An Incremental Approach to MEDLINE MeSH Indexing

Dongqing Zhu[1], Dingcheng Li[2],
Ben Carterette[1], and Hongfang Liu[2]

[1] University of Delaware, 101 Smith Hall, Newark, DE 19716, USA
{zhu,carteret}@cis.udel.edu
[2] Mayo Clinic, 200 First Street SW, Rochester, MN 55905, USA
{li.dingcheng,liu.hongfang}@mayo.edu

Abstract. As an increasing number of new journal articles being added to the MEDLINE database each year, it becomes imperative to build effective systems that can automatically suggest Medical Subject Headings (MeSH) to reduce effort from human annotators. In this paper, we propose three approaches, one building upon another in an incremental way, to automatic MeSH term suggestion: 1) MetaMap-based labeling, which relies on the MetaMap tool to detect MeSH-related concepts for indexing; 2) Search-based labeling, which builds on MetaMap-based approach and further leverages information retrieval techniques for finding similar articles whose existing annotations are used for MeSH suggestion; 3) LLDA-based labeling, which further trains a multi-label classifier based on MeSH ontology for MeSH candidate list pruning. The evaluation on the BioASQ challenge data shows promising results.

Keywords: Biomedical Semantic Indexing, Information Retrieval, Query Formulation, Topic Modeling

1 Introduction

MEDLINE[3] is the U.S. National Library of Medicine's (NLM) premier bibliographic database that contains over 19 million references to journal articles in life sciences with a concentration on biomedicine. MEDLINE records are indexed with Medical Subject Headings (MeSH) and by highly qualified domain experts.

Currently, there are about 0.7 million new journal articles being added to the MEDLINE databsese each year, which makes manual indexing extremely difficult and costly. Besides, the indexing consistency among domain experts is unpredictable and hard to control. Funk and Reid [1] reported a consistency of only 48.2% for MeSH-based indexing. Moreover, the relatively slow speed of indexing new articles and making them available in the search database hinders technology transfer and advancement more or less.

[3] http://www.nlm.nih.gov/pubs/factsheets/medline.html

In order to alleviate those problems, the NLM has developed a tool called Medical Text Indexer (MTI) to assist human annotators with MEDLINE article indexing [2]. Recently, the BioASQ challenge [3, 4] has initiated a series of shared tasks, among which Task 1a (Large-scale Biomedical Indexing) specifically targets on the MEDLINE indexing problem and encourages participants to contribute to the development of tools and systems to automatically suggest MeSH terms to MEDLINE literature.

In this paper, we propose three approaches, one building upon another in an incremental way, to automatic MeSH term suggestion: 1) MetaMap-based labeling, which relies on the MetaMap tool to detect MeSH-related concepts for indexing; 2) Search-based labeling, which builds on MetaMap-based approach and further leverages information retrieval techniques for finding similar articles whose existing annotations are used for MeSH suggestion; 3) LLDA-based labeling, which further trains a multi-label classifier based on MeSH ontology for MeSH candidate list pruning. The evaluation on the BioASQ challenge data presents promising results and produces interesting findings that may benefit future exploration.

The rest of the paper proceeds as follows: Section 2 highlights the related work. Section 3 describes the data and the task. Then, Section 4 elaborates our methods and Section 5 presents and discusses the evaluation results. Finally, Section 6 summarizes our work and points out future research directions.

2 Related Work

There are many existing works related to MeSH-based MEDLINE indexing. We will only highlight a few that are most relevant to our approaches in this section.

The most well-known system for MeSH indexing is the Medical Text Indexer (MTI) developed at NLM [5, 2]. The latest version of MTI consists of three major components: MetaMap [6], Trigram Phrase Matching, and Trigram PubMed Related Citations (Trigram PRC) [7]. MetaMap is a tool that can map text into UMLS concepts, represented by Concept Unique Identifiers (CUI). Trigram Phrase Matching[4] is a method of identifying phrases that have a high probability of being synonyms. It is based on the idea of representing each phrase by a set of character trigrams that are extracted from that phrase. The character trigrams are used as key terms in a representation of the phrase much as words are used as key terms to represent a document. The similarity of phrases is then computed using the vector cosine similarity measure. Trigram PRC is a probabilistic topic-based model for retrieving and ranking related documents with respect to the target document. These three components work independently and in parallel to suggest separate lists of MeSH candidates which are merged

[4] http://ii.nlm.nih.gov/MTI/trigram.shtml

in the final stage. Our search-based systems differ from MTI in that we used MetaMap and information retrieval techniques in a sequential way.

Jimeno-Yepes et al. [8] analyzed the MeSH recommended by MTI and studied a few issues of using machine learning approaches for MeSH suggestion. Their work gives useful insights for improving our LLDA-based system.

Huang et al. [9] formulated the indexing task as a ranking problem. In particular, they used a learning-to-rank algorithm to rank MeSH main headings that were extracted from 20 neighbor documents of the target document. Our search-based approach differs from theirs in that we proposed different query formulation strategies and MeSH candidate ranking methods. We also explored the impact of system parameters on the performance.

3 Data and Task

The training set provided by BioASQ challenge contains over 10 million journal articles, each of which consists of the title, abstract, PubMed identifier (PMID), and gold standard MeSH labels that are manually annotated by experts. BioASQ releases 18 test sets of different sizes (ranging from hundreds to tens of thousands documents) over 18 week. Each set consists of new journal articles (<title, abstract, PMID> triples) that have not been annotated or indexed into the PubMed database. The task is to develop systems that can automatically suggest MeSH terms to the unlabeled articles.

We remove duplicated articles that have same PMID in the training set and obtain a pool of 10,699,707 articles with unique PMID[5]. Furthermore, we randomly sample 4,000 articles from this pool for system training and testing respectively, as shown in Table 1. The reason for creating our own test set instead of using the official ones are two-fold: 1) the gold standard annotations for the BioASQ official test sets are not available to the participants by the time of writing this paper, however, we want to give detailed analysis of our system and present comparable results on both training and testing sets in this paper; 2) we keep adding new features into our system over the whole competition time span, and the official evaluation results do not always reflect the latest development progress of our systems.

[5] Note that we will not distinguish between the singular and plural forms of acronyms (such as CUI and DUI) in this paper, i.e., PMID can either stands for PubMed identifier or identifiers depending on the context.

Table 1. Data

Data	# of articles	Purpose
TRN-0	10,691,707	Training data from BioASQ
TRN-1	10,687,707	Subset of TN-0 used for finding similar articles to the target article
TRN-2	2,000	Subset of TN-0 used for optimizing system parameters
TET	2,000	Subset of TN-0 used for evaluation

4 Systems

4.1 MetaMap-based Labeling

Concept Detection We run MetaMap against an article while restricting its resource to MeSH (i.e., MeSH ontology). We obtain and store the following information: 1) Concepts (denoted as K) which are phrases or terms that map to UMLS CUI; 2) The list L of MetaMap generated CUI candidates c with confidence scores S_c for each K; 3) The negation information for each K.

Figure 1 gives a concrete example in which "cervical cancer" is a detected, non-negated phrase concept (i.e., K) with a MeSH-related CUI candidates list L (C0007847, C0302592, C0006826, C0998265, and etc.). Each c in L has its individual confidence scores S_c, e.g., "C0006826" has a confidence score of 861.

```
{ "candidates": [
                { "cui": "C0007847",
                    "name": "cervical cancer",
                    "preferredname": "Malignant tumor of cervix",
                    "score": 1000                        },
                { "cui": "C0302592",
                    "name": "CERVICAL CANCER",
                    "preferredname": "Cervix carcinoma",
                    "score": 1000                        },
                           ...                           },
                { "cui": "C0998265",
                    "name": "Cancer",
                    "preferredname": "Cancer Genus",
                    "score": 861                         },
                           ...
                 ],
            "neg": false,
            "phrase": "cervical cancer"                         }
```

Fig. 1. CUI candidates for a detected concept by MetaMap, shown as a JSON object.

Concept Weighting We first select non-negated CUI whose confidence scores are above the threshold h. Then, we merge and rank these selected c by aggregating their weighted confidence scores. Here we use superscripts T and A

to denote title and abstract respectively. The final ranking score of a specific c looks like:

$$\text{score}(c) = \alpha \sum_{L \in T} S_c^L + \beta \sum_{L \in A} S_c^L, \tag{1}$$

where α and β are the weights assigned to c in abstract and title respectively, L is the candidate list for each detected concept K, S_c^L is the confidence score of c in list L. If L does not contain c, S_c^L will be zero.

In particular, we fix β to 1.0. However, we vary α (i.e., the weights of c^T) to explore the optimal value of α. We use Equation 1 to rank c and select the top-ranked m ones. Finally, we convert the selected c to MeSH Descriptor Unique Identifiers (DUI).

The above method has three free parameters, i.e., h, α, and m. We set their values by exploring the parameter space as will be described in Section 5.1.

4.2 Search-based Labeling

In this section, we describe another approach for MeSH suggestion which is based on information retrieval techniques. As aforementioned, our approach starts by finding related articles to the target article, and then leverages their existing annotations to suggest MeSH candidates for the target article.

We use the open-source search engine Indri[6] [10] to build an index for the training set 1. In particular, we remove stop words in the title and abstract by using a medical stoplist [11] and use the Porter stemmer for stemming words.

There are three components in our retrieval system: 1) the retrieval model for ranking documents; 2) the query generation module which formulates a query based on the target article; and 3) MeSH aggregation module that aggregates and scores the existing annotations for labeling the target article. Next, we will describe each component in detail.

Retrieval Model

Our retrieval model computes the relevance score of a document based on the following function:

$$\text{score}(Q, D) = \sum_{q_i \in Q} w_i f(q_i, D), \tag{2}$$

where w_i is the weight associated with a matched query term q_i, and $f(q_i, D)$ is the query term matching function defined as:

$$f(q_i, D) = \log \frac{\text{tf}_{q_i,D} + \mu \frac{\text{tf}_{q_i,C}}{|C|}}{|D| + \mu}, \tag{3}$$

[6] `http://www.lemurproject.org/indri/`

where q_i is the ith query term used for text matching. Note that q_i can be either a single word or a phrase. $|D|$ and $|C|$ are the document and collection lengths in words respectively, $\text{tf}_{q_i,D}$ and $\text{tf}_{q_i,C}$ are the document and collection term frequencies of q_i respectively, and μ is the Dirichlet smoothing parameter. Smoothing is a common technique for estimating the probability of unseen words in the documents [12, 13].

The above matching function assigns a score to each match of a query term q, and Equation 2 aggregates the scores based on weight w to obtain the final document relevance score. We implement this retrieval model in Indri by formulating queries that look like: #weight(w_0 q_0 w_1 q_1 ...w_i q_i...).

Query Formulation

Our next step is to formulate a query Q that can be representative of the content of the article. In this section, we will describe how we effectively generate query terms q as well as their weights w for our ranking function shown by Equation 2.

Term Query (TQ) The first type of queries is based on single words/terms in the article, i.e., terms in a term query are all single-word expressions. In particular, we formulate Q based on words occurring in the concepts detected by MetaMap from both title and abstract, i.e., query terms q come from words in K^T and K^A. Similar to what we have described in Section 4.1, we assign equal weight 1.0 to all q^A (i.e., query terms from K^A), but use a varying weight γ for all q^T. A term query in Indri looks like:

```
#weight(2.0 examination 2.0 cow 2.0 ultrasonographic 3.0 navel
3.0 urachal 3.0 extra-abdominal 2.0 pathologic 2.0 abscess)
```

Phrase Query (PQ) The second type of queries are from K^T and K^A directly, i.e., we use concepts (usually phrases) as query terms q_i. Again, we assign equal weight 1.0 to all q^A (i.e., K^A), but use a varying weight γ for all q^T (i.e., K^T). The following shows an Indri phrase query example:

```
#weight(3.5 #uw2(hiv-1 infection) 4.5 #uw2(differential
susceptibility) 2.0 #uw2(actin dynamics) 2.0 actin
4.5 #uw2(cortical actin) 4.5 #uw3(naive t cells)
2.5 dichotomy 3.5 #uw2(human memory)
3.5 #uw3(chemotactic actin activity) 2.0 cd45ro)
```

"#uwN(t1 t2)" means words t1 and t2 can be in any order within a text window of N words, and thus it takes possible variants of a phrase into consideration.

Long Query (LQ) The term query considers single words only and ignores the term proximity information in concepts. Thus, it may hurt retrieval precision. On the other hand, the phrase query poses "stricter" matching criteria, i.e., if a relevant document does not have an exact match for a concept phrase K (e.g., for "#uw2(hearing loss)" to match "loss of hearing"), it will not get any credit by Equation 3. Therefore, we formulate a long query that consists of qi from both TQ and PQ, i.e., both single word query terms and phrase query terms.

For the above three types of queries (i.e., TQ, PQ, and LQ), to prevent Q from being too long (computationally expensive when retrieving against a large database) we remove q that occur only once in the the abstract and title combined unless all the terms occur only once (which is a very rare case).

Result Aggregation

For each target article, we formulate query Q and rank documents based on Equations 2 and 3. Then, we take the top-ranked k documents, weight their existing MeSH annotations (i.e., DUI) by their individual relevance scores shown in Equation 2, and aggregate the wegiths for each DUI. Finally, we select the top-ranked m DUI as MeSH annotations for the target article.

In our Search-based Labeling method, we will also allow three free parameters: μ (the Dirichlet parameter in Equation 3), k (the number of top-ranked documents used for DUI aggregation) and m (the number of DUI). We will discuss how to set these parameter in Section 5.1.

4.3 LLDA-based Labeling

The MeSH indexing can also be cast as a multi-labeled classification task. Therefore, the labeled latent Dirichlet allocation (LLDA) [14], a supervised variation of the unsupervised LDA used for credit attribution in multi-labeled corpora, fits well to this MeSH indexing task.

In LDA, each document may be viewed as a mixture of various topics, and the topic distribution has a Dirichlet prior. As an extension of LDA, LLDA further incorporates observed label information, and thus can generate topics that predict labels. Therefore, we train an LLDA model with a subset (∼15%) of set of TRN-0 (see Table1) and use the existing MeSH annotations as labels.

However, the MeSH ontology contains too many labels (over 25,000 descriptors) for our LLDA to handle. Therefore, we only use MeSH terms at the category level (i.e., children of the root) to form our label set. MeSH annotations of articles are all converted to their corresponding ancestors in this category-level set.

Given a target article, our LLDA will predict its category level labels which will be further used to filter irrelevant labels assigned by previous MetaMap-based or search-based systems. Our goal is to remove false positives and improve precision.

5 Evaluation

BioASQ evaluates the MeSH annotations by two different groups of metrics, i.e., flat measures and hierarchical measures, among which Micro F-measure (MiF) and Lowest Common Ancestor F-measure (LCA-F) are the primary evaluation metrics respectively. Thus, we will report Precision, Recall, F-measure for both Microaveraging and LCA measures, i.e., (MiP, MiR, MiF) and (LCA-P, LCA-R, LCA-F).

We have five systems, namely the MetaMap-based system (MM), Search-based systems (TQ, PQ, and LQ), and the LLDA-based system (LLDA). Note that for convenience in the rest of paper we will refer to each system by their short names given in parentheses.

5.1 Parameter Exploration

As mentioned in Section 3, we use set TRN-2 to train system parameters. In this section, we show how each free parameter affects performance.

MetaMap-based Labeling

System MM has three free parameters, i.e., h (title concept weight), α (confidence score threshold for CUI candidates), and m (number of DUI in the final suggested list). To get the best setting for MM, we explore the range (400, 1000, 100) for h, (0, 5.0, 0.5) for α, and (8, 41, 4) for m, and try all different value combinations. Note that the third element is the range is the step size.

Table 2(a) shows that MM achieves the best MiF score (0.2697) when $w = 4.5$, $h = 600$, and $m = 12$ (the best setting). To explore the impact of each free parameter on the performance, we fix two of them based on the best setting, vary the left one, and obtain the performance curves as shown in the left column of Figure 2.

In particular, the performance curve in Figure 2(a), where we vary the weight of title concepts, shows that we should assign higher weights to the title concepts.

Table 2. Evaluation

(a) Training

System	MiP	MiR	**MiF**	LCA-P	LCA-R	**LCA-F**
MM ($w = 4.5$, $h = 600$, $m = 12$)	0.2617	0.2781	0.2697	0.3303	0.2831	0.2931
TQ ($\mu = 125$, $k = 20$, $m = 12$)	0.5766	0.5058	0.5389	0.4143	0.4655	0.3978

(b) Testing

System	MiP	MiR	**MiF**	LCA-P	LCA-R	**LCA-F**
MM ($w = 4.5$, $h = 600$, $m = 12$)	0.2660	0.2780	0.2719	0.3322	0.2862	0.2963
TQ ($\mu = 125$, $k = 20$, $m = 12$)	0.5842	0.5044	**0.5413**	0.4697	0.3979	**0.4168**
PQ (same setting as TQ)	0.5141	0.4389	0.4735	0.4257	0.3496	0.3710
LQ (same setting as TQ)	0.5748	0.4953	0.5321	0.4638	0.3918	0.4110
LLDA ($\mu = 125$, $k = 20$, $m = 20$)	0.5843	0.4400	0.5017	0.3322	0.2842	0.2950

This is expected because the title of an article usually contains the most representative information and the concepts in title are very likely to associate with MeSH annotations.

In Figure 2(b), as we lower the confidence score threshold for MetaMap CUI candidates from 1000 to 700, the precision declines while the recall improves. However, the precision bounces back when h is below 700, and the best performance for MiF, MiP, and MiR all appears at 600.

In Figure 2(c), the precision decreases and the recall increases, both monotonically, as we increase m, the number of DUI for annotating an article. This is also expected because DUI ranked lower down the list are less likely to be correct annotations, and consequently hurt the precision but improve the recall.

Search-based Labeling

Now we explore the parameter setting for search-based systems, which also have three free parameters: μ (the Dirichlet parameter in Equation 3), k (the number of top-ranked documents used for DUI aggregation) and m (the number of DUI). In particular, we will train system TQ and use it as a reference for setting corresponding parameters in PQ and LQ.

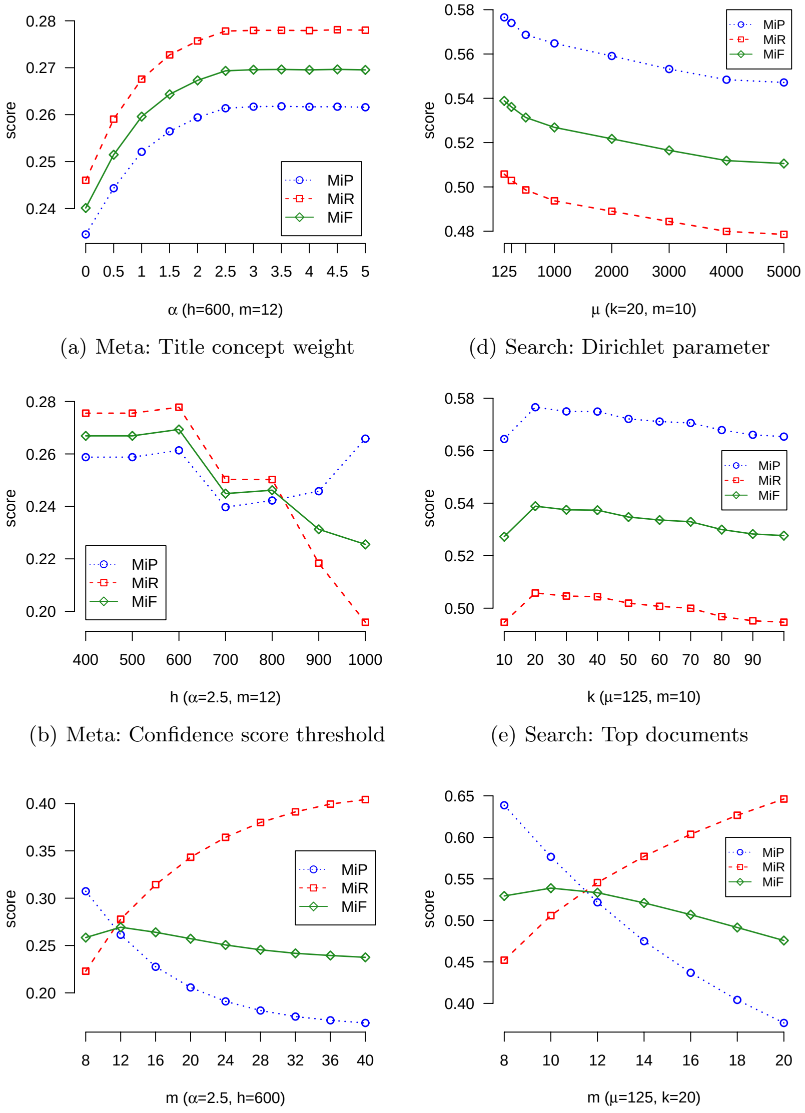
Table 2(a) shows that TQ achieves the best MiF score (0.5389) when $\mu = 125$, $k = 20$, and $m = 12$ (the best setting). Again, to explore the impact of each free parameter on the performance, we fix two of them based on the best setting, vary the left one, and obtain the performance curves as shown in the right column of Figure 2.

In Figure 2(d), the performance degrades as we increase μ (i.e., more smoothing with the collection-level statistic). This may be because our search-based labeling uses the top-ranked documents for MeSH suggestion and it desires a document set that has a high precision, and on the other hand, less smoothing makes sure that the relevant information remain highly concentrated in these documents which consequently appear among the top of the rank list.

In Figure 2(e), as we increase the number of top-ranked documents the performance peaks early at $k = 20$ and declines after that point, which is expected because our search-based labeling desires only a few highly relevant documents that can provide a more reliable set of MeSH candidates.

Figure 2(f) looks very similar to Figure 2(c) in which the system tries to strike a balance between precision and recall by varying m. However, method TQ only needs top 10 candidates to achieve the best MiF, as opposed to top 12 in MetaMap-based labeling, indicating that our search-based method is more precision-focused.

Due to the high similarity among search-based systems, we will simply use the best parameter setting of TQ ($\mu = 125$, $k = 20$, and $m = 12$) for systems PQ and LQ in our testing stage which is presented next.



(a) Meta: Title concept weight (d) Search: Dirichlet parameter

(b) Meta: Confidence score threshold (e) Search: Top documents

(c) Meta: Cutoff of MeSH candidate list (f) Search: Cutoff of MeSH candidate list

Fig. 2. Parameter setting for MetaMap-based and Search-based labeling methods

5.2 Test and Comparison

Table 2(b) shows the evaluation results on the test set (i.e., set TET in Table 1). System MM and TQ both obtain comparable results to those on the training set, indicating that our parameter setting process results in consistent performance.

System TQ, as the simplest among search-based systems, achieves the best performance. Though a direct comparison between TQ and the top-performing BioASQ challenge systems[7] is impossible at this stage (since the gold standards have not been released and we have have not submitted runs from this best system), TQ presents quite promising results. However, systems PQ and LQ are doing worse than TQ. The reason might be that the simple term frequency based phrase weighting strategy could not well distinguish important concepts from unimportant ones, and consequently hurts the precision.

In System LLDA, we use the predicted category labels to prune the annotation list from system TQ. We start with a long candidate list by setting m to 20, and then prune this list with LLDA. Table 2(b) shows that LLDA does not produce positive results. This might be because the category level MeSH terms are broad concepts that are not discriminative enough to distinguish one from another.

6 Conclusion and Future Work

In this paper, we proposed three approaches for automatic MeSH term suggestion: 1) MetaMap-based labeling, which relies on the MetaMap tool to detect MeSH-related concepts for indexing; 2) Search-based labeling, which builds upon MetaMap-based approach and further leverages information retrieval techniques for finding similar articles with existing annotations and uses them for MeSH suggestion; 3) LLDA-based labeling, which further builds on Search-based labeling and trains a multi-label classifier based on MeSH ontology for MeSH candidate list pruning.

Our evaluation on the BioASQ challenge data showed promising results for the Search-based labeling. In addition, we explored the impact of different system parameters (e.g., the weight for title concepts, CUI confidence scores, Dirichlet prior, number of top-ranked documents, etc.) on the system performance. We also proposed a new multi-label classification system based on LLDA for MeSH candidate list pruning. We believe the research findings presented in this paper would be useful for designing similar systems for biomedical semantic indexing.

For future work, we plan to explore better concept weighting strategies (e.g., by incorporating corpus-level statistics or using information from external sources) for systems PQ and LQ. As for the LLDA-based labeling, we will extend LLDA model by leveraging hierarchical information in MeSH ontology. In addition, we plan to compare our approaches with existing methodologies and carry out a thorough error analysis to look for aspects that we can further improve.

[7] http://bioasq.lip6.fr/results/

7 Acknowledgements

The work was supported by ABI: 0845523 from United States National Science Foundation, R01LM009959 from United States National Institute of Health. The challenge is organized in the context of the BioASQ project. BioASQ is a Support Action funded under the European Commission FP7 ICT Work Programme, within the Intelligent Information Management objective (ICT- 2011.4.4(d)).

References

1. Funk, M.E., Reid, C.A.: Indexing consistency in medline. Bulletin of the Medical Library Association **71**(2) (1983) 176
2. Aronson, A.R., Mork, J.G., Gay, C.W., Humphrey, S.M., Rogers, W.J.: The NLM indexing initiative's medical text indexer. Medinfo **11**(Pt 1) (2004) 268–72
3. Tsatsaronis, G., Schroeder, M., Paliouras, G., Almirantis, Y., Androutsopoulos, I., Gaussier, E., Gallinari, P., Artieres, T., Alvers, M.R., Zschunke, M., et al.: Bioasq: A challenge on large-scale biomedical semantic indexing and question answering. In: 2012 AAAI Fall Symposium Series. (2012)
4. Androutsopoulos, I.: A challenge on large-scale biomedical semantic indexing and question answering. In: BioNLP Workshop. (8 2013)
5. Kim, W., Aronson, A.R., Wilbur, W.J.: Automatic mesh term assignment and quality assessment. In: Proceedings of the AMIA Symposium, American Medical Informatics Association (2001) 319
6. Aronson, A.R.: Effective mapping of biomedical text to the UMLS metathesaurus: The MetaMap program. Proceedings of AMIA Symposium (2001) 17–21
7. Lin, J., Wilbur, W.J.: Pubmed related articles: a probabilistic topic-based model for content similarity. BMC bioinformatics **8**(1) (2007) 423
8. Jimeno Yepes, A., Mork, J.G., Wilkowski, B., Demner Fushman, D., Aronson, A.R.: Medline mesh indexing: lessons learned from machine learning and future directions. In: Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium, ACM (2012) 737–742
9. Huang, M., Névéol, A., Lu, Z.: Recommending mesh terms for annotating biomedical articles. Journal of the American Medical Informatics Association **18**(5) (2011) 660–667
10. Strohman, T., Metzler, D., Turtle, H., Croft, W.B.: Indri: A language model-based search engine for complex queries. In: Proceedings of the International Conference on Intelligent Analysis. (2005)
11. Hersh, W.: Information Retrieval: A Health and Biomedical Perspective. Third edn. Health Informatics. Springer (2009)
12. Chen, S.F., Goodman, J.: An empirical study of smoothing techniques for language modeling. In: Proceedings of the 34th annual meeting on Association for Computational Linguistics, Association for Computational Linguistics (1996) 310–318
13. Zhai, C., Lafferty, J.: A study of smoothing methods for language models applied to ad hoc information retrieval. In: Proceedings of SIGIR. (2001) 334–342
14. Ramage, D., Hall, D., Nallapati, R., Manning, C.D.: Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In: Proceedings of EMNLP: Volume 1 - Volume 1. (2009) 248–256