Automatic Classification of PubMed Abstracts with Latent Semantic Indexing: Working Notes

Joel Robert Adams and Steven Bedrick

Center for Spoken Language Understanding
Oregon Health and Science University,
3181 SW Sam Jackson Park Road, Portland, OR, USA
{adamjo,bedricks}@ohsu.edu
http://www.ohsu.edu/cslu

Abstract. The 2014 BioASQ challenge 2a tasks participants with assigning semantic tags to biomedical journal abstracts. We present a system that uses Latent Semantic Analysis to identify semantically similar documents in MEDLINE to an unlabeled abstract, and then uses a novel ranking scheme to select a list of MeSH headers from candidates drawn from the most similar documents. Our approach achieved good precision, but suffered in terms of recall. We describe several possible strategies to improve our system's performance.

1 Introduction

Biomedical journal articles are manually indexed in the National Library of Medicine's MEDLINE database with semantic descriptors selected from the Medical Subject Headings (MeSH) hierarchy. These descriptors are then used as key features in traditional document retrieval systems such as PubMed, as well as for document classification and recommendation (c.f. [11, 9, 10]) and even for word-sense disambiguation[7]. This manual indexing process is both time-consuming and expensive[1], and as a result the field of automatic MeSH indexing has a long and rich history(c.f. [16, 2, 13], just to name a few). The goal of BioASQ Task 2a is to automatically assign MeSH index headings to un-tagged MEDLINE abstracts.

Previous researchers have tried a wide variety of approaches to this problem, including discriminative classifiers such as Bayesian classifiers[15] and Support Vector Machines[6, 3] as well as tools based on more traditional natural langauge processing techniques[2]. We approach the problem from a document clustering perspective, based on the observation that similar documents often share MeSH terms. For example two articles about treatments prolonging survival of patients with Glioblastoma, one tagged with 15 MeSH descriptors and the other with 17, share 10 of these terms. This work presents a system that uses Latent Semantic Analysis (LSA)[1] to identify semantically "similar" articles to an

[1] Described in brief in section 2.2; for a more complete description of the technique, see in Furnas, et al.[5].

unlabeled ("query") abstract. Given this set of similar abstracts, we use the human-assigned MeSH descriptors of these similar abstracts to build a set of candidate MeSH descriptors. We then use distributional features of these descriptors to attempt to rank the most likely descriptor candidates for our query abstract.

2 Methods

2.1 Data Selection

Due to the large size of the training data, and the changing nature of the MeSH tree, we chose to focus only on the documents included in the list of 1,993 journals that BioAsq has identified as having "small average annotation periods", and only include descriptors which appear in the 2014 edition of MeSH. As such, we trained on a subset of the provided *Training Set v.2014b*, considering only journal articles from 2005 and later.

For development purposes, we used a 90/10 train/test split. For our BioASQ submissions, we tested on the entire training set.

2.2 Latent Semantic Analysis

Latent Semantic Analysis (LSA) is a technique for analyzing semantic relationships between documents. It is an extension of standard vector-space retrieval[14] that is more robust in the face of synonymy[4]. LSA has been applied to a wide variety of information retrieval tasks, ranging from standard ad-hoc retrieval[5] to cross-language information retrieval[12]. Using LSA, one may perform vector-space retrieval on a low-rank approximation of a term-document matrix, in which "related" words end up grouped together (and are therefore retrieved together). The combination of dimensionality reduction and semantic grouping make LSA a natural fit for the problem of computing document similarity for automatic indexing.

LSA produces this matrix approximation using the singular value decomposition (SVD). The SVD effectively "splits" a term-document matrix X into three new matrices, T, S, and D, which may be multiplied together in order to re-create the original matrix ($X = TSD'$). The S matrix contains the "singular values" of X, and T and D map terms and documents (respectively) onto singluar values. By multiplying more or less complete subsets of the decomposed matrices, one may create more or less accurate approximations of the original matrix.

Given the LSA-produced approximation of the term-document matrix, and a query document, one may perform retrieval as follows. The query document is transformed into a term vector, and this vector is projected into the LSA space. Then, one may use standard vector-space retrieval techniques to score low-rank approximations of corpus documents with the transformed query document.

Our implementation begins by pre-processing MEDLINE abstract using the Python Natural Language Toolkit (NLTK) library.[2] We use NLTK's implementation of the Punkt sentence tokenizer[8] along with the standard NLTK word tokenizer. As part of pre-processing, we removed words found in the standard NLTK English stop word list.

We next used the Gensim library[3] to produce a term-document matrix, in which each "row" represents a term, and each "column" represents a document (i.e., a MEDLINE abstract), and the values in cells represent occurrance counts. We then weighted the counts by their normalized TF/IDF scores, and ran LSA on the resulting matrix. Since the point of LSA is to produce a low-rank approximation of the complete term-document matrix, users of LSA must set an operating point of *how* approximate they wish their new matrix to be. We (somewhat arbitarily) use a the first 200 ranks of our transformed matrix.

2.3 Choosing Closest Neighbors

Once the similarity value is calculated for the new document, its n-closest neighbors are calculated. Based on an initial tuning experiment, a provisional value for n was set at 20. A minimum similarity threshold of .1 was chosen to avoid considering documents with 0 or negative cosine similarity.

The MeSH descriptors associated with these neighbors are the candidates for our new abstract.

2.4 MeSH Descriptor Selection

For our initial submission (Test batch 3, week 4) we developed a simple scoring algorithm to rank the candidate descriptors based on the following assumptions:

1. All else being equal, a MeSH term associated with a *more* similar document should have a greater contribution to the score than a heading from a *less* similar document.
2. Terms which appear *more frequently* in neighboring documents are better candidates than those which only occur a single time.
3. This second point is mediated by the fact that some MeSH headings, such as the check tag "Human" are much more frequent in the corpus than others, so neighbors sharing one of these contributes less information than files sharing a more obscure header.

Let our n neighboring documents $d_1, d_2, \ldots, d_n$ be represented as the ordered pairs $d_i = (s_i, M_i)$ where s_i represents the cosine similarity between document i and the new abstract, and M_i is the set of MeSH terms associated with document i.

[2] `http://www.nltk.org/`
[3] `http://radimrehurek.com/gensim/`

Then for any MeSH header m in our set of candidates, we can define a weighted frequency $f(m)$ as:

$$f(m) = \sum_{i=1}^{n} e(i) \cdot s_i \ . \tag{1}$$

Where:

$$e(i) = \begin{cases} 1 & \text{if } m \in M_i \\ 0 & \text{otherwise .} \end{cases} \tag{2}$$

And define an inverse document frequency $idf(m)$ over the training corpus:

$$idf(m) = \log(\frac{N}{1+C}) \tag{3}$$

Where N is the number of documents in the training corpus and C is the number of documents in the training corpus which contain m.

Then our score for term m is:

$$score(m) = f(m) \cdot idf(m) \tag{4}$$

We then assign a lower threshold of 1.5 and return the highest scored MeSH headers up to a maximum of 12 headers.

3 Results and Discussion

3.1 Flat Measures

Table 1. Flat Measures

Batch	System	Micro-P	Micro-R	Micro-F
3: Wk 4	Baseline	0.2466	**0.2942**	**0.2683**
3: Wk 4	mesh_lsi	**0.2815**	0.2370	0.2573
3: Wk 5	Baseline	0.2315	**0.3088**	**0.2646**
3: Wk 5	mesh_lsi	**0.2688**	0.2423	0.2549

The Micro-Precision score of our system outperforms the BioASQ baseline system – MTI and MTI First Line Index. This suggests that the Latent Semantic Indexing approach is returning semantically relevant MeSH headings.

However, our system consistently performs below baseline on Micro-Recall and, due to this, the Micro-F measure.

This seems consistent with our scoring approach. As an example, let us consider Table 2 which lists the candidates and scores for a document which was manually labelled with the following MeSH descriptors: *'C-Reactive Protein',*

Table 2. Example Candidates and Scores for A Sample Abstract

MeSH Descriptor	Score
C-Reactive Protein	9.008
Biological Markers	5.399
Risk Factors	4.959
Cross-Sectional Studies	4.539
Logistic Models	3.513
Cardiovascular Diseases	3.322
Predictive Value of Tests	3.267
Aged	3.265
Cohort Studies	3.117
Middle Aged	2.942
Ankle Brachial Index	2.814
Female	2.558
Venous Thromboembolism	2.447
Male	2.391
...	...
Humans	1.382
...	...
Renal Dialysis	0.878
...	...
Haplotypes	0.8322
...	...

'Cardiovascular Diseases', 'Female', 'Haplotypes', 'Humans', 'Male', 'Middle Aged', 'Renal Dialysis', 'Risk Factors'.

The horizontal line below the term 'Female' marks the 12 term threshold. Ellipses mark where terms were removed for clarity. Terms in the actual list of headers are marked in bold.

In this particular example, a total of 147 candidate terms were considered. The candidate list includes *all* of the MeSH terms that were manually applied to the abstract. However, our current selection criteria excludes 'Male' due to our choice of assigning a maximum of 12 terms, and further excludes three more potential true positives from consideration because their score is below our chosen threshold of 1.5.

Simply increasing the ceiling on the number of allowed MeSH headers would allow the term 'Male'. However, not without reducing precision. As such, modifications will need to be made to the scoring rule to improve scores for relevant terms like 'Male' and 'Haplotypes' while reducing irrelevant terms like 'Ankle Brachial Index'.

3.2 Hierarchical Measures

In Table 3 you can see our performance in the hierarchical Lowest Common Ancestor measures. Again, our system's precision is competitive with the BioASQ baseline system but recall is lower.

Table 3. Hierarchical Measures

Batch	System	LCA-P	LCA-R	LCA-F
3: Wk 4	Baseline	**0.3271**	**0.3207**	**0.3107**
3: Wk 4	mesh_lsi	0.3230	0.2699	0.2844
3: Wk 5	Baseline	0.3061	**0.3345**	**0.3059**
3: Wk 5	mesh_lsi	**0.3177**	0.2782	0.2874

3.3 Training Performance Figures

For the two submissions that we entered, both training and evaluation were performed on a single 2.9 GHz MacBook Pro with 8GB of memory. Under those conditions, training the LSA model took approximately 6 hours, and once that was complete, the system could generate MeSH headers for approximately 20 abstracts **??**er minute.

This made evaluating changes over the system unwieldy. Subsequently both the training and the assignment of headers have been updated to run on a cluster, but we have yet to evaluate the performance gains.

4 Conclusion and Future Work

The results for the system are encouraging, and suggest that this is a viable approach to semantic tagging. However there are a number of potential avenues for improvement that we will continue to explore.

The scoring and selection of candidates naively seems to be the area where the largest gains could be made, particularly in recall. We'll begin by separating the features of cosine similarity and term frequency in the candidate set, in order to allow for separate weighting of these features.

In addition, we are experimenting with adding a feature to track whether a candidate is a major MeSH term in the relevant training document, as these are should represent the primary concern of a given article.

Another potential source of information is the hierarchical structure of MeSH terms. Once a candidate set is chosen, leveraging the structure of the MeSH tree should help us to reduce cases of over and under-specialization

There is also room for improvement in the LSA model. The list of stopwords should be given some consideration. Numbers without context seem largely irrelevant in this case, and section headers which appear in some but not all PubMed abstracts (such as 'RESULTS' and 'CONCLUSION') should probably be ignored. In addition, we are investigating stemming and normalization of acronyms to improve document matching.

Finally, there are a number of variables that could be tuned. We are investigating the effects of both varying the number of similar documents considered, and replacing *n-closest* with a similarity threshold for documents. Similarly, we are investigating removing the hard-ceiling on number of MeSH terms associated

with an abstract, and instead basing this decision on the distribution of scores among the candidates.

This investigation is still fairly preliminary . We'll continue to refine and document the system going forward.

References

1. Aronson, A.R., Bodenreider, O., Chang, H.F., Humphrey, S.M., Mork, J.G., Nelson, S.J., Rindflesch, T.C., Wilbur, W.J.: The NLM Indexing Initiative. Proceedings / AMIA Annual Symposium AMIA Symposium pp. 17–21 (2000)
2. Aronson, A.R., Lang, F.M.: An overview of MetaMap: historical perspective and recent advances. Journal of the American Medical Informatics Association : JAMIA 17(3), 229–236 (May 2010)
3. Cai, L., Hofmann, T.: Hierarchical Document Categorization with Support Vector Machines. In: Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management. pp. 78–87. ACM, New York, NY, USA (2004)
4. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. JASIS 41(6), 391–407 (1990)
5. Furnas, G., Deerwester, S., Dumais, S., Landauer, T.K., Harshman, R., Streeter, L., Lochbaum, K.: Information retrieval using a singular value decomposition model of latent semantic structure. SIGIR '88: Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval (May 1988)
6. 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. pp. 737–742. ACM, New York, NY, USA (2012)
7. Jimeno-Yepes, A.J., McInnes, B.T., Aronson, A.R.: Exploiting MeSH indexing in MEDLINE to generate a data set for word sense disambiguation. BMC Bioinformatics 12, 223 (2011)
8. Kiss, T., Strunk, J.: Unsupervised Multilingual Sentence Boundary Detection. Computational Linguistics 32(4), 485–525 (Dec 2006)
9. Lin, J., DiCuccio, M., Grigoryan, V., Wilbur, W.: Navigating information spaces: A case study of related article search in PubMed. Information Processing and Management 44(5), 1771–1783 (2008)
10. Lin, J., Wilbur, W.J.: PubMed related articles: a probabilistic topic-based model for content similarity. BMC Bioinformatics 8, 423 (2007)
11. Lin, Y., Li, W., Chen, K., Liu, Y.: A document clustering and ranking system for exploring MEDLINE citations. Journal of the American Medical Informatics Association : JAMIA 14(5), 651–661 (2007)
12. Littman, M.L., Dumais, S.T., Landauer, T.K.: Automatic Cross-Language Information Retrieval Using Latent Semantic Indexing. In: Grefenstette, G. (ed.) Cross-Language Information Retrieval: The Spring International Series on Information Retrieval, pp. 51–62. Springer (1998)
13. Ruch, P.: Automatic assignment of biomedical categories: toward a generic approach. Bioinformatics (Oxford, England) 22(6), 658–664 (Mar 2006)
14. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Communications of the ACM 18(11) (Nov 1975)

15. Sohn, S., Kim, W., Comeau, D.C., Wilbur, W.J.: Optimal training sets for Bayesian prediction of MeSH assignment. Journal of the American Medical Informatics Association : JAMIA 15(4), 546–553 (Jul 2008)
16. Trieschnigg, D., Pezik, P., Lee, V., de Jong, F., Kraaij, W., Rebholz-Schuhmann, D.: MeSH Up: effective MeSH text classification for improved document retrieval. Bioinformatics (Oxford, England) 25(11), 1412–1418 (Jun 2009)