Comparison and combination of several MeSH indexing approaches

Antonio Jose Jimeno Yepes, PhD[1,2], James G. Mork, MSc[2], Dina Demner-Fushman, MD, PhD[2], Alan R. Aronson, PhD[2]
[1]NICTA Victoria Research Lab, Melbourne VIC 3010, Australia; [2]National Library of Medicine, 8600 Rockville Pike, Bethesda, MD 20894, USA

Abstract

MeSH indexing of MEDLINE is becoming a more difficult task for the group of highly qualified indexing staff at the US National Library of Medicine, due to the large yearly growth of MEDLINE and the increasing size of MeSH. Since 2002, this task has been assisted by the Medical Text Indexer or MTI program. We extend previous machine learning analysis by adding a more diverse set of MeSH headings targeting examples where MTI has been shown to perform poorly. Machine learning algorithms exceed MTI's performance on MeSH headings that are used very frequently and headings for which the indexing frequency is very low. We find that when we combine the MTI suggestions and the prediction of the learning algorithms, the performance improves compared to any single method for most of the evaluated MeSH headings.

Introduction

The NLM indexing process involves analysis of journal articles for subject matter and subsequent assignment of appropriate subject headings, drawn from MeSH®, the NLM controlled vocabulary. Maintaining the high quality of MEDLINE® indexing is made difficult by the demand of the ever increasing size of the biomedical literature and MeSH on a relatively small group of highly qualified indexing contractors and staff at the US National Library of Medicine (NLM). We hope that the situation can be eased through improvements to the recommendations made by NLM's indexing tool, the Medical Text Indexer (MTI)[1]. MTI is a support tool for assisting indexers as they add MeSH indexing to MEDLINE citations; further details about MTI are presented in the Methods section.

Our motivation is to improve MTI's recommendations using automatic methods, viz. machine learning, because previously indexed citations are available as training data from MEDLINE. Automatic or semi-automatic methods to improve the indexing of selected MeSH headings (MHs) are preferred due to the large number of MeSH headings. We previously explored a semi-automatic bottom-up approach[2], which suggested terms that should be considered for building indexing rules. We also evaluated several machine learning algorithms on selected MeSH headings[3,4,5]. We found that the bottom-up approach improved the performance on complex MeSH headings like *Molecular Sequence Data*, while the machine learning methods contributed to the improvement of MTI's performance on a set of MeSH headings named Check Tags (special set of MeSH headings defined in the Methods section).

Despite all these efforts, we concluded that assignment of MeSH headings is a difficult process and that no single method performed better than another one over the whole range of MeSH headings[4,5]. Furthermore, there are some issues inherent to MeSH indexing and text categorization tasks that need to be taken into account when using machine learning:

1. Imbalance between the number of citations indexed with a MeSH heading (positive instances) and the number of citations not indexed with it (negative instances). Usually, the number of negative instances overwhelms the number of positives. Machine learning algorithms tend to have problems with imbalanced sets, building models that tend to predict all previously unseen instances as belonging to the majority class.
2. Even if a MeSH heading is correctly identified with a citation, it might not be significant enough to be included in the indexing.
3. Inconsistencies in the annotations might appear due to:

   (a) Inconsistency between MeSH indexers[6].

(b) Changes in indexing policy that, over time, can introduce inconsistencies with previously-indexed citations. These changes can even apply to routine changes to the structure of MeSH.

In this work, we extend previous analysis adding a more diverse set of MeSH headings, targeting examples in which MTI has shown to have poor performance. We expect to overcome some of the problems presented above by: 1) using a larger number of machine learning methods that have been chosen to deal with a large number of examples. 2) by preparing our training and testing sets using MHs which were already in MeSH during the current indexing period, and 3) by selecting recently indexed citations. In addition, machine learning algorithms exceed MTI's performance on very frequent MeSH headings and headings for which the indexing frequency is very low. We find that when we combine the output of the algorithms, the performance improves. This shows that the indexing methods are complementary to each other. The aggregation of the indexing algorithms through voting improves the indexing performance for most of the MeSH headings.

Related work

The task of MeSH indexing has been considered as a text categorization problem in the machine learning community. We find that most of the methods fit either into pattern matching methods which are based on a reference terminology (like UMLS® or MeSH) and machine learning approaches which learn a model from examples of previously indexed citations. Among the pattern matching methods, we find MetaMap[7], as mentioned above, and an information retrieval approach by Ruch[8]. Pattern matching considers only the inner structure of the terms but not the terms with which they co-occur. This means that if a document is related to a MeSH heading but the heading does not appear in the reference source, it will not be suggested.

Initial work using machine learning was based on the OHSUMED collection[9] containing all MEDLINE citations in 270 medical journals over a five-year period (1987-1991) including MeSH indexing provided for a large body of data that enabled us to view MH assignment as a classification problem. The scope of the collection determines the subset of MeSH that can be explored. For example, Lewis et al.[10] and Ruiz and Srinivasan[11] used 49 categories related to heart diseases with at least 75 training documents, and Yetisgen-Yildiz and Pratt[12] expanded the number of headings to 634 disease categories. Poulter et al.[13] provides an overview of these and other studies of classification methods applied to MEDLINE and MeSH subsets. Other machine learning algorithms have been evaluated which rely on a more complex representation of the citations which do not rely only on unigrams or bigrams, e.g., learning based on Inductive Logic Programming[14].

MeSH 2013 contains 26,853 terms and over 214,000 entry terms to assist the indexers in determining the appropriate terms to assign to a MEDLINE citation. Small scale studies with machine learning approaches already exist[12,15]. On the other hand, the presence of a large number of MeSH headings has forced machine learning approaches to be combined with information retrieval methods designed to reduce the search space. For instance, PRC and k-NN approaches by Trieschnigg et al.[16] and Huang et al. look for similar citations in MEDLINE and predict MeSH headings by a voting mechanism on the top-scoring citations. An approach based on a deterministic variant of Random Indexing has been proposed as well to overcome the size problem[17].

In addition to the size problem, imbalance in the data set is another pervasive problem. Limited work exists to tackle this problem. Yeganova et al.[21] evaluated cost based methods and a variant of Support Vector Machine (SVM) based on modified Huber Loss, which showed better performance than SVM in some cases but better performance compared to cost-based approaches. We have previously evaluated several approaches, including oversampling of the minority class and SVM trained on multivariate measures[3].

Methods

In this section, we present how the framework is trained, how it is used to index citations, and we show the base methods used for MeSH indexing. The methods include MTI and several machine learning algorithms. For training and testing, we have used a data set of MEDLINE citations from November 2012 to February 2013. Having a recent data set ensures that the latest indexing policies are taken into account. Experience has shown that indexing policy changes from year to year. This changing policy can lead to problems training a machine learning algorithm when it is confronted with these conflicting examples in the final indexing. The data set has a total of 143,853 citations. From this set 2/3 were selected for training (94,942) and 1/3 was selected for testing (48,911) the indexing methods. This data set is available from the MTI_ML package web site (http://ii.nlm nih.gov/MTI_ML/index.shtml).

MeSH heading selection

For this study, we have selected MeSH headings from four different groups. The first group includes a set of Check Tags for which MTI is already using machine learning to produce recommendations. We want to use this as a baseline for judging the improvements over existing methods. For the additional groups (defined below) we selected the MeSH headings according to MTI's recent performance. From each group, we have selected the 10 top MeSH headings based on their frequency within the group. The description for each of these groups is presented below:

1. **Check Tag Performance:** Table 1 shows F_1-measure (F_1) performance on the list of Check Tags. Check Tags are a special class of MeSH headings routinely considered for every article, which cover species, sex and human age groups, historical periods, and pregnancy. We studied the indexing of Check Tags in previous work,[4,5] and some of them are now suggested by MTI based on the machine learning models coming from that work. A total of 12 MHs are currently processed by the machine learning process.
2. **MeSH headings annotated by MTI with low precision:** Table 2 contains the top terms in this category sorted by frequency from MeSH headings with 50% or less precision.
3. **MeSH headings annotated by MTI with low recall**: Table 3 contains the top terms in this category sorted by frequency.
4. **MeSH headings for which MTI did not return any results:** Table 4 shows the top terms in this category. The frequency for all of these MeSH headings is 1% of the total number of instances of the testing set.

Indexing algorithms

MTI has two main components: MetaMap[18] and the PubMed Related Citations (PRC)[19] algorithm. MetaMap performs an analysis of the citations and annotates them with Unified Medical Language System (UMLS) concepts. Then, the mapping from UMLS to MeSH follows the Restrict-to-MeSH approach, which is based primarily on the semantic relationships among UMLS concepts. The PRC algorithm is a modified k-NN algorithm which relies on document similarity to assign MeSH headings. This method tends to increase the recall of MetaMap by proposing indexing candidates for MeSH headings which are not explicitly present in the citation but have a similar context. Finally, a post-processing step arranges the list of MeSH headings, and tailors the output to reflect NLM indexing policy. In addition, this post-processing step incorporates suggestions from indexers' feedback.

We have selected several learning algorithms. Due to the large number of examples, we have developed specific algorithms that can handle binary features efficiently, both in terms of memory and in computation requirements. We enumerate the learning algorithms below with implementation specifications. As in previous work, MeSH indexing is considered a binary classification. This means that each algorithm will predict if the document should be indexed or not with the MeSH heading it was trained for.

One of the algorithms that we have extensively used is AdaBoostM1 (Ada)[22] using an implementation of decision trees based on C4.5[23] as base learning algorithm. In previous work, Ada had performed well on the Check Tags set and we were interested on evaluating its performance with a larger set of MHs. Our implementation of the C4.5 relies on binary features, which provide a more efficient implementation of the decision tree in terms of memory and time required for training. As with many learning algorithms, the imbalance in the data set seems to bias the model towards the most frequent category. We have used oversampling with AdaBoostM1 to deal with this issue (Ada Over).

SVM has been shown to perform well on text categorization tasks[26]. We have used an implementation of SVM with linear kernel based on Hinge loss and stochastic gradient descent[27]. We have considered, as well, the modified Huber loss based on Zhang's work used by Yeganova et al., which has been shown to improve the performance of Hinge loss in the case of very imbalanced training sets[27]. It is a wide margin classifier with a quadratic loss function. We have limited our work to linear kernels due to the size of our data sets, but it would be worth exploring efficient implementations for learning with more complex kernels.

Finally, we have considered Naïve Bayes (NB) and logistic regression (LR), probabilistic methods that assume independence between the features. NB can be seen as a generative algorithm while LR is considered a discriminative version of it, targeting the posterior compared to NB that estimates the posterior probability given the priors and the evidence. We used the Mallet package[28] for NB and LR.

Except for MTI, Naïve Bayes and Logistic regression, all of the other algorithms are available from the MTI ML package (http://ii.nlm.nih.gov/MTI_ML/index.shtml). To ensure that all the algorithms used the same set of features, we converted the vector representation produced by the MTI ML package to an intermediate format allowed by Mallet.

Combinations of methods have theoretical properties that have proved to increase performance of individual methods or within the same set of methods.[29] Ensemble methods have been successfully applied in information retrieval[30]. Better performance of these methods has been observed as well in biomedical tasks other than text categorization[31,32]. We therefore first collected binary predictions of each of the indexing methods presented above (assign a given MH or not) for each citation in the test set, and then counted the votes of the predictions by the methods. If the sum of the votes was over a given threshold, the MeSH heading was predicted by this voting method. We have performed experiments with different voting thresholds based on the methods presented above. For example, vote 2 denotes that at least two indexing methods agree that the citation should be indexed with the MeSH heading under evaluation. Results of the most promising voting thresholds are shown in the Results section.

Results

In this section, we present the performance of the indexing methods on the selected MeSH headings. Results are spread over four tables. Each table contains a set of the MeSH headings based on the groups presented in the Methods section. Performance of the indexing algorithms is measured based on the F_1-measure (F_1), which is the harmonic mean between precision and recall. Each table contains the number of positive examples in the test set and the performance of the evaluated algorithms. This means that if, for instance, the MeSH heading Adolescent has 3,824 positives in the test set, there are 45,087 citations that have not been indexed with this MeSH heading. Considering the voting combination, we show only the results when two or three indexing methods agree. When a higher agreement level is required, the precision is higher but at the expense of a loss in recall.

First, we present the results for the Check Tag set, shown in Table 1. MTI's performance on these MHs is based on the performance of AdaBoost with oversampling, so it is not shown. We find that SVM shows a better performance in many of the MHs. NB shows a lower performance compared to LR, even though it typically approaches its best performance when the proportion of positive examples is high. The combination of methods improves over any of the individual methods. The combination of three methods seems to perform better when there are a larger number of positives. The best performing Check Tag is *Humans*, which is as well the most frequent of the MeSH headings.

Table 1. Check Tags performance, F_1.

MH	Positive	NB	LR	SVM	SVM HL	Ada	Ada Over	Vote 2	Vote 3
Adolescent	3824	0.3694	0.4144	0.4101	0.4126	0.3290	0.3891	**0.4708**	0.4383
Adult	8792	0.5162	0.5555	0.5700	0.5545	0.5622	0.5699	0.6214	**0.6225**
Aged	6151	0.4934	0.5376	0.5482	0.5365	0.5319	0.5614	0.5978	**0.6005**
Aged, 80 and over	2328	0.2996	0.3009	0.3055	0.2959	0.1892	0.3227	**0.3753**	0.3319
Child, Preschool	1573	0.1426	0.4396	0.4409	0.4363	0.4250	0.4954	**0.5129**	0.4895

Female	16483	0.6664	0.7373	0.7517	0.7298	0.7490	0.7454	0.7647	**0.7764**
Humans	35967	0.8932	0.9233	0.9269	0.9208	0.9261	0.9082	0.9260	**0.9337**
Infant	1281	0.0900	0.4142	0.4228	0.4067	0.3881	0.4441	**0.4796**	0.4450
Male	15530	0.6482	0.7150	0.7287	0.7082	0.7294	0.7227	0.7489	**0.7582**
Middle Aged	8392	0.5525	0.6077	0.6377	0.6121	0.6193	0.6371	0.6597	**0.6731**
Swine	285	0.0207	0.5681	0.6111	0.5978	0.6715	0.7071	**0.7323**	0.6641
Young Adult	3807	0.3371	0.3158	0.3046	0.3134	0.1642	0.2722	**0.3973**	0.3294

The low precision set is shown in Table 2. We find that MTI has better performance in a larger number of the examples compared to the learning algorithms and has a larger precision compared to the learning methods. On the other hand, when it is combined with other learning algorithms in the voting scheme, the performance is much higher. An exception to this is *Molecular Sequence Data*; this MH has been studied in previous work where a set of rules were developed using a bottom-up approach[2]. In comparison to the Check Tags, only the combination of two methods seems to improve the performance of individual methods. This seems to be the case in the other two MeSH heading groups as well.

Table 2. F_1 performance on the low precision performance set.

MH	Positive	MTI	NB	LR	SVM	SVM HL	Ada	Ada Over	Vote 2	Vote 3
Age Factors	889	0.0844	0.0079	0.1372	0.1244	0.1406	0.0387	0.1450	**0.1748**	0.0892
Brain	823	**0.5201**	0.0360	0.3384	0.3358	0.3309	0.3299	0.4182	0.4700	0.3992
Cell Line	781	0.2876	0.1267	0.2219	0.2265	0.2253	0.1094	0.2083	**0.3059**	0.2502
Cells, Cultured	1079	0.3046	0.2608	0.2735	0.2784	0.2665	0.1365	0.2688	**0.3894**	0.2983
Models, Molecular	851	0.4292	0.2860	0.3710	0.3734	0.3584	0.2000	0.3634	**0.4763**	0.3960
Molecular Sequence Data	1527	**0.5495**	0.4094	0.3116	0.2995	0.3275	0.1715	0.3195	0.5118	0.3938
RNA, Messenger	628	0.4477	0.0744	0.3698	0.3779	0.3618	0.3277	0.4385	**0.4626**	0.4158
Severity of Illness Index	751	0.1824	0.0056	0.1924	0.1742	0.1826	0.0888	0.1755	**0.2415**	0.1512
Time Factors	2153	0.0980	0.0538	0.1393	0.0924	0.1284	0.0274	0.0612	**0.1513**	0.0809
United States	2658	0.3585	0.2432	0.3269	0.3236	0.3323	0.2899	0.3655	**0.4128**	0.3504

The low recall performing MeSH headings are presented in Table 3. MTI's performance is still better than most of the learning algorithms; but when two indexing methods agree, the performance is better for almost all MeSH headings in this set. AdaBoost with oversampling improves the performance of most of the other learning algorithms. The MeSH headings Age Factors, Time Factors and United States have both low recall and low precision. Results for these MeSH headings are shown only in Table 2.

Table 3. F_1 performance on the low recall performance set.

MH	Positive	MTI	NB	LR	SVM	SVM HL	Ada	Ada Over	Vote 2	Vote 3
Child	2780	0.5836	0.3478	0.5168	0.5447	0.5084	0.5707	0.5723	**0.5854**	0.5776
Follow-Up Studies	1470	0.0407	0.2010	0.2300	0.2104	0.2178	0.1347	0.2269	**0.2741**	0.2049
Reproducibility of Results	1206	0.3191	0.1411	0.3094	0.3106	0.3138	0.2230	0.2923	**0.3722**	0.3179
Retrospective Studies	2183	**0.6608**	0.3972	0.6197	0.6317	0.6065	0.6580	0.6532	0.6502	0.6592
Risk Assessment	1014	**0.2556**	0.0084	0.1610	0.1387	0.1449	0.0728	0.1369	0.2189	0.1436
Risk Factors	2365	**0.4989**	0.3301	0.3769	0.3778	0.3722	0.3722	0.4276	0.4774	0.4496

Treatment Outcome	2999	0.4202	0.3859	0.3431	0.3393	0.3472	0.3472	0.3639	**0.4421**	0.4004

Finally, we present the MeSH headings for which MTI does not provide any recommendations. These MeSH headings, like *History, 19th Century*, do not match the content of the positive citations. As we can see in Table 4, the learning algorithms are capable of learning a model that can be used to index some of the mentions but do not really perform very well overall on these MeSH headings. In this case, the combination of methods only improves on a selected number of examples. We find as well that for the MeSH headings *Causality* and *Drug Therapy*, the learning algorithms cannot provide any recommendations. Looking at relevant citations, we find a large variety of possible indexing rules which require contextual information (e.g. identifying an effect for *Causality*, or identifying a specific disease targeted by *Drug Therapy*). Our bottom-up approach to indexing[2] could be considered for these two MeSH headings.

Table 4. No recall performance, F_1.

MH	Positive	MTI	NB	LR	SVM	SVM HL	Ada	Ada Over	Vote 2	Vote 3
Causality	54	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Drug Therapy	52	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
History, 19th Century	225	0.0000	0.0588	**0.1692**	0.1143	0.1641	0.1509	0.1812	0.1502	0.0672
History, 21st Century	318	0.0000	0.1748	0.0981	0.0413	0.0924	0.1497	**0.1925**	0.1123	0.0364
Mice, Mutant Strains	60	0.0000	0.0000	0.0000	0.0000	**0.0323**	0.0000	0.0000	0.0000	0.0000
Mortality	87	0.0000	0.0119	0.0625	0.0990	0.1000	0.0000	0.0840	**0.1132**	0.0645
Neoplasm Metastasis	119	0.0000	0.0000	0.1889	0.2111	0.2054	0.0822	0.2283	**0.2289**	0.1667
Radiotherapy	57	0.0000	0.0159	0.0000	0.0000	**0.0667**	**0.0667**	0.0580	0.0635	0.0000
Random Allocation	118	0.0000	0.0000	0.0548	0.0909	0.0755	0.0000	**0.0444**	**0.0964**	0.0882
Ultrasonography	87	0.0000	0.0000	0.0825	0.0645	0.1212	0.0000	**0.2240**	**0.2385**	0.0444

Discussion

As we have seen in the Results section, the combination of methods seems to perform better than any single method in almost all the evaluated MeSH headings. This is an example of the complexity involved in MeSH indexing and the complementarity of the methods, which are capable of covering different aspects required to index each one of the MeSH headings. Previous results with combination of methods based on voting[4,5] did not show this improvement, which could be attributed to the limited variety of methods used in the experiments. The voting mechanism seems to perform well when 2 or 3 indexing methods agree. This reinforces the complementarity of the indexing methods.

There are several possible reasons for this complementarity[28], the learning algorithms might not have enough training data compared to the size of the hypothesis space, so the learning algorithms might identify several hypothesis with similar performance. The second is that many learning algorithms perform a local search which might get stuck at local optima, e.g. the greedy splitting rule for decision trees as used in C4.5. This explains why AdaBoost with C4.5 will generally improve the performance of C4.5. In addition, this might help to average the inconsistencies that we find in the indexing. And finally, the true function might not be representable by any of the hypotheses. For instance, SVM with a linear kernel will try to identify the separating hyperplane, but a hyperplane might not perform well when several features are related.

Considering the number of indexing methods that need to agree in the voting combination, we find the following. The optimal number of combined indexing algorithms seems to be when two or three indexing algorithms agree. When we just take any suggestion by any indexing method (vote = 1), the recall is high but the precision is very low. On the other hand, the more systems are required to agree (vote > 3) the higher the precision but the lower the recall.

Other combination methods could be used with the indexing methods in which the confidence of each system could be used within a linear combination, reflecting as well the confidence of each one of the indexing methods.

MTI seems to perform better than individual learning algorithms in many MeSH headings. This does not seem to be the case when the MeSH headings tend to be very frequent, e.g. Check Tags, or not very frequent, e.g. as shown in Table 4. There are two possible explanations for this behavior. The first one is that MTI is a mixture of different methods, which makes it more robust to variations in data, while machine learning methods depend on the quality of the training data. Among the methods used by MTI, we find the PRC method which resembles k-NN and relies on MEDLINE citations already indexed. On the other hand, MTI relies on terminological resources which might not cover ways a MeSH heading might appear in MEDLINE citations or cases in which PRC might not be an appropriate method.

Looking closer at the performance of the learning algorithms, AdaBoostM1 has highest precision. The performance of AdaBoost improves with the oversampling. SVM HL has better performance with MeSH headings with little number of positive examples but in general SVM produces better results. In the no recall set, MTI's PRC component did not work; and a reason for this is that, despite the other learning algorithms, it is an instance based learning algorithm. This means that it is not building a model but comparing the current citation being indexed with already indexed ones available from MEDLINE based on the related citations algorithm[19]. Citations deal with many topics and in the case of not very frequent MeSH headings, it seems to fail to identify similar citations. Perhaps identifying relevant sentences in the citations denoting these low frequency MeSH headings and using those to recover related citations might improve the performance of PRC in these cases as well as the performance of other learning algorithms.

Comparing discriminative versus generative methods, LR performs better than NB in almost all the evaluated MeSH headings, except in a few cases. Despite having some headings with a large number of examples, LR still performs better compared to NB, which indicates that discriminant approaches are preferred, in contrast to NG and Jordan[33] findings in similar problems. Comparing LR and NB with SVM and SVM HL, we find that the large margin classifiers like SVM perform much better.

A large number of training examples is required to properly train a classifier. We can see this illustrated in Figure 1. The performance of the evaluated learning algorithms has been plotted versus the frequency of the MeSH headings. The more examples are available the better the performance of the classifiers. Examples of this are *Humans*, *Female* and *Male*. We find that the fewer the examples with a MeSH heading, the more difficult it is to train a model with good performance. There are exceptions like *Swine* with an F_1 over 0.7. (*Swine* belongs to the CT set and might be easily identified by a small set of key words denoting it in text.)

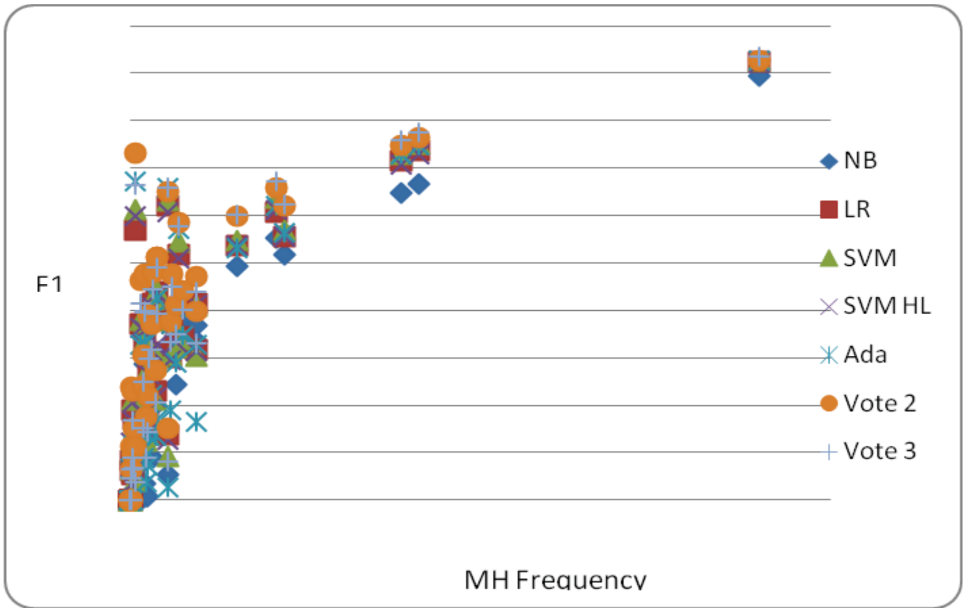

Figure 1. MeSH heading frequency versus F_1 measure comparison of the machine learning methods evaluated

Even though we did not deeply explore the imbalance problem, the results using AdaBoost with oversampling and the modified Huber loss improve the performance compared to other learning methods. In the case of the SVM, we have used the modified Huber Loss which has shown an improvement in the performance of the *no recall* set while the performance usually is better for SVM otherwise.

Conclusions and Future work

We have evaluated several indexing algorithms on a set of selected MeSH headings. The results confirm the conclusions from previous work that there is no single indexing algorithm which is better than another one for all the MeSH headings. In addition, we find that combining different indexing algorithms using a simple voting approach, improves the results by a significant amount compared to the best single performing method. We have evaluated a large range of learning algorithms; we believe that using a combination of different methods could further improve the performance of MTI.

As mentioned in the introduction, just finding the mention of a MeSH heading in the citation does not mean that it should be indexed with that MeSH heading. There are terms mentioned in the citation that might not be relevant to indexing. We would like to include an additional layer to the indexing algorithm in which sections or sentences of the citation are selected and used in the indexing instead of all the citation. There are two possibilities, for instance, the selection of sections relevant and irrelevant. The first one based on a Hidden Markov Model, which could be used to identify relevant sentences, which has been used already in information retrieval[34]. Another is to use a topic model given the current indexing as topics[35] or to use a model developed to identify different topics in the sentences of the citation[36], which might provide a finer grain indexing.

Finally, there will be some MeSH headings which might not be possible to index properly using the title and abstract available in MEDLINE citations since indexers use the full text during their work. On the other hand, we could still try to improve the recall by combining the knowledge-based methods, like the MetaMap and Restrict-to-MeSH part of MTI (based on MetaMap and MeSH respectively), and statistics from MEDLINE citations in an iterative feedback loop. MetaMap and Restrict-to-MeSH use information that is relevant which does not seem to be possible to learn from MEDLINE but it contains information that machine learning algorithms miss.

Funding

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

Acknowledgements

This work was also supported in part by the Intramural Research Program of the NIH, National Library of Medicine.

We would like to thank Lana Yeganova, Won Kim and W. John Wilbur for suggesting the use of the modified Huber Loss and for allowing us to use their tools to learn more about it.

References

1. Aronson, A. R., Bodenreider, O., Chang, H. F., Humphrey, S. M., Mork, J. G., Nelson, S. J., Rindflesch, T. C., et al. (2000). The NLM Indexing Initiative. *Proceedings of the AMIA Symposium*, 17–21. Retrieved from http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2243970&tool=pmcentrez&rendertype=abstract
2. Jimeno-Yepes, A., Wilkowski, B., Mork, J. G., Van Lenten, E., Fushman, D. D., & Aronson, A. R. (2011). A bottom-up approach to MEDLINE indexing recommendations. In *AMIA Annual Symposium Proceedings* (Vol. 2011, p. 1583). American Medical Informatics Association

3. Jimeno-Yepes, A., Mork, J. G., Wilkowski, B., Demner-Fushman, D., & Aronson, A. R. (2012). MEDLINE MeSH Indexing: Lessons Learned from Machine Learning and Future Directions - DTU Orbit. *IHI '12 Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*
4. Jimeno-Yepes, Antonio, et al. "Automatic algorithm selection for MeSH Heading indexing based on meta-learning.", *Fourth International Symposium on Languages in Biology and Medicine* (2011)
5. Jimeno-Yepes, Antonio, et al. "A One-Size-Fits-All Indexing Method Does Not Exist: Automatic Selection Based on Meta-Learning." Journal of Computing Science and Engineering 6.2 (2012): 151-160.
6. Funk, M. E., & Reid, C. A. (1983). Indexing consistency in MEDLINE. *Bulletin of the Medical Library Association*, *71*(2), 176.
7. Aronson, A. R., & Lang, F. M. (2010). An overview of MetaMap: historical perspective and recent advances. *Journal of the American Medical Informatics Association*, *17*(3), 229-236.
8. Ruch, P. (2006). Automatic assignment of biomedical categories: toward a generic approach. *Bioinformatics*, *22*(6), 658-664.
9. Hersh, W., Buckley, C., Leone, T. J., & Hickam, D. (1994, August). OHSUMED: an interactive retrieval evaluation and new large test collection for research. In Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 192-201). Springer-Verlag New York, Inc.
10. Lewis, D. D., Schapire, R. E., Callan, J. P., & Papka, R. (1996, August). Training algorithms for linear text classifiers. In Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 298-306). ACM
11. Ruiz, M. E., & Srinivasan, P. (1999, August). Hierarchical neural networks for text categorization (poster abstract). In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval (pp. 281-282). ACM
12. Yetisgen-Yildiz, M., & Pratt, W. (2005). The effect of feature representation on MEDLINE document classification. In AMIA Annual Symposium Proceedings (Vol. 2005, p. 849). American Medical Informatics Association
13. Poulter, G. L., Rubin, D. L., Altman, R. B., & Seoighe, C. (2008). MScanner: a classifier for retrieving Medline citations. *BMC bioinformatics*, *9*(1), 108.
14. A. Névéol, S. Shooshan, and V. Claveau. Automatic inference of indexing rules for MEDLINE. BMC bioinformatics, 9 (Suppl 11):S11, 2008
15. Aphinyanaphongs, Y., Tsamardinos, I., Statnikov, A., Hardin, D., & Aliferis, C. F. (2005). Text categorization models for high-quality article retrieval in internal medicine. *Journal of the American Medical Informatics Association*, *12*(2), 207-216.
16. Trieschnigg, D., Pezik, P., Lee, V., De Jong, F., Kraaij, W., & Rebholz-Schuhmann, D. (2009). MeSH Up: effective MeSH text classification for improved document retrieval. Bioinformatics, 25(11), 1412-1418.
17. Huang, M., Névéol, A., & Lu, Z. (2011). Recommending MeSH terms for annotating biomedical articles. *Journal of the American Medical Informatics Association*, *18*(5), 660-667.
18. Wahle, M., Widdows, D., Herskovic, J. R., Bernstam, E. V., & Cohen, T. (2012). Deterministic binary vectors for efficient automated indexing of MEDLINE/PubMed abstracts. In *AMIA Annual Symposium Proceedings* (Vol. 2012, p. 940). American Medical Informatics Association.
19. Lin, J., & Wilbur, W. J. (2007). PubMed related articles: a probabilistic topic-based model for content similarity. *BMC bioinformatics*, *8*(1), 423.
20. Bodenreider, O., Nelson, S. J., Hole, W. T., & Chang, H. F. (1998). Beyond synonymy: exploiting the UMLS semantics in mapping vocabularies. (C. G. Chute, Ed.)*Proceedings of the AMIA Symposium*, 815–819. Retrieved from http://www ncbi nlm nih.gov/pmc/articles/PMC2232139/
21. Yeganova, Lana, et al. "Text mining techniques for leveraging positively labeled data." *Proceedings of BioNLP 2011 Workshop*. Association for Computational Linguistics, 2011
22. Freund, Y., & Schapire, R. E. (1996, July). Experiments with a new boosting algorithm. In *Machine Learning-International Workshop-* (pp. 148-156). Morgan Kaufmann Publishers, Inc.
23. Quinlan, J. R. (1993). *C4. 5: programs for machine learning* (Vol. 1). Morgan Kaufmann.
24. Freund, Y., & Schapire, R. E. (1996, July). Experiments with a new boosting algorithm. In *Machine Learning-International Workshop-* (pp. 148-156). Morgan Kaufmann Publishers, Inc.
25. Yeganova, L., Comeau, D. C., Kim, W., & Wilbur, W. J. (2011, June). Text mining techniques for leveraging positively labeled data. In Proceedings of BioNLP 2011 Workshop (pp. 155-163). Association for Computational Linguistics.

26. Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. Machine learning: ECML-98, 137-142.
27. Zhang, Tong. "Solving large scale linear prediction problems using stochastic gradient descent algorithms." *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004.
28. McCallum, Andrew Kachites. "Mallet: A machine learning for language toolkit." (2002). http://mallet.cs.umass.edu/
29. Dietterich, T. (2000). Ensemble methods in machine learning. *Multiple classifier systems*, 1-15.
30. Croft, W. Bruce. "Combining approaches to information retrieval." *Advances in information retrieval*. Springer US, 2002. 1-36.
31. Kim, J. D., Ohta, T., Pyysalo, S., Kano, Y., & Tsujii, J. I. (2009, June). Overview of BioNLP'09 shared task on event extraction. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task* (pp. 1-9). Association for Computational Linguistics.
32. Hirschman, L., Yeh, A., Blaschke, C., & Valencia, A. (2005). Overview of BioCreAtIvE: critical assessment of information extraction for biology. BMC bioinformatics, 6(Suppl 1), S1.
33. Ng, A. Y., & Jordan, M. I. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. (T. G. Dietterich, S. Becker, & Z. Ghahramani, Eds.)*Advances in neural information processing systems*,*2*(14), 841-848. Nips.
34. Dulac-Arnold, Gabriel, Ludovic Denoyer, and Patrick Gallinari. "Text classification: a sequential reading approach." *Advances in Information Retrieval* (2011): 411-423.
35. Ramage, D., Hall, D., Nallapati, R., & Manning, C. D. (2009, August). Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1* (pp. 248-256). Association for Computational Linguistics.
36. Jin, B., Chen, V., Chen, L., & Lu, X. (2011). Mapping annotations with textual evidence using an scLDA model. In *AMIA Annual Symposium Proceedings* (Vol. 2011, p. 834). American Medical Informatics Association.