The University of Alberta participation in the BioASQ Challenge: The Wishart system

Yifeng Liu

Department of Computing Science, University of Alberta
Canada
yifeng@cs.ualberta.ca

Abstract. This paper describes the Wishart system that participated in the first BioASQ Challenge on large-scale biomedical semantic indexing and question-answering.

1 BioASQ Task 1a: Large-Scale Online Biomedical Semantic Indexing

PubMed indexes journal articles using the Medical Subject Headings (MeSH) terms. Predicting MeSH terms for new journal abstracts in PubMed is a challenging task, both computationally and methodologically. Computationally speaking, there are 26,000+ MeSH terms to predict and 22+ million abstracts (gigabytes of text) in the training dataset. In one hand, training a single machine learned classifier to predict a total of 26,000+ classes will not be accurate or computationally feasible; in the other hand, training 26,000+ classifiers will be computationally intensive, if not computationally unfeasible. Apart from the intensive computational challenge, the training dataset (manually indexed PubMed abstracts) is imperfect in the sense that 1) not all relevant MeSH terms are referenced with an abstract, and 2) MeSH term relevant to the body of the article but not directly relevant to the abstract could also be referenced by human annotators. To overcome the aforementioned challenge, we employ a hybrid approach, which takes into account both surface terms in a given abstract (N-gram classification), as well as similar abstracts that are well annotated in PubMed (K-Nearest Neighbor predictions). The final submission is an ensemble classification of results from both predictors.

1.1 N-gram Classification

Using the supplied training data, we trained 26,000+ binary Support Vector Machine classifiers, each of which predicts whether or not an abstract should be referenced with a certain MeSH label. The training dataset for a particular MeSH term is constructed using N-grams (unigram, bigram, trigram) found in abstracts referencing this particular MeSH term. In cases where there are too many referencing abstracts (e.g. D000328 Adult, D000818 Animals, etc.) we sample the training data according to our computational capacity. Given a

testing abstract, we generate the same type of N-gram classification features as used in the training phase, and predict whether this abstract is referencing one or more MeSH terms corresponding to a total of 26,000+ classifiers. That is, to predict the set of MeSH terms referencing by a single testing abstract, we need to perform 26,000+ classifications, each classification predicts either a yes or a "no" for a particular MeSH label. We exploit feature selection to improve classification accuracy as well prediction speed. Training 26,000+ classifiers are computationally intensive and we used a computer cluster to speed up the training process from a few CPU years to about two wall-clock weeks. Once trained, classifiers can be applied relatively efficiently to classify new test abstracts. Typically speaking, the classification process takes a couple hours on a single desktop computer.

1.2 K-Nearest Neighbor Prediction

Besides using surface terms found in an abstract, similar documents calculated by PubMed can also assist annotation of unlabeled testing abstracts. Given a testing abstract, we query NCBI Entrez for a list of similar articles and examine their referencing MeSH terms. To reduce the amount of queries sent to NCBI Entrez, we utilize a local index of Medline articles (built in-house) for efficient PubMed ID to MeSH term retrieval. We rank the list of top N similar abstracts and predict the top M most frequent MeSH terms. There are two parameters we need to optimize here: N, the number of similar documents to examine, and M, the number of most frequent MeSH terms to predict. From empirical statistics, we found that documents ranked lower than 100 have few common MeSH terms with training instances, and so we empirically set N=100. When it comes to optimizing M, the number of labels to predict, we found that this approach is great for recalls / label coverage. Predicting the 10 most frequent labels found in similar abstracts covers more than half true labels, and predicting 20 most frequent labels covers about 75% true labels. Increasing the number of labels to 30+ and we have almost 95% coverage of true labels. The problem of increasing the number of predicting labels, however, is the amount of false positives that it introduces: as we increase the number of labels from M=10 to M=30, we introduce almost 3 folds of false positive labels. From meta statistics of PubMed, we know that each abstract is referencing 10$\sim$15 MeSH terms and so to reduce the amount of false positives, we set the number of labels to predict empirically to M=10. This approach is extremely quick as it only involves sending a few queries to NCBI Entrez and a few thousands search queries to our local document index. A run of the KNN predictor typically takes less than 5 minutes to complete.

1.3 Ensemble Classifications

Both N-gram classifications and K-Nearest Neighbors (KNN) predictions are independent as they use totally different feature sources and so their predictions are orthogonal to each other. Common MeSH labels predicted by both approaches are highly accurate, and they are included in the final prediction.

For the remaining labels where two classifiers diverge, we rank them based on an empirical scoring function combining both the confidence score of the support vector machine classifier, and the frequency score of the K-nearest neighbor predictor. We bias towards the KNN predictor empirically to improve prediction accuracy. To produce the final prediction, we add first the common labels and then adjust the ranking score cut-off such that on average each abstract in the testing dataset are referencing to around 10 15 MeSH terms.

2 BioASQ Task 1b: Introductory Biomedical Semantic Question Answering

This task is composed of two phases. In phase A, we need to retrieve relevant annotations (concepts, articles, snippets, and RDF triplets) to a given question; in phase B, we need to produce an exact and ideal answer from a gold set of relevant concepts. To tackle phase A, we use a mix of query processing, query expansion, and document ranking techniques. To tackle phase B, we use mainly concept-assisted summarizations.

2.1 Phase A

This phase is made easy as we only need to send search queries to a shared document index provided by BioASQ, instead of making our own indices for concepts and documents. The challenge here is that our search query and returned results may not be perfect, so our focus is on query processing and result ranking. Converting a natural language question to a sophisticated search query is a non-trivial task. To accomplish query conversion, we extract noun phrases from a question, and then reference/disambiguate them using our in-house thesauri of biomedical entities. Once referenced, we expand the search query by including synonyms and relevant biomedical entities (genes, proteins, drugs, diseases, pathways, metabolites, etc.) found using PolySearch [1][1]. Relevant biomedical entities may not be mentioned explicitly in the question body, but they are important in retrieving further relevant concepts, documents and snippets. We query PolySearch for a list of relevant biomedical entities, and use the set of relevant biomedical entities to help ranking the retrieved set of concepts, articles, snippets, and RDF triplets. Basically we favor retrieved concepts that coincide with our set of relevant biomedical entities. Articles containing more relevant biomedical entities are ranked higher, and that sentences containing one or more search query terms, along with a subset of relevant biomedical entities are extracted as relevant snippets.

2.2 Phase B

In this phase we assume the set of given concepts, articles, snippets, and RDF triplets are perfect. We focus first on producing an ideal answer from the given

[1] http://wishart.biology.ualberta.ca/polysearch/

annotations. Similar to our approach to Phase A, we retrieve a set of relevant biomedical entities to our query terms using PolySearch, and use this set to augment the given gold concepts. Sentences in the retrieved articles and snippets are extracted and ranked according to their cosine similarity to the set of gold concepts. Finally we stitch the top ranked sentences to produce a summary as an ideal answer, and from the ideal answer, we generate exact answers according to different question types. In examining the result produced using this approach, we found that there are a few cases where result could be greatly improved using reasoning. However, due to the limit amount of training data and time we could afford to spend on solving this task, we did not exploit reasoning for this phase. It will be our future work to exploit the power of reasoning and ontology in producing better answers to the given set of questions and gold annotations.

References

1. Dean Cheng, Craig Knox, Nelson Young, Paul Stothard, Sambasivarao Damaraju, and David S. Wishart. Polysearch: a web-based text mining system for extracting relationships between human diseases, genes, mutations, drugs and metabolites. *Nucleic Acids Research*, 36(Web-Server-Issue):399–405, 2008.