Two hierarchical text categorization approaches for BioASQ semantic indexing challenge

Francisco J. Ribadas[1], Luis M. de Campos[2],
Víctor M. Darriba[1], Alfonso E. Romero[3]

[1] Departamento de Informática, Universidade de Vigo
E.S. Enxeñería Informática, Edificio Politécnico,
Campus As Lagoas, s/n, 32004 Ourense (Spain)
{ribadas,darriba}@uvigo.es
[2] Departamento de Ciencias de la Computación e Inteligencia Artificial
Universidad de Granada
E.T.S.I. Informática y de Telecomunicación,
Daniel Saucedo Aranda, s/n, 18071 Granada (Spain)
lci@decsai.ugr.es
[3] Centre for Systems and Synthetic Biology, and Department of Computer Science,
Royal Holloway, University of London Egham, TW20 0EX, United Kingdom
aeromero@cs.rhul.ac.uk

Abstract. This paper describes our participation in the BioASQ semantic indexing challenge with two hierarchical text categorization systems. Both systems originated from previous research in thesaurus topic assignment applied on small domains from the legal document management field. One of the described systems employs a classical top-down approach based on a collection of local classifiers. The other system builds a Bayesian network induced by the thesaurus structure and contents, taking into account descriptor labels and related terms. We describe the adaptations required to deal with a large thesaurus like MeSH and a huge document collection and discuss the results obtained in the BioASQ challenge and the limitations of both approaches.

1 Introduction

Text classification on hierarchies is a research field that has had limited presence at both machine learning and natural language processing fields, although it has recently started to gain greater attention. This rise is mainly due to the increasing amount of available on-line resources involving large conceptual taxonomies such as web directories or huge document collections like MEDLINE [4], EUR-Lex [5] or even Wikipedia. This resource availability makes hierarchical text categorization a promising research field in which to experiment and combine many different

[4] A bibliographic database of life sciences and biomedical information whose contents are indexed using Medical Subject Headings (MeSH) thesaurus (see http://www.ncbi.nlm.nih.gov/pubmed).

[5] A service providing legal texts of the European Union that employs EUROVOC multilingual thesaurus in indexing and searching tasks (see http://eur-lex.europa.eu).

approaches proposed by researchers in machine learning and natural language processing. Proof of this interest are the recent Large Scale Hierarchical Text Classification (LSHTC) challenges [8] which have offered an environment to evaluate both performance and efficiency issues of new text categorization methods on real world collections.

In this context, the BioASQ challenge goes a step further by offering a huge real world environment in a complex domain, biomedical document management, which is currently experiencing a boom and where automatic text understanding tools are becoming a need. Of the two main areas in BioASQ challenge, semantic indexing and question answering (QA), our work falls into the first one. Our research groups have previous experience in small and medium scale automatic text indexing using medium size thesauri in the legal domain, employing two different methods to accomplish the hierarchical categorization task. Our intention in this participation in BioASQ is to check the suitability of our previous approaches in a larger domain, with a much more complex terminology and with strict time and processing restrictions.

The rest of the paper is organized as follows. Section 2 briefly describes the two hierarchical classification schemes that we have employed in our BioASQ challenge participation. Section 3 gives details about the preprocessing and adaptations made in both approaches to make them able to deal with the training dataset and the requirements of the BioASQ semantic indexing task. Section 4 discusses some experimental results with different parametrization of our categorization tools, and finally, we detail the more relevant conclusions of our participation in the challenge.

2 Our hierarchical text categorization approaches

In our participation in the BioASQ challenge we have employed two systems developed by two different research groups. Both of them model the thesaurus descriptor assignment task as a hierarchical categorization problem, and the two categorization tools were result of independent previous research on automatic indexing using thesaurus structures.

In both cases the original domains were very different from those considered in the BioASQ challenge, parliamentary resolutions in one case and public grants and subsidies on the other. Additionally, in the initial versions of these tools both the size and the complexity of the thesaurus being employed were significantly smaller than in the case of the MeSH thesaurus. In the parliamentary documents case, the multilingual thesaurus EUROVOC was in use as indexing base, with less than 4000 descriptors, whereas in the subsides and grants publications collection a custom thesaurus with about 1800 descriptors was employed.

2.1 The HACE approach: top-down hierarchy of local classifiers

HACE (Hierarchical Annotation and Categorization Engine) is a generic framework for hierarchical categorization that evolved from previous work on text

categorization on legislative document domain [3]. It is proposed as a framework for experimenting with various configurations of hierarchical classifiers following the classic top-down scheme described as *Local Classifier Per Node Approach* in the taxonomy of hierarchical classification approaches presented by Silla and Freitas in [2] and traces its origins to the work of Koller and Sahami [5].

Roughly speaking, this approach builds a local binary classifier for each node in the hierarchy of classes, except for the root node, which will be responsible for determining the pertinence of assigning that class or one of its descendants as a label for each input example being classified. HACE allows both tree-shaped hierarchies and taxonomies structured as DAG (directed acyclic graph). In the second case, it will create as many local models as hierarchical contexts the node may appear, that is, the framework will build a local model for every different parent a node can have in the considered DAG, what we call a context. The HACE framework aims to provide a modular collection of components to build and train the local classifiers associated with each node in the class taxonomy, covering the following aspects:

- strategy for building/selecting positive examples set with a bottom-up procedure
- strategy for building/selecting negative examples set
- feature selection method used at each local model: employing conventional feature selection (Information Gain, Chi Squared, etc) or features extracted from thesaurus labels
- classification algorithm being used to perform the "routing" decisions at each local model
- strategies for handling unbalanced classes: reweighting, selecting boundary negative examples, distribution of negative examples in an ensemble of classifiers

Additionally, HACE offers features specifically designed for classification tasks in large textual data collections. In particular, textual repositories are backed by an Apache Lucene [6] textual index with three fields storing document ID, categories list and full text. This index helps in computing feature vectors during local model training and in other complementary tasks like searching for similar documents. In the case of large hierarchies or problems with large amounts of training examples an incremental bottom-up scheme for positive example selection can be employed. This approach helps to mitigate performance problems when building local models in higher classes in the topology when a "less exclusive" policy, as defined in Eisner et al work [6], is employed. This positive example selection policy considers as positive example every example labeled with any descendant of the current class. This behaviour can lead to the accumulation of huge and unmanageable training sets when dealing with local models at the top of the taxonomy. The current version of HACE supports two bottom-up positive example selection methods: a simple random selection with a fixed amount of

[6] http://lucene.apache.org

examples per local model and a k-means clustering based approach, where examples closer to the identified centroids are selected as positive examples useful to represent the current class and its descendants in further local model building in higher levels of the taxonomy.

The HACE framework also allows the use of a local classifier per node approach using a sort of "contextual" classifier following an approach inspired by [7] that complements content based routing decisions with bottom-up contextual information coming from node descendants, and, optionally, from node siblings. The intuition behind this idea of exploiting contextual information is to try to reduce false negatives in classifications based exclusively on content, adding information about content based routing decisions performed by descendant nodes on current example.

Thus, after the training phase, each node/context in the taxonomy of classes will have an associated local model characterized by a list of positive examples that provide a representation of the concepts linked to the corresponding class, a list of features selected as relevant to make the local routing decisions and the content based classifier that exploits these features. Optionally, these local models may include a classifier/router based on context, that uses as metafeatures content based decisions made by surrounding local models. During classification of new examples, the set of local models is consulted using a pachinko-like approach to determine in a top-down fashion the list of potential classes that will be employed to label those unlabeled examples. This pachinko-like approach starts at the taxonomy root and consults every direct descendant node model to determine the next branch, or set of branches, where this top-down procedure will be repeated until a leaf node is reached or all of the descendants of a internal node decide to discard the current example. Those nodes where this top-down search stops are included in the final list of assigned labels for the current example.

An additional feature available during classification phase and useful for text classification tasks in large hierarchies is the ability to perform a guided top-down search with a pre-filtering step. This pre-filtering step exploits the set of descriptors linked to the most similar documents retrieved from the Lucene index that backs feature vector building. For a given document to be labeled, the Lucene index is queried using the document text contents to retrieve the top most similar documents with their respective categories. These sets of categories are employed to create with them a weighted ranking of potential labels in a similar way as is described in [9]. The idea is to start the top-down search process in the neighbourhood of those labels (typically with their grandparents) instead of in the taxonomy root. This optimization helps to avoid the negative effect of potential errors (false negatives) commited by local models in the higher levels of the taxonomy which will result in a premature discard of useful paths.

2.2 The Rebayct approach: Bayesian network induced from taxonomy

Rebayct is a software tool for document classification using descriptors extracted from a thesaurus, based on Bayesian networks.

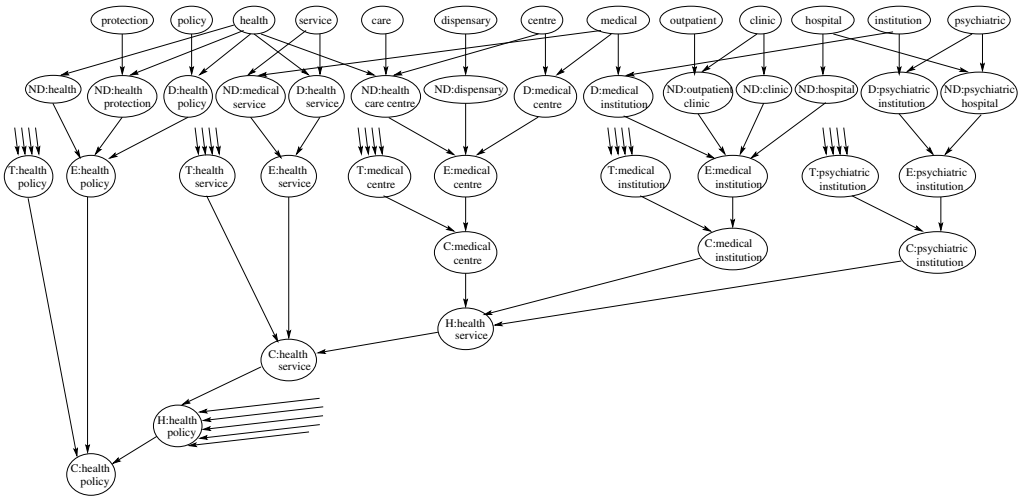

Fig. 1. Bayesian network in the example about *health*

Rebayct creates a Bayesian network to model the hierarchical and equivalence relationships in the thesaurus and extends it to incorporate training data. Then, given a document to be classified, its terms are instantiated in the network and a probabilistic inference algorithm, specifically designed and particularly efficient, computes the posterior probabilities of the descriptors in the thesaurus.

Our model of a thesaurus through a Bayesian network is based on two key ideas: (1) to explicitly distinguish between a concept and the descriptor label and non-descriptor labels used to represent it and (2) to clearly separate, through the use of additional nodes, the different information sources (hierarchy and equivalence relationships, and training data) influencing a concept.

Therefore, according to the first idea, each concept, labeled identically as the descriptor representing it, will be a node C in the network. We shall distinguish between basic and complex concepts: the former do not contain other concepts, whereas the later are composed of other concepts (either basic or complex). Each descriptor and each non-descriptor [7] in the thesaurus will also be nodes D and ND in the network. All the words or terms appearing in either a descriptor label, a non-descriptor label or a training document will be term nodes T. To accomplish with the second key idea, for each concept node C we shall also create three (virtual) nodes: E_C, which will receive the information provided by the equivalence relationships involving C; H_C, which will collect the hierarchical information, i.e. the influence of the concepts contained in C; and T_C, which will concentrate the information obtained for this concept from the training documents.

[7] Usually a synonym or a lexical variation of the descriptor. In the XML version of MeSH thesaurus are linked to the descriptor using *TermList* elements.

With respect to the links, there is an arc from each term node to each descriptor and/or non-descriptor node containing it, as well as from each term node to the virtual training node T_C if the term appears in training documents which are associated with the concept C (these arcs represent the training information). There are also arcs from each non-descriptor node, associated to a concept node C, to the corresponding virtual node E_C (these arcs correspond with the USE relationships in the thesaurus), as well as from the own descriptor node associated with the concept C to E_C. There is also an arc from each concept node C' to the virtual node(s) H_C associated with the broader complex concept(s) C containing C' (these arcs correspond with the BT (Broader Term) relationships in the thesaurus). Finally, there are arcs from the virtual nodes E_C, H_C and T_C to its associated concept node C, representing that the relevance of a given concept will directly depend on the information provided by the equivalence (E_C node) and the hierarchical (H_C node) relationships, together with the training information (T_C node).

For example consider a fragment of the EUROVOC thesaurus composed of two complex descriptors, *health service* and *health policy*, and three basic descriptors, *medical centre*, *medical institution* and *psychiatric institution*. *Health service* is the broader term of *medical centre*, *medical institution* and *psychiatric institution*; *health policy* is in turn the broader term of *health service* (and also of other five descriptors which are not considered). The associated non-descriptors are: *medical service* for *health service*; *health* and *health protection* for *health policy*; *dispensary* and *health care centre* for *medical centre*; *clinic*, *hospital* and *outpatients' clinic* for *medical institution*; and *psychiatric hospital* for *psychiatric institution*. The network corresponding to this example is displayed in Figure 1

The conditional probabilities for the nodes in the network are defined by using several canonical models (additive and an or-gate model) which allow us to perform exact inference efficiently (see [1] for details).

3 Preprocessing for BioASQ

Training data in BioASQ challenge on Large-Scale On-line Biomedical Semantic Indexing consisted in about 11 million annotated articles from MEDLINE collection. Each training document was manually labeled with a set of descriptors taken from the Medical Subject Headings (MeSH) thesaurus.

Although BioASQ organizers also included a concept hierarchy extracted from MeSH thesaurus, in our experiments we have employed the XML version of MeSH 2013 edition to create our own concept taxonomy with a DAG structure. MeSH thesaurus consists of 26,853 descriptors arranged in 16 thematic taxonomies. The hierarchical relationships between descriptors are coded in *TreeNumber* elements. Each MeSH descriptor has one or more *TreeNumbers* describing the places it occupies inside the 16 concept taxonomies. We have exploited these *TreeNumbers* to create our class taxonomy, obtaining a DAG with 26,702 nodes, after the exclusion of 151 descriptors from subhierarchy *"[V] Publication Characteristics"* which are not actually used as labels, and 36,647

doc. selection	max docs	EBP	EBR	EBF	MaP	MaR	MaF	MiP	MiR	MiF
random	500	0.297	0.321	0.309	0.263	0.292	0.272	0.289	0.307	0.298
random	1000	0.336	0.370	0.359	0.294	0.317	0.308	0.327	0.351	0.339
random	2000	0.401	0.437	0.425	0.351	0.390	0.362	0.381	0.411	0.396
k-means	500	0.321	0.343	0.337	0.278	0.309	0.281	0.302	0.319	0.310
k-means	1000	0.364	0.398	0.389	0.316	0.345	0.321	0.331	0.361	0.345
k-means	2000	0.404	0.449	0.436	0.338	0.381	0.356	0.384	0.420	0.402

Table 1. Bottom-up positive document selection experiments.

parent-child links. In our taxonomy extraction process we have only found two direct cycles [8], that where discarded in the final taxonomy. We also extracted the list of related terms for each descriptor in the MeSH thesaurus, usually synonyms or lexical variants, giving a total of 108,117 distinct terms.

Regarding the preprocessing performed on the training and validation documents, we have only employed elementary text processing operations: stopword removal and stemming with the default English stemmer from the Snowball project[9]. Additionally we have processed the resulting tokens to create sets of word bigrams for each document. This way we have built an alternative collection with bigram versions of the original documents and also the word bigrams for descriptor labels and related terms.

The HACE framework was developed from scratch with a modular architecture and with clear guidance to work in large textual collections and incorporate components and adaptations to allow an effective construction of local models in large environments, such as the training set of Task1 of BioASQ challenge. However, Rebayct software was designed with a very specific domain in mind and all its processing is done against memory resident data structures. The size and complexity of the MeSH thesaurus and the huge amount of different tokens in the biomedical training corpus employed in the BioASQ challenge makes it unfeasible to apply the Rebayct approach on the full training set. Therefore, it was necessary to perform a previous selection phase by extracting a reduced training set of 1,242,670 documents, approximately 10 % of available documents. This process employed a Lucene index constructed from the whole collection. For every descriptor ID a Lucene query was launched selecting the top 50 documents for each descriptor not previously included in the list of selected documents. These top documents use to have quite few assigned descriptors and potentially are good samples of the kind of documents linked to the considered descriptor.

Additionally, these 1,242,670 documents were split into five groups of 248.534 training instances, each of these five datasets was employed to train a Rebayct model. In the annotation phase every Rebayct model was applied to the unlabeled test documents and the resulting label lists were combined in a similar way as is done in ensemble methods using bagging approaches.

[8] Descriptor D009014 (*Morals*) with descriptor D004989 (*Ethics*) and descriptor D006885 (*Hydroxybutyrates*) with descriptor D020155 (*3-Hydroxybutyric Acid*).

[9] http://snowball.tartarus.org

	EBP	EBR	EBF	MaP	MaR	MaF	MiP	MiR	MiF
guided search	0.435	0.458	0.451	0.394	0.421	0.409	0.403	0.487	0.445
bigram features	0.397	0.429	0.413	0.346	0.405	0.369	0.390	0.432	0.411

Table 2. Guided top-down search and word bigram features results.

4 Experimental results

For evaluation and parameter tuning, a custom evaluation dataset was built randomly selecting a set of 2,000 documents from the training set provided by the BioASQ challenge organizers. This evaluation set covers 6,413 different MeSH descriptors. Also an arbitrary limit of 15 descriptors was employed to restrict the maximum size of the list of labels assigned to each evaluation document by our tools. This limit was set from the average number of assigned descriptors in the BioASQ training dataset, that according to the BioASQ team is 12,55 MeSH descriptors per article.

As classification performance measures we have employed a set of flat measures similar to the one employed by the BioASQ challenge, using MULAN [10] multilabel learning framework to compute them. The considered measures are the following: Example Based Precision (EBP), Example Based Recall (EBR), Example Based F-Measure (EBF), Macro Precision (MaP), Macro Recall (MaR), Macro F-Measure (MaF), Micro Precision (MiP), Micro Recall (MiR) and Micro F-Measure (MiF).

4.1 HACE experiments

Several aspects of hierarchical categorization can be tuned in the HACE framework. After a preliminary tuning phase using a fragment of MeSH subhierarchy *"[C] Diseases"* and a reduced set of training documents, we decided to employ as local classifier for node models the Support Vector Machines implementation available in Weka [4] plug-in for LibSVM library [11]. We also employ a fairly aggressive feature selection procedure based on Information Gain (IG), selecting the top 100 features with best IG values.

We concentrate our experiments on evaluating the effectiveness of bottom-up positive examples selection, using two strategies: a simple random document selection among descendant nodes selected documents and a k-means based document selection. In both cases we evaluated this positive document selection with a maximum number of 500, 1000 and 2000 instances for each node. Table 1 summarizes the obtained results. Results confirm the intuition that using more documents per node model increases the overall performance. Using the k-means based bottom-up instance selection obtains a small improvement in the performance. The random documents selection gives slightly lower values, but they are obtained with much less computational effort, since it does not require the processing of documents to extract feature vectors neither distance computations needed by k-means clustering.

	EBP	EBR	EBF	MaP	MaR	MaF	MiP	MiR	MiF
single model	0.240	0.345	0.270	0.240	0.317	0.273	0.241	0.319	0.275
agregated 5 models	0.264	0.376	0.296	0.264	0.349	0.301	0.268	0.355	0.306
single model with bigrams	0.281	0.399	0.315	0.281	0.372	0.320	0.281	0.372	0.321

Table 3. Rebayct experiment results.

We also have evaluated the effect of using a guided top-down classification based on a previously selected list of candidate descriptors using a similarity query against the Lucene index used to support the feature vector computation. For any given test document the 10 most similar documents in the index are selected and a similarity weighted ranking is done with the assigned descriptors. From this descriptor list the top 20 are selected to start a top-down search along the taxonomy local models starting at their grandparents. An additional experiment was performed comparing single token features against word bigram based features. The results obtained in these two situations are shown in table 2, where a random document selection approach with a limit of 2000 instances was used. The guided top-down search improves the performance and reduces the overall computational cost. On the other hand the performance gain due to using word bigram features is not very relevant.

4.2 Rebayct experiments

Rebayct customization capabilities are more restricted. In table 3 we show the results obtained with three configurations: using only one of the trained models, aggregating the results of 5 models built with different training subsets and, finally, one single model using word bigrams as instance features and in descriptor labels and related terms. The best results are obtained when bigrams are employed. This seems reasonable since biomedical documents tend to employ complex terminologies with long nominal phrases and many named entities that word bigrams are able to partially cover.

4.3 Official BioASQ results

As an illustration of the performance of our systems at BioASQ challenge, Table 4 shown the results obtained by our runs in test set batch number 3. In this batch we have participated with the following five configurations, using both single word tokens and word bigrams.

hace1. HACE framework using k-means bottom-up positive example selection with up to 2000 examples per node, Information Gain feature selection with up to 100 features per node and a SVM classifier as content based router for each model.

hace2. Same configuration as **hace1** using the guided top-down search approach described in section 2.2.

hace2-ne. Same configuration as **hace2** using word bigrams as textual features.
rebayct. Combination of five Rebayct models trained with five splits of the 1,242,670 documents in the reduced training set described in section 3.
rebayct2. A Rebayct model trained on one of these 248.534 documents split using word bigrams as document features and also in descriptor labels and related term labels (non-descriptors).

Table 4 shows the official measures for the best system at each one of the six runs in batch number 3 as well as the performance measures for our five systems. Results were taken from the BioASQ online system [10] which ranks the participating systems performance based on two measures: one hierarchical, Lowest Common Ancestor F-measure (LCA-F), and one flat measure, Label-based micro F-measure (MiF). For each one of these measures the ranking position of our systems are included to give an approximated idea of the overall performance of our systems in comparison with other BioASQ participants. Obtained results are in accordance with the results described in sections 4.1 and 4.2 and confirm that our systems are not among the most competitive systems in the BioASQ challenge.

5 Conclusions

We have taken part in the BioASQ biomedical semantic indexing challenge with two different hierarchical text categorization systems, a hierarchy of local classifiers and an induced Bayesian network. As shown in the previous section the performance of our two systems in BioASQ challenge was not very good.

In the case of Rebayct system we have some limitations that make it unsuitable for a huge domain like the one we are working with in BioASQ challenge. The Rebayct approach is able to manage both hierarchical information taken from the thesaurus links and information extracted from the training instances. In our experiments we have confirmed that in the case of BioASQ challenge the more relevant element is the training data, mainly due to the large amount of available instances. In other domains with a lack in training data the Rebayct ability to label documents with small or no training would make this tool more attractive.

The HACE framework was designed to deal with large categorization problems. There are many components and parameters to configure and a more deep parameter tuning could improve the reported results. In preliminary experiments with smaller document collections we have evaluated several strategies to deal with unbalanced categorization in local classifiers obtaining some improvements in overall categorization performance. Another important line of research which can lead to improvements in categorization quality in complex domains like biomedical semantic indexing is related with the collection preprocessing using natural language processing approaches more sophisticated than simple stemming and stop-word removal, like domain specific lemmatization or

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

named entities recognition. With our HACE framework participation in BioASQ challenge we have also confirmed the relevance of working with large training datasets, since the best results were obtained using the guided top-down search, which starts with a first step that is essentially a kind of k nearest neighbours assisted by a Lucene index.

Acknowledgements

Research reported in this paper has been partially funded by "Ministerio de Economía y Competitividad" and FEDER under the project TIN2010-18552-C03-01, by "Xunta de Galicia" under the projects CN 2012/319 and CN 2012/317 and by "Consejera de Innovación, Ciencia y Empresa de la Junta de Andalucía" under the project P09-TIC-4526.

References

1. L.M. de Campos, A.E. Romero. Bayesian network models for hierarchical text classification from a thesaurus. International Journal of Approximate Reasoning 50(7):932-944, 2009.
2. C. N. Silla Jr., A. A. Freitas. A survey of hierarchical classification across different application domains. Data Mining and Knowledge Discovery. Vol. 22, No. 1-2, pp. 31-72, 2011.
3. Francisco J. Ribadas, Erica Lloves, Victor M. Darriba. Thesaurus topic assignment using hierarchical text categorization, Proc. of ACM SIGIR 2007 Workshop on Improving Non-English Web Searching (iNEWS07), pp. 65-68, Amsterdam, The Netherlands, 2007.
4. I. Witten and E. Frank Data Mining: Practical machine learning tools and techniques, 2nd Ed. *Morgan Kaufmann*, San Francisco, 2005.
5. D. Koller and M. Sahami. Hierarchically classifying documents using very few words. Proc. of 14th Int. Conf. on Machine Learning, pp. 170–178, Nashville, US, 1997
6. Eisner R, Poulin B, Szafron D, Lu P, Greiner R. Improving protein function prediction using the hierarchical structure of the gene ontology. Proc. of the IEEE Symp. on Computational Intelligence in Bioinformatics and Computational Biology, pp 110, 2005
7. P.N. Bennett and N. Nguyen. Refined experts: improving classification in large taxonomies. Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval. 1118, 2009
8. LSHTC Challenge: The Pascal Challenge on Large Scale Hierarchical Text classification. http://lshtc.iit.demokritos.gr/
9. D Trieschnigg, P Pezik, V Lee, F De Jong, W Kraaij, D Rebholz-Schuhmann. MeSH Up: effective MeSH text classification for improved document retrieval. Bioinformatics 25 (11), 1412-1418, 2009.
10. G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek, I. Vlahavas. Mulan: A Java Library for Multi-Label Learning, Journal of Machine Learning Research, 12, pp. 2411-2414. 2012.
11. C.-C. Chang and C.-J. Lin. LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1–27:27, 2011

week 1, labeled documents: 1947/7650

system	flat rank	MiF	EBP	EBR	EBF	MaP	MaR	MaF	MiP	MiR	Acc.	hier. rank	LCA-F	HiP	HiR	HiF	LCA-P	LCA-R
best	1/24	0.572	0.561	0.595	0.560	0.585	0.457	0.441	0.570	0.575	0.404	1/24	0.483	0.725	0.727	0.705	0.498	0.499
hace2	16/24	0.403	0.382	0.461	0.398	0.285	0.339	0.305	0.382	0.425	0.258	15/24	0.383	0.588	0.644	0.587	0.382	0.419
rebayct2	18/24	0.337	0.320	0.382	0.332	0.549	0.186	0.186	0.320	0.356	0.206	19/24	0.320	0.590	0.565	0.550	0.318	0.356
rebayct	21/24	0.295	0.280	0.332	0.290	0.517	0.153	0.153	0.280	0.312	0.175	21/24	0.288	0.549	0.500	0.497	0.283	0.325

week 2, labeled documents: 2674/10233

system	flat rank	MiF	EBP	EBR	EBF	MaP	MaR	MaF	MiP	MiR	Acc.	hier. rank	LCA-F	HiP	HiR	HiF	LCA-P	LCA-R
best	1/24	0.578	0.577	0.589	0.566	0.598	0.435	0.427	0.585	0.572	0.410	1/24	0.486	0.735	0.711	0.702	0.507	0.496
hace2	13/24	0.488	0.473	0.533	0.481	0.436	0.363	0.346	0.473	0.505	0.330	11/24	0.433	0.660	0.672	0.641	0.444	0.456
hace1	18/24	0.415	0.402	0.459	0.411	0.294	0.324	0.297	0.402	0.429	0.268	17/24	0.388	0.599	0.629	0.588	0.393	0.416
rebayct	21/24	0.302	0.293	0.330	0.297	0.545	0.145	0.149	0.293	0.312	0.181	22/24	0.291	0.557	0.489	0.497	0.289	0.320

week 3, labeled documents: 2001/8861

system	flat rank	MiF	EBP	EBR	EBF	MaP	MaR	MaF	MiP	MiR	Acc.	hier. rank	LCA-F	HiP	HiR	HiF	LCA-P	LCA-R
best	1/27	0.575	0.567	0.596	0.565	0.580	0.454	0.437	0.572	0.579	0.408	1/27	0.486	0.723	0.719	0.700	0.502	0.501
hace2	13/27	0.476	0.450	0.535	0.469	0.417	0.385	0.358	0.450	0.506	0.318	13/27	0.426	0.636	0.672	0.627	0.430	0.456
hace2-ne	14/27	0.474	0.448	0.533	0.467	0.411	0.376	0.350	0.448	0.504	0.317	14/27	0.425	0.633	0.670	0.625	0.429	0.456
hace1	18/27	0.409	0.386	0.467	0.405	0.290	0.339	0.305	0.386	0.434	0.263	17/27	0.386	0.578	0.627	0.576	0.385	0.419
rebayct2	19/27	0.349	0.330	0.396	0.344	0.407	0.251	0.236	0.330	0.371	0.214	19/27	0.343	0.567	0.599	0.557	0.341	0.375
rebayct	24/27	0.301	0.284	0.339	0.295	0.518	0.155	0.155	0.284	0.319	0.179	25/27	0.292	0.541	0.492	0.490	0.287	0.325

week 4, labeled documents: 972/1986

system	flat rank	MiF	EBP	EBR	EBF	MaP	MaR	MaF	MiP	MiR	Acc.	hier. rank	LCA-F	HiP	HiR	HiF	LCA-P	LCA-R
best	1/29	0.563	0.539	0.593	0.547	0.531	0.461	0.443	0.547	0.581	0.394	1/29	0.473	0.696	0.721	0.686	0.480	0.497
hace2	18/29	0.470	0.412	0.569	0.459	0.368	0.416	0.373	0.412	0.549	0.312	15/29	0.421	0.592	0.714	0.623	0.400	0.482
hace2-ne	19/29	0.466	0.408	0.562	0.454	0.366	0.412	0.371	0.408	0.544	0.309	17/29	0.418	0.588	0.709	0.619	0.395	0.479
hace1	20/29	0.398	0.348	0.490	0.391	0.257	0.362	0.319	0.348	0.464	0.253	20/29	0.378	0.537	0.669	0.572	0.352	0.442
rebayct2	21/29	0.330	0.289	0.409	0.325	0.336	0.286	0.263	0.289	0.386	0.201	22/29	0.329	0.503	0.632	0.536	0.304	0.391
rebayct	25/29	0.287	0.251	0.346	0.279	0.462	0.189	0.181	0.251	0.335	0.169	27/29	0.279	0.488	0.526	0.484	0.254	0.339

week 5, labeled documents: 732/1750

system	flat rank	MiF	EBP	EBR	EBF	MaP	MaR	MaF	MiP	MiR	Acc.	hier. rank	LCA-F	HiP	HiR	HiF	LCA-P	LCA-R
best	1/28	0.567	0.551	0.589	0.550	0.532	0.457	0.439	0.553	0.581	0.396	1/28	0.476	0.704	0.707	0.683	0.491	0.491
hace2	17/28	0.473	0.416	0.563	0.461	0.366	0.420	0.382	0.416	0.549	0.314	15/28	0.429	0.599	0.712	0.629	0.409	0.481
hace2-ne	18/28	0.471	0.413	0.556	0.457	0.365	0.414	0.375	0.413	0.546	0.312	18/28	0.425	0.595	0.705	0.623	0.405	0.479
hace1	19/28	0.410	0.360	0.499	0.403	0.260	0.371	0.331	0.360	0.475	0.263	19/28	0.390	0.549	0.675	0.584	0.366	0.447
rebayct2	20/28	0.339	0.298	0.410	0.332	0.348	0.299	0.277	0.298	0.394	0.206	20/28	0.338	0.523	0.622	0.545	0.315	0.394
rebayct	23/28	0.290	0.255	0.348	0.283	0.446	0.197	0.187	0.255	0.337	0.171	26/28	0.284	0.492	0.510	0.480	0.259	0.339

week 5, labeled documents: 305/1357

system	flat rank	MiF	EBP	EBR	EBF	MaP	MaR	MaF	MiP	MiR	Acc.	hier. rank	LCA-F	HiP	HiR	HiF	LCA-P	LCA-R
best	1/33	0.571	0.562	0.594	0.560	0.474	0.564	0.535	0.560	0.583	0.403	1/33	0.494	0.693	0.743	0.692	0.488	0.531
hace2	15/33	0.503	0.471	0.561	0.493	0.395	0.404	0.382	0.471	0.540	0.340	13/33	0.457	0.652	0.685	0.641	0.462	0.487
hace2-ne	16/33	0.495	0.463	0.553	0.486	0.394	0.403	0.379	0.463	0.532	0.333	15/33	0.450	0.643	0.676	0.633	0.454	0.479
hace1	22/33	0.434	0.406	0.492	0.429	0.296	0.366	0.342	0.406	0.466	0.285	17/33	0.408	0.586	0.643	0.590	0.406	0.443
rebayct2	26/33	0.347	0.324	0.400	0.344	0.340	0.271	0.248	0.324	0.372	0.214	20/33	0.357	0.566	0.606	0.562	0.351	0.394
rebayct	29/33	0.307	0.288	0.351	0.304	0.445	0.185	0.171	0.288	0.330	0.184	29/33	0.308	0.547	0.510	0.505	0.296	0.347

Table 4. Official results for BioASQ batch 3.