CoLe and UTAI participation at the 2014 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. In this paper we describe our participation in the second edition of the BioASQ biomedical semantic indexing challenge. As in the previous edition, the approach followed by our team is based on the use of hierarchical classification techniques, employing two systems, one based on a hierarchy of local classifiers and a another one using a Bayesian network built from the MeSH thesaurus structure. In this second participation we have focused our experiments in evaluating different document pre-processing alternatives, trying to take advantage of basic linguistic processing operations. We also have tested different methods for combining the results of ensembles of our classifiers. Unfortunately obtained results are a bit disappointing and our new proposals only were able to get marginal improvements when applied to our Bayesian network based hierarchical classifier.

1 Introduction

This article describes the joint participation of a group from the University of Vigo and another group from the University of Granada in the biomedical semantic indexing task of the 2014 BioASQ challenge. Participants in this task are asked to classify new MEDLINE documents, labeling those documents with descriptors taken from MeSH hierarchy. Three batches, of five test sets each, were released. In our case, due to technical and organizational issues, we only were able to take part into Test Batch 2 and Test Batch 3, and actual innovations with respect to our previous participation were only tested on Test Batch 3 runs.

Both groups (CoLe [4] from University of Vigo and UTAI [5] from University of Granada) have participated in the previous BioASQ edition. Our previous participation assessed the use of two different machine learning based techniques. Both approaches modelled the task of assigning descriptors from the MeSH hierarchy to MEDLINE documents as a hierarchical classification problem. The first one of these systems employed a classical top-down approach based on a collection of local classifiers, trained to determine whether the class corresponding to every hierarchy node, or to one of its descendant classes, is suitable to be assigned to a given document. The other system builds a Bayesian network induced by the thesaurus structure and contents, taking into account the influence of tokens taken from descriptor labels and from related term labels, and also tokens coming from the training documents.

Previous year participation aimed to check the suitability of our approaches in a huge domain, with a complex terminology and with strict time and processing restrictions. In that case minimal text pre-processing was performed in order to evaluate the performance of a pure machine learning approach in thesaurus descriptor assignment. In this year participation we have added minor changes to our automatic categorization tools with respect to versions employed in the previous editon of the challenge, focusing our experiments in testing the effect of performing more powerful linguistic processing on the input documents and in testing several ways to build and to combine ensembles of classification models.

The rest of the paper is organized as follows. Section 2 briefly describes the main ideas behind the two hierarchical classification schemes that we have employed in our two participation in the BioASQ challenge. Section 3 gives details about the linguistic pre-processing techniques that were applied in our official runs for the BioASQ semantic indexing task. Section 4 shows the approaches regarding classifier ensemble build and combination that were tested. Section 5 discusses our official runs in the BioASQ challenge and details the most relevant conclusions of our participation.

2 Hierarchical text categorization methods

There have been no substantial changes in the hierarchical categorization schemes employed in this year experiments with respect to our previous participation in the BioASQ semantic indexing challenge. We have only incorporated some bug fixes and small performance improvements.

In order to help the reader, this section includes a short summary with the main characteristics of our two hierarchical categorization approaches, which are described in deeper detail in their respective research papers.

2.1 HACE framework: 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

[4] *Compiler and Languages* group, http://www.grupocole.org/

[5] *Uncertainty Treatment in Artificial Intelligence* group, http://decsai.ugr.es/utai/

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 [4].

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 in, 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 the set of positive examples following a bottom-up procedure
- strategy for building/selecting the set of negative examples
- 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 or asymmetric bagging

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] index, which helps in the efficient computation of 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 is employed. This positive example selection policy considers as positive example every instance 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 examples per local model and a k-means clustering based approach, where examples close to the identified centroids are selected as positive examples in order to represent

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

the current class and its descendants in further local model building at 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 [5] 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, as a result of 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 in a top-down fashion to build the list of classes that will be employed to label that example. This top-down search 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 the feature vectors. For a given document to be labeled, the Lucene index is queried using the document textual 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 [6]. 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) committed by local models in the higher levels of the taxonomy which will result in a premature discard of useful paths.

2.2 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.

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 includes, for each concept in the taxonomy, the information coming from descriptor labels and non-descriptor labels and is able to manage the different information sources (hierarchy and equivalence relationships, and training data) influencing a concept. Each concept, labeled identically as the descriptor representing it, will be a node C in the network. Each descriptor and each non-descriptor [7] in the thesaurus will also be nodes D and ND in the network. All the words or tokens appearing in either a descriptor label, a non-descriptor label or a training document will be term nodes T. Additionally, for each concept node C we shall also create three (virtual) nodes: E_C, which will receive the information provided by the equivalence relationships between descriptor and non-descriptors 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.

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, 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).

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 Document pre-processing

In our first participation in the BioASQ semantic indexing challenge we applied elemental text processing operations. During the pre-processing phase performed on training and validation documents we only employed stop-word removal and the default English stemmer from the Snowball project[8]. Additionally, preliminary experiments extracting unrestricted word bigrams from documents and

[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.

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

thesaurus labels were done in our final runs, obtaining some performance improvements, but also increasing the training phase computational cost.

In this second participation we have tried to employ more sophisticated linguistic processing as our first attempt to improve the performance of our systems. Given that in this challenge we are dealing with a complex domain, biomedical documentation, where a very specific language and a rich terminology is employed, it is reasonable to propose the use of advanced text processing procedures. Nevertheless, in this competition we are also dealing with a huge categorization problem and we had to limit our initial plans to balance text processing complexity and problem size and cost. So, we have discarded to employ more powerful linguistic tools like a dependency parser or even, named entity recognizer, and try to approximate the results given by those kinds of tools using simpler approaches.

Over the given documents collections we have applied three pre-processing operations described in the following sections: (1) expanding local abbreviations and acronyms, (2) applying morphosyntactic analysis with a part-of-speech (PoS) tagger and a lemmatizer to filter non-content PoS and to normalize tokens using lemmas, and (3) identifying "real" word bigrams using association measures.

3.1 Expanding local abbreviations

One of the advantages of dealing with the the kind of document collection being employed in the BioASQ semantic indexing challenge is the high writing quality of the texts. MEDLINE abstracts are written by experts and they have been reviewed by other domain experts prior to final publication. Also, scientific language has a well established set of conventions which are used in a fairly consistent way by authors. One of these conventions is the use of abbreviations and acronyms, most of them with a local usage and meaning within a particular paper. Another common practice is the expansion of these short forms into its long form in its first usage, using a well known set of patterns. Additionally these short forms, both common use abbreviations and the local ones, tend to refer to the main topics of a paper and use to appear several times within the document, being a potential indicator of actual document contents.

We have employed a slightly adapted version of the local abbreviation identification method described in [7]. This method [9] scans the input texts searching for <*short-form, long-form*> pair candidates, determined by adjacency to parentheses [10], and uses several heuristics to identify the correct long forms in the ambiguous cases, giving preference to the larger ones. Authors report their method achieves 96% precision and 82% recall in an evaluation dataset comprising 1000 randomly selected MEDLINE abstracts.

[9] Source code provided by original authors is available at http://biotext.berkeley.edu/software.html

[10] Authors employ two basic patterns: `long_form '('short_form')'` and `short_form '('long_form')'`

Table 1. Examples of local abbreviations and acronyms that were identified.

short form	long form	
IFP	*inflammatory fibroid polyps*	5 occurrences within the abstract
ACTH	*adrenocorticotropic hormone*	5 occurrences within the abstract
PEM	*protein energy malnutrition*	2 occurrences within the abstract
QoL	*quality of life*$^{(*)}$	14 occurrences within the abstract
	(*) This is an actual MeSH descriptor, with ID D011788 and tree number F01.829.458	

In our case we have pre-processed with this tool the 4,458,300 documents in the training set B provided by the BioASQ challenge organization, which is restricted to the journals being employed to provide the set of actual test document collections in every evaluation batch. An example of the kind of replacement pairs that were identified is shown in table 1. Once all pair candidates for each abstract are identified and filtered, we have expanded every occurrence of the respective short forms within the abstract text. In our pre-processing phase this abbreviation expansion took place in a 10.1 % of the processed documents.

3.2 Morphosyntactic processing

Even our initial aim was to try to evaluate the effect of complex linguistic processing, like dependence parsing, in the overall categorization performance due to technical difficulties we had to switch our initial plan and employ much more simple tools. We limited the linguistic processing in our runs to deal with the effects of morphosyntactic variation using a lemmatizer to identify lexical roots instead using word stems and to replace stop-word removal with a content-word selection procedure based on part-of-speech (PoS) tags.

We have delegated the linguistic processing tasks to the tools provided by the ClearNLP project [11]. ClearNLP project offers a set of state-of-the-art components written in the Java programming language, together with a collection of pre-trained models, ready to be used in typical natural language processing tasks, like dependence parsing, semantic role labeling, PoS tagging and morphological analysis.

In our case we have employed the PoS tagger [8] from the ClearNLP project to tokenize and assign PoS tags to tokens coming from the training documents, from the documents included in every test batch and also from the labels taken from MeSH thesaurus descriptors. We also employed the two tagging models available on ClearNLP repository to feed this PoS tagger: a general domain model, trained on various corpora, mainly newswires and web texts, and a specific medical domain model, trained on corpora collected by several research projects, mainly from clinical notes and health articles. Even most part of MEDLINE abstract contents go beyond pure medical texts, those trained resources offered to us a fairly good results with no need of additional specific training.

[11] Available at http://clearnlp.wikispaces.com/

In order to filter the content-words from the processed MEDLINE abstracts, we have applied a simple selection criteria based on the employment of the PoS that are considered to carry the sentence meaning. Only tokens tagged as a noun, verb, adjective or as an unknown word are taking into account to constitute the final document representation to be employed on further processing phases. In case of ambiguous PoS tag assignment, whether the second most probable PoS tag is included in the list of acceptable tags, that token is also taken into account.

After PoS filtering, the ClearNLP lemmatizer is applied on the surviving tokens in order to extract the canonical form of those words. This way we have a method to normalize the considered word forms that is slightly more consistent than the simple stemming algorithm employed in our previous participation, that had had some issues regarding the complexity of the specific terminology being employed in the MEDLINE abstracts. Like in the previous case, we have customized the lemmatization process using the general dictionary model and the medical dictionary model available in the ClearNLP project repositories.

3.3 Bigram association measures

Following our approach of trying to improve the linguistic pre-processing without the employment of complex and costly NLP tools, we have gone one step further in our previous year experiments with word bigrams, which gave us some performance improvements. What we have done is to employ those bigrams, conveniently filtered, as a simple replacement for actual multiword terms. The intuition behind this is idea of exploiting multiword terms in text categorization tasks is that in a domain with a complex terminology, like the one in the BioASQ challenge, those multiword terms used to carry a higher discriminative power as evidences than the simple terms that constitute them.

In order to discover multiword terms present in MEDLINE documents without having to appeal to deep natural language analysis, we have created a set of approximately multiwords terms by computing association measures over the set of bigrams built from the filtered and normalized lists of tokens extracted by means of the previously discussed use of ClearNLP PoS tagger and lemmatizer. To compute those association measures we have employed the Ngram Statistics Package (NSP) described in [9]. The NSP toolkit [12] includes utility Perl scripts to produce a list of n-grams occurring in the input files and scripts to run the association measure selected by the user in order to determine which of those n-grams can be considered as multiword collocations. Several association measures are available, such as Dice coefficient, Jaccard coefficient, Log-likelihood ratio, Mutual Information, Pearson's Chi Squared Test, Pointwise Mutual Information, Poisson Stirling Measure and many other.

In our experiments we have pre-processed with the NSP tools the set of PoS filtered lemmas extracted from the documents in the 4,458,300 training collection provided by BioASQ organizers to create a preliminary list of bigrams, discarding those bigrams occurring in less than 5 documents. This bigram list was ranked

[12] Available at http://ngram.sourceforge.net/

according to the Mutual Information values computed by the NSP statistics toolkit using the following formula:

$$MI(w_1, w_2) = log_2 \left(\frac{P(w_1, w_2)}{P(w_1)P(w_2)} \right)$$

where w_1, w_2 are the lemmas being part of the evaluated bigram, $P(w_1)$ and $P(w_2)$ are the relative frequency of each single lemma and $P(w_1, w_2)$ is the relative frequency of the evaluated bigram. In our runs, assigning different values to k, we have selected the top k bigrams from the resulting ranked list to be considered as actual multiword terms, which will be treated as a single unit within our text classification procedures.

4 Combining ensembles of classification models

The second line of improvement with respect to our participation in the previous edition of the BioASQ challenge is related with the building and combination of ensembles of classification models with the expectation that a mixing of proposals coming from an expert committee could improve the overall quality of the individual predictions.

We have tested in our official runs two strategies to build those ensembles of classifiers: (1) splitting the classification task across a set of partial models specialized in each one of the MeSH subhierarchies and (2) an iterative building approach that splits training instances trying to ensure a greatest diversity in the resulting categorization models. As it will be explained in following sections, both strategies have demonstrated to be useful only when applied on the Rebayct hierarchical categorization models, having a minimal or null impact when employed together with the HACE framework.

4.1 Model per subhierarchy approach

Our first classifier ensemble approach is a direct consequence of the MeSH thesaurus structure. According to their *Tree Numbers*, MeSH descriptors are arranged in a set of partially overlapping subhierarchies. Without lost of generality, those subhierarchies can be assumed to be independent, once shared subtrees included in the actual DAG (direct acyclic graph) are conviniently replicated. In this way we can arrange the 27,149 descriptors in 2014 MeSH thesaurus distribution into 16 subhierarchies, comprising a total amount of 55,611 descriptors distributed as it is shown in table 2.

With this arrangement in mind we can split the whole categorization task into 15 [13] smaller partial categorization tasks, plus an additional phase where partial results will be aggregated to build the final ranked list of proposed classes.

[13] Descriptors from subhierarchy *(V) Publication Characteristics* are not employed to label MEDLINE contents.

Table 2. Descriptor distribution among MeSH subhierarchies

subhierarchy	# of descriptors	subhierarchy	# of descriptors
(A) Anatomy	2,927	(I) Anthropology, Education, Sociology and Social Phenomena	651
(B) Organisms	5,196		
(C) Diseases	11,303	(J) Technology and Food and Beverages	601
(D) Chemicals and Drugs	20,992		
(E) Analytical, Diagnostic and Therapeutic Techniques and Equipment	4,764	(K) Humanities	218
		(L) Information Science	519
		(M) Persons	258
(F) Psychiatry and Psychology	1,150	(N) Health Care	2,350
(G) Biological Sciences	3,428	(V) Publication Characteristics	188
(H) Physical Sciences	513	(Z) Geographic Locations	553

In the case of the Rebayct tool, 15 independent models can be trained employing as training documents the respective splits taken from the original traning set, according to the subhierarchy links with the corresponding document true classes. During the labeling phase, a previous step is needed to determine: (1) which models should be checked, (2) how many labels should be recovered from each partial ranked list to be included in the final list of proposed classes and (3) which weights should be assigned to the predictions made by each selected submodel in order to rank the final list of classes. In our runs we have covered all this requirements using a simple strategy that exploits the Lucene index build in our runs with the HACE framework. For every new document, this index is queried and the most similar training documents are retrieved. Then, a simple weighted voting scheme is employed. The subhierarchies to which the real descriptors of those similar documents belongs to receive a weighted vote equals to the similarity scores assigned by the Lucene engine. By aggregating those scores we can determine which submodels have to be checked, how many descriptors should be retrieved from them and how they should be ranked in the list of candidate classes.

In the case of the HACE framework, splitting into 15 subcategorization tasks is implicitly done, since the basic top-down search procedure is always started in an independent way over all of the roots of the 15 subhierarchies. Additionally, in the case of our guided top-down search, the starting nodes are determined by an initial similarity search over the backing Lucene index, which in fact results in a approach very close to the one previously described for the Rebayct case.

4.2 Introducing diversity into the trained models

Another ensemble based strategy was tested with the Rebayct tool. Due to memory usage limitations of this tool only a limited amount of documents could be employed in the training phase. In our previous participation a bagging based approach was applied to try to take advantage of the huge amount of available training data. We employed 5 set of about 250,000 randomly selected documents to train 5 different Rebayct models that were combined through simple voting in the labeling phase. Even that this simple ensemble scheme achieved some

improvements with respect to the usage of a single Rebayct model, a thorough review of the lists of classes being proposed showed that there was little diversity on the models being employed. Those models used to return fairly similar lists of candidates classes, which were heavily dominated in their top positions by the most frequent classes in the MEDLINE training corpus.

To try to overcome this behaviour an iterative approach mixing instance selection and model building was tested. From a randomized version of the original 4,458,300 documents training collection, examples were taken one at a time. Those examples where labeled with the "current" Rebayct model, initially built from the thesaurus labels without using training data. Whether the model performance on the given example was considered bad (measured using a threshold on the F value obtained for that particular prediction) that example was recorded as a "interesting" one. In the case of an acceptable performance of the "current" model on the given example, this instance is no further taken into account, since its considered that the current model actually knows how to deal with that kind of instances. Once the number of "interesting" examples reaches a fixed amount (100,000 documents in our experiments) a new model is trained with those examples and it becomes the new "current" model to be employed in the next instance selection iteration.

5 Official runs in BioASQ indexing challenge

In table 3 the official performance measures obtained by our runs in the Test Batch number 3 are shown. Even we were sending our results during the Test Batch number 2, the systems participating in those runs were essentially the same that were employed in the first BioASQ competition, with the difference of being trained with the new 4,458,300 documents collection pre-processed by the ClearNLP tools and with minor parameter tuning in the HACE runs. Innovations described in previous sections only take part in the Test Batch number 3.

The official runs sent by our group during our participation in the Test Batch number 3 were created using the following configurations. In the case of models built with the HACE framework, the only relevant change with respect to previous year participation was the use of filtered word bigrams and some improvements in the configuration of term selection and document selection at each local model, which this time has a fairly larger number of accepted terms and documents.

hace1. HACE framework using k-means bottom-up positive example selection with up to 5000 examples per node, Information Gain feature selection with up to 500 top ranked features per node and a SVM classifier as content based router for each model. This run employs the guided top-down search approach described in section 2.1, where a previous filtering step exploits a set of candidate descriptors taken from the similar documents retrieved from the Lucene index. Bigrams from the top 1 % of the ranked list of word bigrams, ordered according the selected association measure, were considered as actual multiword terms during training and classification.

hace2. Same configuration as **hace1** without using the guided top-down search approach.

hace-ne. Same configuration as **hace1** using the top 5 % of the ranked list of word bigrams.

rebayct. A set of 15 Rebayct models following the ensemble scheme described in section 4.1. Bigrams from the top 1 % of the ranked list of word bigrams were considered as actual multiword terms during training and classification.

rebayct2. A set of 20 Rebayct models following the ensemble scheme described in section 4.2, setting a threshold of 0.40 in the F values employed to determine the success of the Rebayct model in the current iteration. Bigrams from the top 1 % of the ranked list of word bigrams were considered as actual multiword terms during training and classification.

6 Results discussion and conclusions

The results of our participation in the second edition of the BioASQ biomedical sematic indexing challenge have been a bit disappointing, with our official runs falling into the bottom third of the ranked list of participating systems and with small performance improvements in relation with our first participation at the challenge. The new strategies that has been included in our systems did not get significant improvements in the case of HACE framework and only in the Rebayct system runs some marginal improvements were achieved, that could lead to further lines of work.

Using lemmatization and content-word filtering based on PoS tagging produced minor improvements over the use of stemming and stop-word removal employed in our previous participation. Our hypothesis in the case of HACE based experiments is that the feature selection phase at every taxonomy node performs a very aggressive removal of terms that could overtake the decisions about the extracted single word terms made by the linguistic tools employed at document pre-processing phase. The behaviour of single word terms in the Bayesian network built by Rebayct system it is not so clear and needs a deep analysis of the resulting topology and probabilities.

Employing bigrams as a replacement of complex multiword terms had shown its suitability in our first participation, specially in the runs involving Rebayct models. The usage of association measures in ranking and selection of relevant word bigrams was apparently able to catch useful multiword terms in the biomedical domain, maintaining the overall categorization performance with a smaller cost in time and storage during training and labeling phases. Being this reduction in memory footprint extremely relevant in the case of Rebayct categorization system. In the case of HACE framework it is not evident the effect of word bigrams in the categorization performance, but the use of association measures in reducing the total number of considered bigrams also had a great effect on reducing time and space requirements in HACE training and labeling processes.

Ensemble based approaches only have had real effect on the Rebayct based runs, and tested strategies have helped to improve the categorization performance. Building Bayesian networks specific to each MeSH sub-hierarchy lead

to an improvement in the categorization quality. The mechanism employed to select and weight the pertinent sub-hierarchies is an aspect to be improved in future research, since the partial results obtained in the evaluation of individual sub-hierarchy models used to be better than the combination of those partial results. Maybe some kind of regression model approach like the one employed by the winners of first edition of BioASQ challenge could give better results than the simple voting scheme based on Lucene similarity that we have applied in our experiments. The incremental instance selection method has been probably the main contribution of our participation in this challenge. This approach to building ensembles of Rebayct models was able to introduce some diversity into the final set of models, that had a positive effect on the overall categorization results. Nevertheless, more work is needed on defining the iterative instance selection process and tuning model parameters.

Acknowledgements

Research reported in this paper has been partially funded by "Ministerio de Economía y Competitividad" and FEDER (project TIN2010-18552-C03-01), by "Xunta de Galicia" (project CN 2012/317) and by "Consejería de Innovación, Ciencia y Empresa de la Junta de Andalucía" (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. 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
5. 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. 11–18, 2009
6. 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.
7. A.S. Schwartz, M.A. Hearst. Algorithm for Identifying Abbreviation Definitions in Biomedical Text. Pacific Symposium on Biocomputing 8:451-462(2003)
8. Jinho D. Choi, Martha Palmer. Fast and Robust Part-of-Speech Tagging Using Dynamic Model Selection, Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL'12), 363-367, Jeju, Korea, 2012.
9. Pedersen, T. and Banerjee, S. The Design, Implementation and Use of the Ngram Statistics Package. In Proceedings of the 4th International Conference on Intelligent Text Processing and Computational Linguistics, p. 370-381. 2003. Mexico City.

Table 3. Official results for BioASQ batch 3.

week 1, labeled documents: 2978/4342																		
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	1/36	0.6001	0.5929	0.6236	0.5898	0.5547	0.5298	0.5004	0.5928	0.6075	0.4327	1/36	0.5030	0.7476	0.7329	0.7186	0.5212	0.5154
hace1	27/36	0.4790	0.5163	0.4697	0.4718	0.4814	0.3095	0.3099	0.5163	0.4468	0.3236	21/36	0.4171	0.7016	0.5963	0.6182	0.4725	0.4015
hace_ne	29/36	0.4736	0.5104	0.4634	0.4658	0.4812	0.2927	0.2942	0.5104	0.4417	0.3177	22/36	0.4098	0.6942	0.5842	0.6080	0.4661	0.3939
hace2	30/36	0.4707	0.5073	0.4599	0.4629	0.4814	0.2913	0.2929	0.5073	0.4390	0.3157	23/36	0.4097	0.6970	0.5852	0.6101	0.4663	0.3933
rebayct_2	34/36	0.3420	0.3687	0.3393	0.3384	0.4424	0.1825	0.1820	0.3687	0.3190	0.2109	33/36	0.3321	0.6159	0.5280	0.5440	0.3654	0.3286
rebayct	35/36	0.2927	0.3154	0.2843	0.2867	0.5708	0.0990	0.1056	0.3154	0.2730	0.1737	35/36	0.2760	0.5858	0.4057	0.4559	0.3057	0.2730

week 2, labeled documents: 5717/8840																		
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	1/39	0.6052	0.6252	0.6025	0.5983	0.5902	0.5176	0.5028	0.6346	0.5785	0.4416	1/39	0.5105	0.7631	0.7401	0.7306	0.5324	0.5208
hace1	26/39	0.4741	0.5012	0.4755	0.4683	0.4749	0.3120	0.3127	0.5012	0.4497	0.3189	23/39	0.4172	0.6935	0.6107	0.6245	0.4613	0.4097
hace_ne	28/39	0.4674	0.4942	0.4661	0.4607	0.4684	0.2916	0.2941	0.4942	0.4434	0.3130	25/29	0.4111	0.6908	0.5994	0.6165	0.4571	0.4021
hace2	29/39	0.4672	0.4939	0.4673	0.4610	0.4725	0.2924	0.2934	0.4939	0.4432	0.3130	24/39	0.4120	0.6918	0.6022	0.6187	0.4571	0.4034
rebayct_2	34/39	0.3450	0.3648	0.3482	0.3414	0.4442	0.1971	0.1955	0.3648	0.3273	0.2129	33/39	0.3345	0.6117	0.5448	0.5525	0.3587	0.3373
rebayct	35/39	0.2931	0.3099	0.2920	0.2881	0.5922	0.1182	0.1281	0.3099	0.2781	0.1747	35/39	0.2789	0.5893	0.4310	0.4750	0.3001	0.2831

week 3, labeled documents: 2737/3702																		
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	1/39	0.6297	0.6478	0.6312	0.6237	0.5892	0.5344	0.5156	0.6561	0.6054	0.4679	1/39	0.5230	0.7767	0.7269	0.7309	0.5560	0.5210
hace1	27/39	0.4958	0.5232	0.5005	0.4911	0.4666	0.3336	0.3279	0.5232	0.4711	0.3398	23/39	0.4324	0.6953	0.6176	0.6270	0.4775	0.4261
hace_ne	29/39	0.4888	0.5158	0.4916	0.4833	0.4551	0.3169	0.3121	0.5158	0.4645	0.3333	27/39	0.4247	0.6901	0.6066	0.6188	0.4708	0.4173
hace2	30/39	0.4875	0.5144	0.4908	0.4824	0.4570	0.3164	0.3090	0.5144	0.4632	0.3325	25/39	0.4258	0.6906	0.6069	0.6189	0.4709	0.4190
rebayct_2	34/39	0.4625	0.4880	0.4698	0.4591	0.5001	0.3425	0.3371	0.4881	0.4394	0.3079	31/39	0.4072	0.6929	0.6376	0.6391	0.4341	0.4117
rebayct	37/39	0.3760	0.3968	0.3794	0.3723	0.6400	0.2073	0.2160	0.3968	0.3573	0.2371	36/39	0.3308	0.6492	0.5136	0.5490	0.3555	0.3363

week 4, labeled documents: 3050/4726																		
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	1/39	0.6311	0.6526	0.6298	0.6238	0.5945	0.5349	0.5196	0.6530	0.6106	0.4675	1/39	0.5233	0.7842	0.7326	0.7377	0.5549	0.5227
hace1	22/39	0.4916	0.5331	0.4832	0.4876	0.4773	0.3113	0.3110	0.5331	0.4561	0.3362	18/39	0.4244	0.7102	0.5984	0.6257	0.4805	0.4065
hace_ne	23/39	0.4876	0.5287	0.4779	0.4829	0.4741	0.2936	0.2961	0.5287	0.4524	0.3319	22/39	0.4200	0.7079	0.5912	0.6202	0.4793	0.4002
hace2	24/39	0.4859	0.5269	0.4770	0.4815	0.4785	0.2974	0.2989	0.5269	0.4509	0.3305	20/39	0.4207	0.7092	0.5907	0.6204	0.4800	0.4003
rebayct_2	28/39	0.4638	0.5029	0.4601	0.4615	0.5279	0.3188	0.3227	0.5029	0.4303	0.3094	26/39	0.4062	0.7146	0.6248	0.6441	0.4433	0.4004
rebayct	34/39	0.3718	0.4031	0.3670	0.3690	0.6444	0.1786	0.1908	0.4031	0.3449	0.2342	33/39	0.3289	0.6668	0.5002	0.5488	0.3612	0.3257

week 5, labeled documents: 2997/4533																		
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	1/39	0.6219	0.6490	0.6201	0.6156	0.6052	0.5398	0.5246	0.6459	0.5996	0.4608	1/39	0.5165	0.7470	0.7573	0.7337	0.5245	0.5374
hace_ne	27/39	0.4604	0.4747	0.4721	0.4549	0.4033	0.3024	0.2942	0.4747	0.4470	0.3088	21/39	0.4074	0.6610	0.6067	0.6060	0.4419	0.4074
hace1	28/39	0.4599	0.4742	0.4733	0.4551	0.3954	0.3160	0.3042	0.4742	0.4465	0.3085	19/39	0.4097	0.6645	0.6145	0.6122	0.4438	0.4102
hace2	29/39	0.4582	0.4724	0.4700	0.4526	0.3977	0.3017	0.2924	0.4724	0.4448	0.3068	22/39	0.4071	0.6623	0.6082	0.6072	0.4425	0.4062
rebayct_2	31/39	0.4510	0.4651	0.4683	0.4478	0.4836	0.3310	0.3244	0.4650	0.4379	0.2981	24/39	0.4001	0.6797	0.6372	0.6326	0.4241	0.4080
rebayct	35/39	0.3403	0.3509	0.3518	0.3370	0.6218	0.1503	0.1595	0.3509	0.3304	0.2100	35/39	0.3053	0.6151	0.4800	0.5143	0.3230	0.3147