IIITH at BioASQ Challenge 2015 Task 3a: Extreme Classification of PubMed Articles using MeSH Labels

Avinash Kamineni*[1], Nausheen Fatma*[1], Arpita Das*[1], Manish Shrivastava[1], and Manoj Chinnakotla[2]

[1] International Institute of Information Technology Hyderabad, India
[2] Microsoft, India
{avinash.kamineni,nausheen.fatma,arpita.das}@research.iiit.ac.in
m.shrivastava@iiit.ac.in,manojc@microsoft.com

Abstract. Automating the process of indexing journal abstracts has been a topic of research for several years. Biomedical Semantic Indexing aims to assign correct MeSH terms to the PubMed documents. In this paper we report our participation in the Task 3a of BioASQ challenge 2015. The participating teams were provided with PubMed articles and asked to return relevant MeSH terms. We tried three different approaches: Nearest Neighbours, IDF-Ratio based indexing and multi-label classification. The official challenge results demonstrate that we consistently performed better than the baseline approaches for Task 3a.

Keywords: MeSH Indexing; Biomedical Semantic Indexing; Hierarchical Text Classification; FastXML; PubMed; Information Retrieval and Extraction; Metamap

1 Introduction

The annotation of biomedical journals by the experts is both expensive and time-consuming. Therefore, Large Scale Hierarchical Text Classification in this domain has gained much importance over the past few years. It is also helpful in fields like Question Answering, Information Retrieval, Categorization etc. The challenge introduced by BioASQ [23] deals with handling large scale complex data and automatically assigning relevant MeSH [1] terms to the PubMed [3] articles.

Researchers have tried to crack the problem of biomedical semantic indexing using a wide variety of methods such as Latent Semantic Analysis [14], Latent Dirichlet Allocation (LDA) [7], Support Vector Machines [9] etc. We approach the problem from a document clustering perspective, based on the observation that similar documents often share MeSH terms. In this paper, we built a generic model for tagging the documents with MeSH terms which can be utilized in

* These authors contributed equally

any other domain. Three different approaches namely Nearest Neighbours, IDF-Ratio based learning and *FastXML* [21] based extreme classification were used. All the three approaches beat the BioASQ baseline and had high precision values, however the values of recall were comparatively low.

The rest of the paper is divided into following sections: *Section 2*, describe the previous works done in BioASQ semantic indexing task. *Sections 3* explains the model using different approaches in detail. *Section 4*, contain the experiments performed and the results obtained. *Section 5*, comprises of the conclusion and future work.

2 Related Work

Semantic Indexing has been a topic of research for several years. Amongst the successful unsupervised models, the most well known one is Latent Semantic Analysis (LSA) [14] developed by Deerwester et al. LSA takes the high dimensional vector space representation of documents and applies dimension reduction by Singular Value Decomposition (SVD) on it. The similarities between documents are more reliably estimated in the latent semantic space than in the original one. However, LSA lacks solid statistical foundation. Hence, Hoffman et al. introduced Probabilistic Latent Semantic Analysis (PLSA) [15] based on a statistical latent class model.This model dealt with domain specific synonymy and polysemy. David M. Blei et al. introduced Latent Dirichlet Allocation (LDA) [7] considering the mixture models that capture the exchangeability (Exchangeability and related topics David J. Aldous) of both words and documents. Each item of a collection is modeled as a finite mixture over an underlying set of topics.

Few supervised methods were also developed in this area. Bing Bai et al. proposed Supervised Semantic Indexing (SSI) [6] which defines a class of models that can be trained on a supervised signal (i.e., labeled data) to provide a ranking of a database of documents given a query. Sutanu et al. proposed sprinkling [11] to automatically index documents. Sprinkling is a simple extension of LSI based on augmenting the set of features using additional terms that encode class knowledge. But sprinkling treats all classes in the same way. To overcome this problem, they proposed Adaptive Sprinkling (AS) which leverages confusion matrices to emphasise the differences between those classes which are hard to separate.

Considering prediction of MeSH headings, we have Medical Text Indexer (MTI) [20], the official solution of National Library of Medicine (NLM). The major components of MTI are:

1. MetaMap Indexing (MMI) [4]
2. PubMed Related Citations [17]
3. Restrict to MeSH [8]
4. Extract MeSH Descriptors
5. Clustering and Ranking [2]

The approach of Tsoumakas, G. et al. [24] performed better than MTI. MetaLabeler [22] by Tang et al. used binary classification model trained using linear

SVM. Also a regression model was trained to predict the number of MeSH headings for each citation. Finally, given a target citation, different MeSH headings were ranked according to the SVM prediction score of each classifier, and the top K MeSH headings were returned. Learning to rank (LTR) method, which was utilized by Lu et al. [19] [16] for automatic MeSH annotation. In this method, each citation was deemed as a query and each MeSH headings as a document. LTR method was utilized to rank candidate MeSH headings with respect to target citation. The candidate MeSH headings came from similar citations (nearest neighbors). In the similar line of thought Huang et al. reformulated the indexing task as a ranking problem [16]. They retrieved 20 neighbor documents, obtained a list of MeSH main headings from neighbors, and ranked the MeSH headings using ListNet learning-to-rank algorithm [10].

3 Our Approach

Our system mainly consists of three different modules. We compare these different systems. In this section, we explain these approaches in detail.

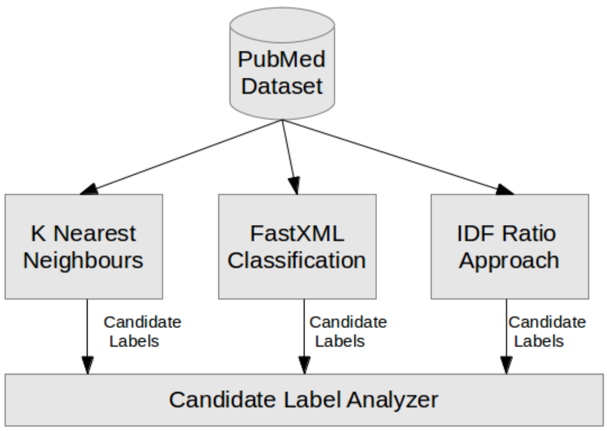

Fig. 1: System Modules

We have implemented three distinct techniques to index articles. Eventually, our aim was to find which of these techniques contribute the most in finding relevant MeSH terms. The following are the three techniques:

1. K Nearest Neighbours approach
2. IDF-Ratio based approach
3. Extreme Classification using FastXML.

3.1 K Nearest Neighbours Approach

In this approach,we use a K Nearest Neighbours [12] based lazy learning approach to find the most relevant MeSH headings.

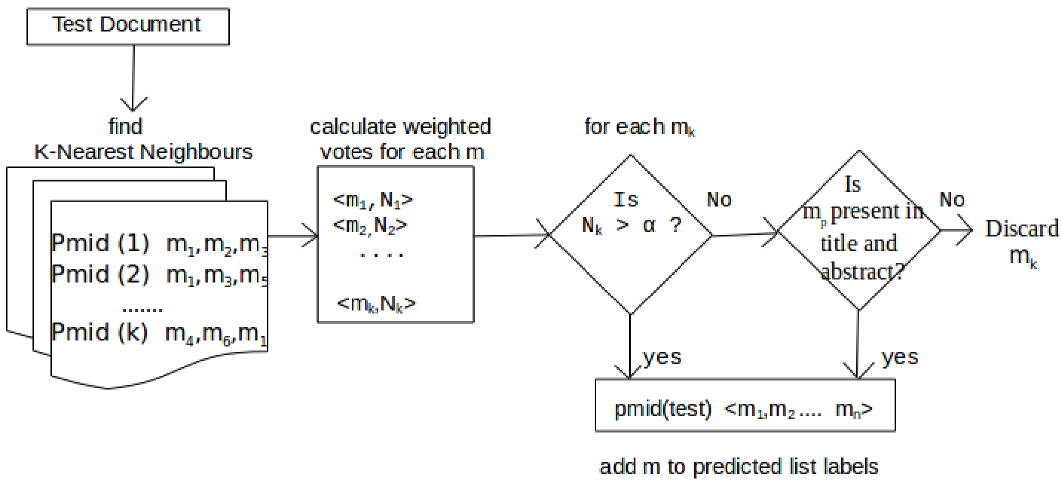

Fig. 2: K Nearest Neighbours approach

The method is as follows :

1. The training files were first converted to Lucene index with fields "pmid","title","abstractText","meshMajors".

2. K Nearest Neighbours are retrieved for finding the candidate MeSH terms

   For a given unknown test instance, the fields abstract and title were concatenated as a single string. We then find K Nearest Neighbours (with k=60) from the Lucene index. Similarity of documents is computed by finding the number of overlapping words and giving them different weights based on TF-IDF [18].

3. Rank to each candidate MeSH term is given by its number of occurrences in the neighbours

   Top 60 (k=60) similar records were retrieved and a HashMap was created with every MeSH term found in the neighbours as key and the count of total number of times that MeSH term occurs in the all the neighbours together as value. The HashMap keys become our candidate MeSH terms for the given test instance.

4. Threshold is used for final predictions

   For every <key,value> pair in the hashmap created above, the *value* is compared against a threshold α. If *value* $>= \alpha$ then the key is included in a set S. If the *value* $< \alpha$ then we check if the key (which is a MeSH term) exists in the title or abstract. If the key is present in the title or abstract then it is very likely that the key is a relevant label and is added to the set S. After all the <key,value> pairs have been iterated,the set S becomes our final MeSH label set for x.
   α was set to 12 empirically for k=60. It was observed that threshold $\alpha = k/5$ generally gave optimum results for unweighted votes.

Query	**k**	**alpha**	**precision**	**recall**
Title + abstract -stopwords	60	12	0.510845	0.503196
Title + abstract -stopwords	75	3.75	0.472817	0.539864
nounphrases(From Title + abstract)	75	3.75	0.451753	0.540818
Nouns(From Title + abstract)	75	3.85	0.464746	0.541609
Nouns(From Title + abstract)	75	15	0.511757	0.487618
Nouns(From Title + abstract)	60	12	0.50631	0.496969

Table 1: Results of different approaches for Nearest Neighbours Candidate Selection

Some variations using this approach were also tried :

1. Weighted votes are used with similarity distance score as weight.
2. Using just noun phrases as queries
3. Using just nouns as queries

3.2 IDF-Ratio based approach

We know that IDF (Inverse Document Frequency) measures the importance of a particular term in a set of documents. But certain terms like "is", "and", and "are", may appear frequently but have little importance. Hence idf weighs down the frequently occurring terms and boosts up the rare and significant ones. IDF for a term t can be expressed as:

$$IDF(t) = \log \frac{N}{N_t} \tag{1}$$

where, N is total number of documents, N_t is number of documents with term t.

Here for the task of semantic indexing we need to find that how much a particular word is important for a MeSH term. In other words we want to find out which particular word(s) in a document can lead to a MeSH term.For extracting this information the novel concept of IDF-Ratio is introduced. This ratio identifies the word(s) in a document that will certainly result in a MeSH term. The IDF Ratio with respect to a MeSH term for a word can be expressed as :

$$IDF - Ratio(t|m) = (\frac{\frac{N_m}{N_{tm}}}{\frac{N}{N_t}}) \tag{2}$$

where, N_m is number of times a particular MeSH term m is occurring, N_{tm} is total number of times the term t occurred with that MeSH term m. Thus, IDF-Ratio(t|m) for a t term exists for every 27455 MeSH terms (m) provided.

We have IDF Ratio of a word for all the MeSH terms. It does not make sense to consider all the 27455 MeSH terms for a single word, since a word cannot lead to all the MeSH terms. So it is necessary to filter out the unwanted MeSH terms for each word. We do this by thresholding. After experimenting with different values, a threshold of 0.55 was found to be optimum. Now every word is related to 5-15 relevant MeSH terms which it can potentially lead to. Some of the MeSH terms like “humans”, “male”, “female”, “animals” are very common and occurs with almost every word, so for any word, the IDF Ratios with respect to these MeSH terms are very high. So almost all the words lead to these MeSH terms.

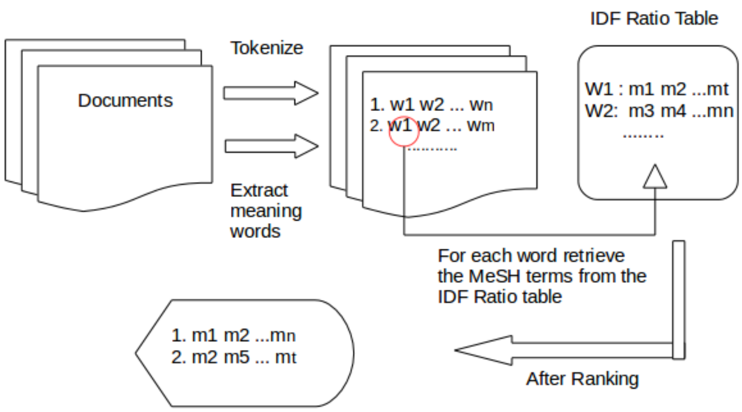

Fig. 3: IDF Ratio based indexing

Algorithm

1. `Pre-processing`
   The documents given to index are tokenized. The set of biomedical stopwords are eliminated from the documents. Some Special symbols are removed. The symbols necessary for retaining the meaning of chemical components are kept intact.
2. `Extraction of meaning words`
   POS-Tagger is used to extract the NN ,NNS, NNP,VB,JJ and RB tags from the documents. SENNA [13] is used for the tagging purpose. It uses deep learning (unsupervised convolutional neural network) to tag sentences.
3. `Collection of candidate MeSH terms`
   After obtaining the meaning words we consult the IDF Ratios with respect to the MeSH terms. For each word, we choose a set of MeSH terms it can lead to. Finally we get a candidate set of potential MeSH terms.
4. `Ranking the candidate MeSH terms`
   The MeSH terms in the candidate set has to be ranked correctly. The following ranking approaches were used:
   (a) **Ranking in the order of IDF-Ratio:** The words possess IDF Ratio with respect to the MeSH terms, we can rank these MeSH terms in the order of these ratios. If more than one word in the document leads to the same MeSH term ,their corresponding IDF ratios are simply added.
   (b) **Ranking in terms of maximum intersection:** In a document if several words are pointing to the same MeSH term then that MeSH term must be important for that document. This concept is utilised in this ranking method. We gather the set of MeSH terms for each meaning word and find the intersection of these sets. The elements of intersection are assigned as indices of the document.
   (c) **SVM-Rank:**[3] It is used to rank lists of items. For training, the inputs to SVM-Rank are ordered entries of every possible pair of items which are assigned weights depending upon the correctness of the order. Initial step of optimisation problem is formulated as ordinal regression; however, it is turned into a classification problem due to the pair wise difference. In the semantic indexing task, feature vector is composed for the MeSH terms. The feature vector consists of bag of words, IDF Ratio weights, etc. The above two methods of ranking mentioned in a) and b) did not yield good results, so the rankings obtained through them were included as features for training SVM-Rank. Inclusion of this feature resulted in a slight improvement in the performance.
   The main difficulty was in assigning weights to the MeSH terms. While training, we give all the terms assigned to that document very high weights, but we cannot grade them in some order, as we have no clue which of the tags assigned to the document has more weight and which has less weight. Similarly, we have no other way of giving weights to the

[3] SVM for ranking `http://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html#References`

remaining MeSH terms in the data provided, that are not assigned to that document .

After ranking is done ,the filtered top-ranked MeSH terms are assigned to the document.

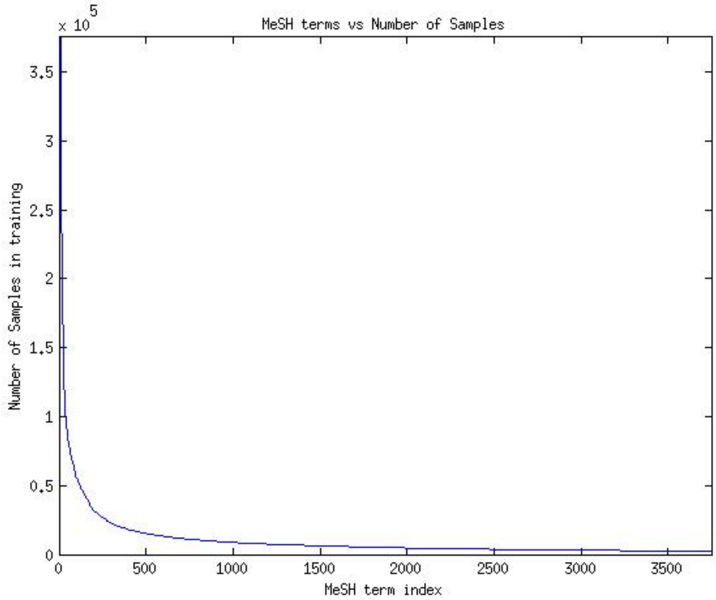

Fig. 4: MeSH Term vs Number of Samples in Task 3a Training Data

3.3 Extreme Classification using FastXML

The main objective of FastXML [21] is to acquire fast and efficient training of a model. Training of 4 Million BioASQ 2015 documents took about 36 hours on a 4 core machine. Also, FastXML is capable of learning the hierarchy of the MeSH terms by optimizing the ranking loss function. Existing approaches optimize local measures of performance which depends solely on predictions made by the current node being partitioned. FastXML allows the hierarchy to be learned node by node, starting from the root and going down to the leaves, thus it is more efficient than learning all the nodes jointly. The frequent MeSH terms could be learnt better compared to the rare ones.

FastXML is based on the assumption that only a few number of labels occur at each region of the feature space. It learns ensemble of trees and does not rely on base classifiers. The output of the classifier is the labels along with their probabilities. It also provides the precision at 1..k, where k is the max number of labels that must be tagged for a document. The experimental results of this approach is explained below.

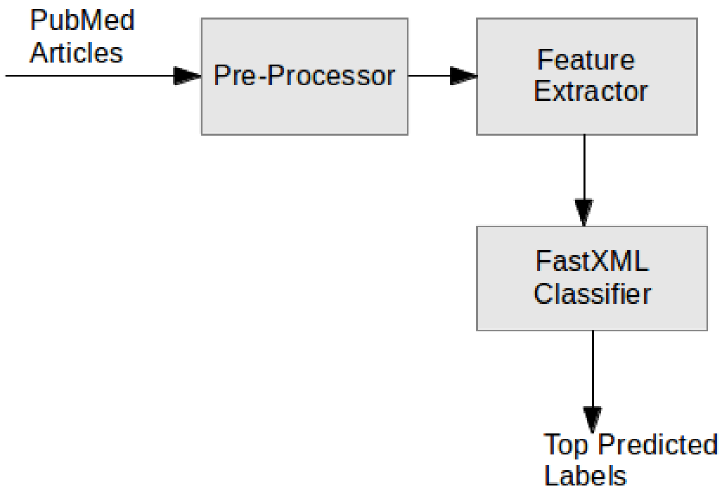

Fig. 5: FastXML approach

1. **Tokenization**
   As the terms in this particular domain contains special symbols in the chemical formulae etc, special care is taken while tokenizing. Few special symbols like (-,) are maintained. This tokenization is done using the tokenization module of word2vec[4] source code provided in Open source software by BioASQ. They also have the vocabulary list of 1.7 million words.[5]
2. **DF Matrix Construction**
   We iterate over each document in the BioASQ 2015 training set and tokenize the title and abstract, for each token we increment the corresponding MeSH term column. So, this gives us a sparse matrix, indexed accordingly, which is later used for feature extraction.

[4] The word vectors can then be used, for example, to estimate the relatedness of two words or to perform query expansion. `http://bioasq.lip6.fr/tools/BioASQword2vec/`

[5] For the unidentified words in the vocabulary, we have done simple Laplace Smoothing for updating the weights of the feature.

3. **Feature Extraction**
   Different features used for the classification.They are described as follows:
   (a) **Unigrams with TF-IDF weights**
   From a document, each token in the title and abstract is taken and their term frequency is found out. Document frequency can be found from the DF Matrix. Hence, we can calculate TF-IDF value for each token. These TF-IDF weights act as a feature.
   (b) **Exact Match MeSH heading feature**
   We create bag of words of all the the 27K MeSH terms. If the document text contains a MeSH heading, then its position in the 27K dimension feature vector is set to 1 , else 0.
   (c) **Noun Phrase Feature**
   We need few specific tools for dealing with the Biomedical domain. Using metamap [5], we can find the part of speech tags for the sentences, identify the chunks from the given article, identify the head or main word from the passage, using which we could find the noun phrases from the article and pass the tokens present in noun phrase as features.
   (d) **Semantic Feature**
   Using nouns extracted from the noun phrases, we try to get the meaning words for the identified concepts from the MetaMap, MeSH Heading Descriptors etc. LESK Algorithm[6] is used to obtain semantically similar words.

MeSH Term/ Vocabulary	Index in Vocabulary	Overall DF	DF in MH1(Human)	DF in MH2	DF in MH3	...	DF in MH26456
aminopepsin	283542	3	3	2	2	..	0
cardio	428700	57	45	0	34	..	0

Table 2: Index of the Document Frequency Matrix

4 Experiments and Results

As a part of the BioASQ 3a challenge 2015, we have made weekly submissions of the two of three batches. We performed better than the baseline System each time. The results of one of submission of 3a Batch 3, Week 3 are shown in the following tables.

In tables 3 and 4, IIIT System 3 represents the Nearest Neighbours approach, IIIT System 4 represents the IDF Ratio based approach and qaiiit system 1 represents the FastXML approach.

[6] Semantic Word Matching Algorithm `http://en.wikipedia.org/wiki/Lesk_algorithm`

Submission /Metric	MIF	MIP	MIR	Acc.
IIIT System 3	0.4978	0.4412	0.5711	0.3323
IIIT System 4	0.4534	0.5737	0.3748	0.2198
qaiiit system 1	0.4164	0.4047	0.4287	0.2672
BioASQ Baseline	0.262	0.2391	0.2897	0.1542

Table 3: Flat Measures of Task 3a Batch 3,Week 3

Submission / Metric	HIP	HIR	HIF	LCAF
IIIT System 3	0.6408	0.7074	0.648	0.4362
qaiiit system 1	0.6191	0.5477	0.5555	0.3767
IIIT System 4	0.7591	0.4304	0.5111	0.3636
BioASQ Baseline	0.5337	0.5831	0.5321	0.3054

Table 4: Hierarchial Measures of Task 3a Batch 3,Week 3

The results of IDF Ratio method are as follows:

1. This method gives a very high precision of 0.84 but the candidate set is too large in number.
2. SVM-Rank gives a very low recall of 0.25 only. This is due to the inability to assign proper weights in descending order to the MeSH terms.
3. Ranking in the order of IDF-Ratio gave a recall of 0.267. Very common MeSH terms like male, females, rats had very high IDF-Ratio value in the overall documents, hence they were assigned to almost all the documents,thus decreasing the recall value.
4. Ranking in terms of maximum intersection also gave a recall of 0.232. This faced the similar problem as that in ranking in the order of IDF-Ratio. Mostly, the common MeSH terms were found in the intersection set .
5. Due to the high precision and low recall the overall F-score reduced to 0.4.

	Precision	Recall	F-Score
SVM-Rank	0.84	0.25	0.39
IDFRatio order	0.84	0.267	0.41
Intersection	0.84	0.232	0.36

Table 5: Showing results for different ranking methods using IDF Ratio

Error Analysis

1. Few of the common MeSH terms like "Humans", "Male","Female" occurs in most of the articles hence these terms are tagged with high probability.
2. Rare MeSH terms like "2-Oxoisovalerate Dehydrogenase (Acylating)", "Hydroxyacyl-CoA Dehydrogenase" occurs in very few articles,hence their probability of being tagged is very low.

Observations

For IDF Ratio based approach, the following observations were made:

1. The concept of IDF Ratio is pretty intuitive, it help us determine the importance of a word for a particular MeSH term. We can determine the presence of which words lead to a MeSH term.
2. As a part of an experiment, hierarchy information was tried to be infused in this method. Several approaches were tried like for a MeSH term, its child, parent and siblings are included till 2 levels in the candidate set, or if a parent is included in candidate set its child is excluded,etc. Several such schemes were applied but with no significant change in results. No particular hierarchial pattern was followed by the data provided.
3. As already mentioned the precision of this approach was high, the candidate set sort of formed a superset of the answers obtained by the other two methods i.e., Extreme Classification and Nearest Neighbour.

5 Conclusion and Future Work

It can be stated that by using the Nearest Neighbours we can limit the candidate MeSH terms by maintaining the precision and recall. By the IDF Ratio approach we can gather all the mesh terms a word can lead to. It sort of captures both lexical and semantic information. By using Extreme Classification, training can be done quickly even on a single machine, this process is scalable. The information of hierarchy between the MeSH terms can be captured. These three approaches mentioned, are implemented independently. The next logical step would be to combine these results and use them as features for the ranking algorithm, which will be done as a part of our future work. Future work includes :

1. To come up with a better ranking algorithm to rank the MeSH terms in the candidate set.
2. To exploit the hierarchy information of the MeSH headings provided.
3. To merge the 3 approaches to get a compact and smaller version of the candidate set.
4. In IDF-Ratio approach we are basically finding the MeSH terms which are pointed by individual words, in future it would be a better idea to find the MeSH terms which the entire document is leading to.

References

1. Medical subject headings https://www.nlm.nih.gov/mesh/.
2. Medical text indexer (mti) processing flow whitepaper.
3. Search engine of medline database http://www.ncbi.nlm.nih.gov/pubmed.
4. Aronson AR. The mmi ranking function whitepaper, (1997).
5. A. R. Aronson. Effective mapping of biomedical text to the umls metathesaurus: the metamap program. *Proc AMIA Symp*, pages 17–21, 2001.

6. Bing Bai, Jason Weston, David Grangier, Ronan Collobert, Kunihiko Sadamasa, Yanjun Qi, Olivier Chapelle, and Kilian Weinberger. Supervised semantic indexing. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 187–196, New York, NY, USA, 2009. ACM.
7. David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation, March 2003.
8. Olivier Bodenreider, Stuart J Nelson, William T Hole, and H Florence Chang. Beyond synonymy: exploiting the umls semantics in mapping vocabularies. In *Proceedings of the AMIA symposium*, page 815. American Medical Informatics Association, 1998.
9. Lijuan Cai and Thomas Hofmann. Hierarchical document categorization with support vector machines. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 78–87. ACM, 2004.
10. Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136. ACM, 2007.
11. Sutanu Chakraborti, Rahman Mukras, Robert Lothian, Nirmalie Wiratunga, Stuart N. K. Watt, and David J. Harper. Supervised latent semantic indexing using adaptive sprinkling. In Manuela M. Veloso, editor, *IJCAI*, pages 1582–1587, 2007.
12. Tsung-Hsien Chiang, Hung-Yi Lo, and Shou-De Lin. A ranking-based knn approach for multi-label classification. In *ACML*, pages 81–96, 2012.
13. Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537, November 2011.
14. Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis, 1990.
15. Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 50–57, New York, NY, USA, 1999. ACM.
16. Minlie Huang, Aurélie Névéol, and Zhiyong Lu. Recommending mesh terms for annotating biomedical articles, 2011.
17. Jimmy Lin and W John Wilbur. Pubmed related articles: a probabilistic topic-based model for content similarity, 2007.
18. Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval.* Cambridge University Press, New York, NY, USA, 2008.
19. Yuqing Mao and Zhiyong Lu. Ncbi at the 2013 bioasq challenge task: Learning to rank for automatic mesh indexing, 2013.
20. James G Mork, Antonio Jimeno-Yepes, and Alan R Aronson. The nlm medical text indexer system for indexing biomedical literature. In *BioASQ@ CLEF*, 2013.
21. Yashoteja Prabhu and Manik Varma. Fastxml: A fast, accurate and stable tree-classifier for extreme multi-label learning. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 263–272, New York, NY, USA, 2014. ACM.
22. Lei Tang, Suju Rajan, and Vijay K Narayanan. Large scale multi-label classification via metalabeler. In *Proceedings of the 18th international conference on World wide web*, pages 211–220. ACM, 2009.
23. George Tsatsaronis, Michael Schroeder, Technische Universitt Dresden, Georgios Paliouras, Yannis Almirantis, Eric Gaussier, Patrick Gallinari, Thierry Artieres, Michael R. Alvers, Matthias Zschunke, Transinsight Gmbh, and Axel cyrille

Ngonga Ngomo. Bioasq: A challenge on large-scale biomedical semantic indexing and question answering.
24. Grigorios Tsoumakas, Manos Laliotis, Nikos Markantonatos, and Ioannis Vlahavas. Large-scale semantic indexing of biomedical publications at bioasq.