73.28**	**77.51**	**75.34**
Change in 1st step	Without distance measure (IR formula + DF)	77.63	71.00	74.17
	Without document frequency +distance (only IR formula)	87.92	48.05	62.14
Change in 2nd step	Without Clustering	55.87	78.81	65.39
Change in 3rd step	Without cutting threshold	66.30	79.12	72.14
When we input noun phrases, not a whole sentence		71.94	76.03	73.93

Even though the characteristics of a biomedical sentence and a UMLS concept entry are different from those of a query and a document in traditional document retrieval, we show that the modified information retrieval model is appropriate with reasonable performance.

In future work, we will expand the volume of the evaluation data, and attempt to apply this model for the mapping of other types of biomedical terms such as genes or drugs. In addition, term variations such as the missing of words or substitution of synonyms will be also be considered.

Acknowledgements

This research was supported by the Alberta Ingenuity Centre for Machine Learning (AICML).

6. References

[1] M. Krallinger, and A. Valencia, "Text-Mining and Information-Retrieval Services for Molecular Biology", *Genome Biology.* 6:224. 2005

[2] C. Blaschke, M.A. Andrade, C. Ouzounis, A. Valencia, "Automatic extraction of biological information from scientific text: protein-protein interactions", *proc. Int. Conf. Intell. Syst. Mol. Biol*. 30A (2) (1999) 60-67

[3] T. Ono, H. Hishigaki, A. Tanigami, T. Takagi, "Automated extraction of information on protein-protein interactions from the biological literature", *Bioinformatics* 17 (2) (2001) 155-161

[4] J. Thomas, D. Milward, C. Ouzounis, S. Pulman, M. Carroll, "Automatic extraction of protein interactions from scientific abstracts", *Pac. Symp. Biocomput*. (2000) 541-552

[5] D.R Swanson, Fish Oil, Raynaud's Syndrome, and Undiscovered Public Knowledge. *Perspectives in. Biology and Medicine* 30(1):7-18, 1986

[6] D.R Swanson, “Medical literature as a potential source of new knowledge”, *BULLETIN OF THE MEDICAL LIBRARY ASSOCIATION* 78 (1): 29-37, 1990

[7] Weeber, M., Klein, H., Aronson, A.R., Mork, J.G., de Jong-van den Berg,. L.T.W., & Vos, R, text-Based Discovery in Biomedicine: The Architecture of the DAD-system, *Proc AMIA Symp*. 35 (20) 903-7, 2000

[8] O. Bodenreider, “The Unified Medical Language System (UMLS): integrating biomedical terminology”. *Nucleic Acids Res 32 Database issue*:D267-70. 2004.

[9] D.A. Lindberg, B.L. Humphreys, A.T. McCray. “The Unified Medical Language System”. *Methods Inf Med 32*(4), pp.281-291, 1993

[10] A.R Aronson: Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. *Proc AMIA Symp*, pp.17-21, 2001

[11] W. Hersh, T.J. Leone, The SAPHIRE server: a new algorithm and implementation. *Proc Annu Symp Comput Appl Med Care,* pp.858-862*,* 1995.

[12] A.J. Butte, I. S. Kohane, Creation and implications of a phenomegenome network. *Nat Biotechnol,* 24(1):pp.55-62, 2006

[13] A.J. Butte, R. Chen, Finding disease-related genomic experiments within an international repository: first steps in translational bioinformatics. *AMIA Annu Symp Proc,* pp.106-110*,* 2006

[14] N.H. Shah, D.L. Rubin, I. Espinosa, K. Montgomery, M.A. Musen, Annotation and query of tissue microarray data using the NCI Thesaurus. *BMC Bioinformatics,* 8:296, 2007

[15] M. Dai, “An Efficient Solution for Mapping Free Text to Ontology Terms”. *AMIA Summit on Translational Bioinformatics. San Francisco, CA,* 2008

[16] S. Gaudan , A. Jimeno Yepes , V. Lee , D. Rebholz-Schuhmann, “Combining evidence, specificity, and proximity towards the normalization of gene ontology terms in text”, *EURASIP Journal on Bioinformatics and Systems Biology*, v.8 n.1, pp.1-9, January 2008

[17] A. Jimeno, E. Jimenez-Ruiz, V. Lee, S. Gaudan, R. Berlanga, and D. Rebholz-Schuhmann, “Assessment of disease named entity recognition on a corpus of annotated sentences”, *BMC bioinformatics*, 9 Suppl 3():S3, 2008

[18] F. Mougin, A. Burgun, and O. Bodenreider, “Mapping data elements to terminological resources for integrating biomedical data sources”, BMC Bioinformatics 7(S-3), 2006

[19] A. Mottaz, Y. L. Yip, P. Ruch, and A. Veuthey*, “*Mapping protein information to disease terminologies”, *Journal of Integrative Bioinformatics*, 4(3):79, 2007

[20] J. Hakenberg, C. Plake, L. Royer, and H. Strobelt, U. Leser, and M. Schroeder, “Gene mention normalization and interaction extraction with context models and sentence motifs”, *Genome Biol*, 9 Suppl 2: S14, 2008

[21] K.B. Cohen, G.K. Acquaah-Mensah, A.E. Dolbey, and L. Hunter, “Contrast and variability in gene names” *ACL-02 workshop on Natural language processing in the biomedical domain*, pp.14-20, 2002

[22] Y. Tsuruoka, J. Mcnaught, and S. Ananiadou, “Normalizing biomedical terms by minimizing ambiguity and variability” *BMC Bioinformatics*, Vol. 9, No. Suppl 3. 2008

[23] H.W. Chun, Y. Tsuruoka, J.D. Kim, R. Shiba, N. Nagata, T. Hishiki, and J. Tsujii, “Extraction of gene-disease relations from Medline using domain dictionaries and machine learning” *Pac Symp Biocomput*, pp. 4-15, 2006

[24] A. Névéol, W. Kim, W. John Wilbur, and Zhiyong Lu, “Exploring Two Biomedical Text Genres for Disease Recognition”, *Proc. of the Workshop on BioNLP*, pp.144–152, 2009

[25] L.K. Tanabe and W. J. Wilbur, “A Priority Model for Named Entities”. *Proc. of HLT- NAACL BioNLP Workshop*, pp.33-40, 2006

[26] Jay M. Ponte and W. Bruce Croft*, “*A Language Modeling Approach to Information Retrieval”, *Proc. of ACM SIGIR conference on Research and development in information retrieval*, pp.206-214, 1998

[27] P. Willet, “Recent trends in hierarchical document clustering: a critical review”. *Information Processing and Management*, Vol.24, pp.577-597, 1988.

[28] Bundschus M, Dejori M, Stetter M, Tresp V, Kriegel HP, “Extraction of semantic biomedical relations from text using conditional random fields”, BMC Bioinformatics, Apr 23;9:207, 2008