CoLe and UTAI at BioASQ 2015: experiments with similarity based descriptor assignment

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 third edition of the BioASQ biomedical semantic indexing challenge. Unlike our participation in previous editions, we have chosen to follow an approach based solely on conventional information retrieval tools. We have evaluated various alternatives for creating textual representations of MEDLINE articles to be stored in an Apache Lucene textual index. Those indexed representations are queried using the contents of the article to be annotated and a ranked list of candidate descriptors is created from the retrieved similar articles. Several strategies to post-process those lists of candidate descriptors were evaluated. Performance in the official runs were far from the most competitive systems, but taking into account that our approach in the performed runs did not employ any external knowledge sources, we think that the proposed method could benefit from richer representations for MEDLINE contents.

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 2015 BioASQ challenge. Participants in this task are asked to classify new MEDLINE articles, labeling those documents with descriptors taken from MeSH hierarchy.

Both groups (CoLe [4] from University of Vigo and UTAI [5] from University of Granada) have participated in the previous BioASQ editions. Our previous par-

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

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

ticipations assessed the use of two different machine learning based techniques: a top-down arrangement of local classifiers and a Bayesian network induced by the thesaurus structure. Both approaches modelled the task of assigning descriptors from the MeSH hierarchy to MEDLINE documents as a hierarchical multilabel classification problem.

In this year participation we have changed the basic approach of our systems, following a similarity based strategy, where the final list of MESH descriptors assigned to a given article is created from the set of most similar MEDLINE articles stored in a textual index created from the training dataset. This neighbor based strategy was partially explored in our previous participations in BioASQ challenge, where a sort of k nearest neighbor was employed as a guide in the top-down traversal of local classifiers approach and also in the selection of submodels (one per MeSH subhierarchy) in the Bayesian network based method. The employment of this k nearest neighbor filtering was mainly due to performance and scalability reasons, but it also had some positive effects on overall annotation quality. For the third BioASQ challenge we have concentrated our efforts on testing the suitability of this similarity based approach and on evaluating several strategies to improve the final ranked list of descriptors.

The rest of the paper is organized as follows. Section 2 briefly describes the main ideas behind the proposed similarity based approach for MEDLINE article annotation and also describes the text processing being applied. Section 3 gives details about the strategies for improving the final list of ranked descriptors by means of several post-processing methods. Finally, section 4 discusses our official runs in the BioASQ challenge and details the most relevant conclusions of our participation.

2 Similarity based descriptor selection

Approaches based on k nearest neighbors (k-NN) have been widely used in the context of large scale multilabel categorization, even with MEDLINE documents [1]. The choosing of k-NN based methods is mainly due to its scalability, minimum parameter tuning requirements and, despite its simplicity, its ability to deliver acceptable results in cases where large amounts of examples are available. The approach we have followed in our BioASQ challenge participation is essentially a large k-NN classifier, backed by an Apache Lucene [6] index, with some optimizations due to MeSH usage recommendations on MEDLINE articles annotation. In the case of MEDLINE annotation with MeSH descriptors, despite of being a complex problem, with more than 25,000 possible classes, arranged in a directed acyclic graph (DAG), the availability of a huge training set labeled by human experts supposes an a priori favorable scenario for labeling estimates based on k-NN.

In our case we have tried to take advantage of certain aspects of semantic indexing process with the MeSH thesaurus to improve the labeling process based

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

A	Pregn	D011247	Pregnancy	**J**	Cats	D002415	Cats
B	Inf New (to 1 mo)	D007231	Infant, Newborn	**K**	Cattle	D002417	Cattle
C	Inf (1 to 23 mo)	D007223	Infant	**L**	Chick Embryo	D002642	Chick Embryo
D	Child Pre (2-5)	D002675	Child, Preschool	**M**	Dogs	D004285	Dogs
E	Child (6-12)	D002648	Child	**O**	Guinea Pigs	D006168	Guinea Pigs
F	Adolesc (13-18)	D000293	Adolescent	**P**	Hamsters	D006224	Cricetinae
R	Young Adult (19-24)	D055815	Young Adult	**Q**	Mice	D051379	Mice
G	Adult (19-44)	D000328	Adult	**S**	Rabbits	D011817	Rabbits
H	Mid Age (45-64)	D008875	Middle Aged	**T**	Rats	D051381	Rats
I	Aged (65-79)	D000368	Aged				
N	Aged (80+)	D000369	Aged, 80 and over	**c**	Ancient	D049690	History, Ancient
				d	Medieval	D049691	History, Medieval
U	Animal	D000818	Animals	**f**	15th Cent	D049668	History, 15th Century
V	Human	D006801	Humans	**g**	16th Cent	D049669	History, 16th Century
W	Male	D008297	Male	**h**	17th Cent	D049670	History, 17th Century
X	Female	D005260	Female	**i**	18th Cent	D049671	History, 18th Century
				j	19th Cent	D049672	History, 19th Century
Y	In Vitro (PT)	D066298	In Vitro Techniques	**k**	20th Cent	D049673	History, 20th Century
b	Comp Study (PT)	D003160	Comparative Study	**o**	21st Cent	D049674	History, 21st Century

Fig. 1. Check Tag list according to MeSH annotation guidelines

on similarity. Following MeSH annotation guidelines [5] we propose a differentiated treatment for Check Tags. According to MeSH guidelines, Check Tags are widely used descriptors, shown in Figure 1, which describe some of the broader aspects of the MEDLINE articles. MeSH annotators can assign an arbitrary number of these Check Tags without any restriction regarding their location in the thesaurus hierarchy.

To try to exploit this singularity, our system separates the processing of Check Tags and the processing of regular MeSH descriptors. In this way, our annotation scheme starts by indexing the contents of the MEDLINE training articles. For each new article to annotate that index is queried using its contents as query terms. The list of similar articles returned by the indexing engine and their corresponding similarity measures are exploited to determine the following results:

- predicted number of Check Tags to be assigned
- predicted number of regular descriptors to be assigned
- ranked list of predicted Check Tags
- ranked list of predicted regular descriptors

The first two aspects conform a regression problem, which aims to predict the number of Check Tags and descriptors to be included in the final list, depending on the number of Check Tags and descriptors assigned to the most similar articles identified by the indexing engine and on their respective scores. The other two tasks are multilabel classification problems, which aim to predict a Check Tags list and a regular descriptors list based on the descriptors and Check Tags manually assigned to the most similar MEDLINE articles. In both cases, regression and multilabel classification based on k-NN, similarity scores calculated by the indexing engine are exploited. These scores are computed during the query processing phase. Query terms employed to retrieve the similar articles are extracted from the original article contents and linked using a global OR operator to conform the final query sent to the indexing engine.

In our case, the scores provided by the indexing engine are similarity measures resulting from the engine internal computations and the weighting scheme being employed, which do not have an uniform and predictable upper bound. In order to get those similarity scores behave like a real distance metric we have applied the following normalization procedure:

1. Articles to be annotated are preprocessed in the same way than the training articles and are indexed by the Lucene engine
2. In classification time, all of the relevant index terms from the article being annotated are joined by an OR operator to create the search query
3. In the similar articles ranking returned by the indexing engine the top result will be the same article used to query the index, this result is discarded but its score value ($score_{\text{MAX}}$) is recorded for future normalization
4. For each element on the remaining articles set the number of Check Tags and regular descriptors are recorded and it is also recorded the list of real Check Tags and the list of real descriptors, assigning to each of them an estimated distance to the article being annotated, equals to $\left(1 - \frac{score}{score_{\text{MAX}}}\right)$, which will be employed in the weighted voting scheme of the k-NN classification.

With this information the number of Check Tags and the number of regular descriptors to be assigned to the article being annotated is predicted using a weighted average scheme, where the weight of each similar article is the inverse of the square of the estimated distance to the article being annotated, that is, $\frac{1}{\left(1-\frac{score}{score_{\text{MAX}}}\right)^2}$.

To create the ranked list of Check Tags and the ranked list of regular descriptors a distance weighted voting scheme is employed, associating the same weight values (the inverse of squared estimated distances) to the respective similar article. Since this is actually a multilabel categorization task, there are as many vote tasks as candidate Check Tags or candidate regular descriptors were extracted from the articles retrieved by the indexing engine. For each candidate, positive votes come from similar articles annotated with it and negative votes come from articles not including it.

2.1 Evaluation of article representations

In our preliminary experiments we have tested several approaches to extract the set of index terms to represent MEDLINE articles in the indexing process. We have also evaluated the effects in annotation performance of the different weighting schemes available in the Apache Lucene indexing engine.

Regarding article representation, we have employed three index term extraction approaches. In this experiment and also in the official BioASQ runs we have worked only with MEDLINE articles from year 2000 onwards, indexing a total amount of 6,697,747 articles. Index terms which occurred in 5 or less articles were discarded and terms which were present in more than 50 % of training documents were also removed.

Table 1. Evaluation of term extraction approaches

iria-1: n-grams from noun phrase chunks

weighting	k	MiF	MiP	MiR	MaF	MaP	MaR	LCA-F	LCA-P	LCA-R	HiF	HiP	HiR
tfidf	5	0,4662	0,4853	0,4485	0,3289	0,4316	0,3339	0,4098	0,4376	0,4108	0,6125	0,6522	0,6183
	10	0,4884	0,5170	**0,4628**	**0,3400**	0,4865	**0,3413**	0,4211	0,4578	**0,4146**	0,6276	0,6789	**0,6225**
	20	0,4937	0,5297	0,4624	0,3302	0,5057	0,3297	**0,4231**	0,4664	0,4119	**0,6284**	0,6881	0,6169
	25	0,4940	0,5321	0,4609	0,3294	0,5150	0,3274	0,4220	0,4678	0,4094	0,6276	0,6893	0,6149
	30	**0,4946**	**0,5341**	0,4606	0,3256	**0,5173**	0,3235	0,4229	**0,4700**	0,4090	0,6277	**0,6909**	0,6136
bm25	5	0,4667	0,4849	0,4497	0,3291	0,4302	0,3354	0,4105	0,4374	0,4117	0,6133	0,6516	0,6191
	10	0,4871	0,5154	0,4618	0,3390	0,4824	0,3412	0,4203	0,4574	0,4136	0,6252	0,6753	0,6209
	20	0,4922	0,5280	0,4610	0,3315	0,5071	0,3317	0,4209	0,4635	0,4104	0,6268	0,6852	0,6162
	25	0,4921	0,5297	0,4595	0,3272	0,5103	0,3265	0,4211	0,4655	0,4089	0,6263	0,6866	0,6141
	30	0,4918	0,5304	0,4584	0,3246	0,5133	0,3235	0,4195	0,4657	0,4064	0,6255	0,6883	0,6115

iria-2: stemming and stop-word removal

weighting	k	MiF	MiP	MiR	MaF	MaP	MaR	LCA-F	LCA-P	LCA-R	HiF	HiP	HiR
tfidf	5	0,4746	0,4929	0,4576	0,3410	0,4323	0,3496	0,4168	0,4446	0,4188	0,6199	0,6586	0,6267
	10	0,4959	0,5240	0,4706	0,3548	0,4899	0,3588	0,4287	0,4660	0,4228	0,6363	0,6876	0,6311
	20	0,5043	0,5401	0,4729	0,3531	0,5249	0,3547	0,4322	0,4762	0,4214	0,6403	0,6998	0,6293
	25	0,5036	0,5413	0,4708	0,3485	0,5290	0,3493	0,4310	0,4761	0,4192	0,6394	0,7013	0,6262
	30	0,5038	0,5432	0,4697	0,3453	0,5301	0,3456	0,4301	0,4775	0,4168	0,6392	0,7032	0,6246
bm25	5	0,4760	0,4935	0,4597	0,3456	0,4332	0,3555	0,4186	0,4449	0,4214	0,6231	0,6594	0,6316
	10	0,4983	0,5259	0,4734	**0,3578**	0,4912	**0,3629**	0,4311	0,4677	**0,4260**	0,6395	0,6898	**0,6343**
	20	0,5061	0,5413	**0,4752**	0,3530	0,5212	0,3554	**0,4330**	0,4760	0,4229	0,6409	0,6998	0,6302
	25	**0,5073**	0,5444	0,4750	0,3509	**0,5291**	0,3534	0,4329	**0,4778**	0,4214	**0,6410**	0,7025	0,6283
	30	0,5060	**0,5446**	0,4724	0,3472	0,5290	0,3488	0,4315	0,4776	0,4185	0,6407	**0,7040**	0,6264

iria-3 lemmatization and PoS tags filtering

weighting	k	MiF	MiP	MiR	MaF	MaP	MaR	LCA-F	LCA-P	LCA-R	HiF	HiP	HiR
tfidf	5	0,4765	0,4932	0,4609	0,3409	0,4346	0,3503	0,4188	0,4445	0,4215	0,6265	0,6610	0,6354
	10	0,4982	0,5251	0,4740	0,3561	0,4952	0,3600	0,4304	0,4655	0,4256	0,6408	0,6888	0,6383
	20	0,5061	0,5404	0,4758	0,3522	0,5292	0,3531	0,4327	0,4754	0,4221	**0,6461**	0,7022	0,6364
	25	0,5060	0,5429	0,4737	0,3477	0,5328	0,3481	0,4314	0,4758	0,4195	0,6427	0,7029	0,6306
	30	0,5062	0,5436	0,4735	0,3458	**0,5358**	0,3466	0,4314	0,4765	0,4194	0,6430	0,7024	0,6314
bm25	5	0,4792	0,4948	0,4646	0,3465	0,4343	0,3566	0,4203	0,4453	0,4239	0,6283	0,6619	0,6386
	10	0,5016	0,5274	0,4782	**0,3594**	0,4976	**0,3650**	0,4324	0,4658	**0,4292**	0,6450	0,6914	**0,6435**
	20	0,5082	0,5416	**0,4787**	0,3561	0,5261	0,3587	**0,4347**	0,4761	0,4248	0,6457	0,7019	0,6368
	25	**0,5090**	0,5444	0,4779	0,3528	0,5336	0,3548	0,4345	**0,4784**	0,4231	0,6456	0,7019	0,6360
	30	0,5087	**0,5453**	0,4767	0,3486	0,5335	0,3496	0,4335	**0,4784**	0,4217	0,6453	**0,7036**	0,6350

Our aim with these experiments was to determine whether linguistic motivated index term extraction could help to improve annotation performance in the k-NN based method we have described. We employed the following methods:

Stemming based representation. This was the simplest approach which employs stop-word removal, using a standard stop-word list for English, and the default English stemmer from the Snowball project[7].

Some additional post-processing was done using regular expression patterns to remove the most frequent ill-formed stems, like tokens starting with numbers or non-alphabetic characters, which did not resemble chemical compound names and similar cases.

Morphosyntactic based representation. To try to deal with the effects of morphosyntactic variation we have employed a lemmatizer to identify lexical roots instead of using word stems and we also replaced stop-word removal with a content-word selection procedure based on part-of-speech (PoS) tags.

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

We have delegated the linguistic processing tasks to the tools provided by the ClearNLP project [8]. 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 [4] from the ClearNLP project to tokenize and assign PoS tags to the MEDLINE articles contents. We employed the biomedical tagging models available on ClearNLP repository to feed this PoS tagger, since those pre-trained resources offered fairly good results with no need of additional training.

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 unknown words are taken into account to constitute the final article representation. In case of ambiguous PoS tag assignment, if 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 simple stemming. Like in the previous case, we have customized the lemmatization process using the biomedical dictionary model available at the ClearNLP project repositories.

Noun phrases based representation. In order to evaluate the contribution of more powerful Natural Language Processing tools, we have employed a surface parsing approach to identify syntactic motivated noun phrases from which meaningful multi-word index terms could be extracted.

We have employed a chunker from the Genia Tagger project [9] to process MEDLINE abstracts and to identify chunks of words tagged as noun phrases. Genia Tagger employs a maximum entropy cyclic dependency network [6] to model the PoS tagging process and its PoS tagger is specifically trained and tuned for biomedical text such as MEDLINE abstracts. Once the input text has been tokenized and PoS tagged by Genia Tagger, a simple surface parser searches for specific PoS patterns in order to detect the boundaries of the different chunks which can constitute a syntactical unit of interest (nominal phrases, prepositional phrases, verbal phrases and other).

In our processing of MEDLINE articles, from each noun phrase chunk identified in the Genia Tagger output we extract the set of word unigrams (lemmas) and all possible overlapping word bigrams and word trigrams, which will constitute the final list of index terms that will represent the given MEDLINE article in the generated Lucene index.

The reason to limit this multi-word index term extraction process to only word bigrams and trigrams was to try to get a balance between repre-

[8] Available at http://www.clearnlp.com/

[9] Available at http://www.nactem.ac.uk/tsujii/GENIA/tagger/.

sentation power and flexibility and generalization capabilities. The chunks identified by Genia Tagger use to be fairly correct and consistent, even when detecting large noun phrases, but employing as index terms the chunker output without some kind of generalization could lead to poor results during the search phase of the k-NN based annotation. With no generalization this approach could degenerate in being able to find similar articles only when an exact match occurs in large multi-word terms.

All these representation methods shared a common preprocessing phase, where local abbreviation and acronyms were identified and expanded employing a slightly adapted version of the local abbreviation identification method described in [3]. This method [10] scans the input texts searching for *<short-form, long-form>* pair candidates, using several heuristics to identify the correct long forms in the ambiguous cases.

Table 1 summarizes the results obtained in our preliminary tests. To get the performance measures of the different configurations we have employed the BioASQ Project Oracle and as evaluation data we used the MEDLINE articles included in test set number 2 in the second batch of the 2014 edition of BioASQ challenge, which were removed from the training collection the three Lucene indexes were built from.

We have evaluated the three index term generation methods using different values for k, the number of similar articles to be used (1) in the estimation of the number of Check Tags and regular descriptors to be assigned and (2) in the set of vote procedures that will construct the final list of Check Tags and descriptors to attach to a given article. We have also evaluated the effect of two index term weighting methods available in version 4.10 of Apache Lucene: a classical *tf-idf* weighting scheme [9] and a more complex one inspired by the Okapi BM25 family of scoring formulae [8]. These weighting schemes are employed by the Lucene engine to compute the similarity scores used to create the ranking of documents relevant to a given query. In our case, the query terms are all of the index terms extracted from the article to be annotated using one of the methods described before.

As can be seen in table 1 and also in the results of our official BioASQ runs, the best results are obtained with stemming and lemmatization with very similar performance values in both cases. There was a marginal gain in flat measures in favor of stemming based representation and with the hierarchical measures in the case of lemmatization. The representation using multi-word terms extracted from noun phrase chunks had poor performance, probably because of the use of overlapping word trigrams. capabilities of our k-NN method and also in the scoring functions of Lucene engine. Very infrequent index terms can have the undesired effect of boosting internal scores in schemes where inverse document frequencies are taken into account.

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

Finally, regarding the effect of taking into account different number of nearest neighbors, the best results are obtained when using values of k around 20, which was the default value in our official runs in BioASQ challenge.

3 Candidate descriptors post-processing

In order to improve the results obtained by the Lucene based k-NN approach depicted in previous sections, we have evaluated several alternatives to try to get better annotation performance. We have followed two different lines of work to improve the prediction accuracy out k-NN based system.

The first weak point in the proposed k-NN based method is related with the fairly simple local decisions performed by our k-NN annotator, given that the performed generalization is just a weighted average and an inverse distance weighted vote. We have tested a couple of approaches employing more sophisticated decision making. In both cases a two-steps procedure is applied.

In a first step an expanded list with a larger amount of candidate Check Tags and candidate regular descriptors is created. Those expanded sets of descriptors will be filtered and refined during the second step. In order to add diversity to these expanded candidates lists, the size of both lists (expanded candidates Check Tags and expanded candidate regular descriptors) is twice the size previously predicted by the weighted average procedure described in section 2. Two methods were tested to perform the filtering step:

Training a per-article multilabel classifier. In this approach, after creating the expanded list of candidate Check Tags and the expanded list of regular descriptors for the MEDLINE article being annotated, two multilabel classifiers, on per expanded list, are trained. The label set for these classifiers are the two lists of expanded candidates, and the training instances comprises up to 1000 most similar articles extracted by the indexing engine. Once the training of both classifiers is completed, the contents of the article being annotated are used as input to those models in order to extract the final ranked list of Check Tags and the final list of regular descriptors, using the cut off limits identified by the weighted average estimator.

In our preliminary evaluation we have employed as multilabel categorization strategy a custom implementation of Classifier Chains [11], using as base classifiers instances of Support Vector Machines trained using the LibSVM project [2] tools. This evaluation was done with a reduced test set and the obtained results were slightly better than the basic k-NN, but still far from the most competitive teams in BioASQ challenge.

Unfortunately, we were unable to use this method effectively in our official runs of BioASQ challenge. Due to the time restrictions imposed in the challenge and the large training times required by this approach, we were unable to finish any submission on time.

Iterative k-NN vote. Instead of employing a multilabel classifier to support the second step we tested the use of another k-NN method backed by the same Lucene index to post-process the expanded lists of candidates.

For each candidate (both Check Tag or regular descriptor) in each expanded list a new query is sent to the index engine. Our index is queried using the representation of the article being annotated in order to get the list of similar articles which have among their respective extended candidate list the candidate descriptor being evaluated at this moment.

This new list of similar articles, with their normalized distances, is employed in a second voting process. In this case, similar articles where the candidate descriptor was actually assigned as a relevant descriptor are considered as positive votes. Whereas, similar articles where the candidate descriptor would have been a wrong assignment are treated as negative votes.

What this second step does with the extended candidate lists can be seen as a sort of "learning to discard" procedure. We are evaluating the actual usage of every candidate descriptor in a similar document which also had it as one of its own extended candidates. So, extended candidates that have not been considered as relevant descriptors in the weighted majority of similar documents retrieved during this second phase are discarded.

Although this approach imposes an extreme use of the Lucene index and implies large disk reading loads, we were able to make it suitable to fulfill the BioASQ challenge time restrictions.

Another weak point of our basic k-NN method when applied in the context of MeSH annotation is that it does not exploit the hierarchical information carried by the thesaurus structure, whose usage is explicitly described in official MeSH annotation guidelines. To try to overcome this limitation we evaluated the use of semantic similarity measures among MeSH descriptors as a method to expand and rearrange the ranked list of regular descriptor assigned by the basic k-NN method described in previous sections.

Descriptor expansion with hierarchical similarity measures. We have employed D. Lin's semantic similarity measure [7], a well known semantic measure suitable to capture and summarize in a number between 0 and 1 the proximity of two concepts belonging to a common concept taxonomy.

$$sim(s_i, s_j) = \frac{2 \times logP(LCA(s_i, s_j))}{logP(s_i) + logP(s_j)} \tag{1}$$

We have followed the original formula (1), where s_i and s_j are concepts in a taxonomy, $LCA(s_i, s_j)$ represents the lowest common ancestor of both concepts and $P(s_k)$ is an estimation of the probability assigned to concept s_k. In our case this probability is computed as the ratio between the number of MeSH descriptors belonging to the subtree rooted at descriptor s_k and the total number of descriptor in the MeSH thesaurus.

In our preliminary tests we applied Lin's measure in a very simple fashion. The ranked list of candidate regular descriptors returned by the basic k-NN based method is expanded adding all MeSH descriptors in a radio of 3 hops, according to the thesaurus hierarchical relationships. The score of those new added descriptors is computed by multiplying the score of the original candidate descriptor with the value of Lin's similarity between it and the added descriptor. For a given descriptor (original or expanded), combined scores coming from the expansion process of different initial candidate descriptors are accumulated.

Once the expanded list of descriptors is created and ranked according to the new scores, two simple heuristics derived from MeSH annotation guidelines [5] are employed to remove redundant annotations. These removal heuristics are applied iteratively and limited to a window of the top-most $n + 3$ descriptors, where n is the number of regular descriptors predicted by our k-NN based scheme.

- when tree or more siblings appear in the descriptor window, all of them are replaced by their common parent
- more specific descriptors (descendants) are preferred over more general ones (ancestors) occurring inside the considered window, and replace them

The surviving descriptors are cut off at the number of descriptor predicted by the weighted average predictor, using the combined scores to rank the list.

A priori this approach seemed to be a promising and effective way to add hierarchical information from the MeSH thesaurus to the k-NN prediction. However, the results we obtained were very disappointing, even worse than the vanilla k-NN approach, and lead us to not submit the results obtained with this method in our official runs.

4 Official BioASQ runs and discussion

Even we have tested several alternatives to try to improve the results obtained by the basic Lucene based k-NN method, only the most simple ones have been submitted to the official batches of BioASQ challenge. Our original objective was to try to approximate to the performance values obtained by the two NLM Medical Text Indexer (MTI) [10] baselines ("Default MTI" and "MTI First Line Indexer"), since this is the reference tool employed by MEDLINE indexers.

In table 2 the official performance measures obtained by our runs in the Test Batch number 3 are shown. The name of our runs (*"iria"*) originally stood for *Information Retrieval based Iterative Annotator* since the initial aim of this participation at BioASQ challenge was to evaluate different approaches to improve the initial ranked list of candidate descriptors retrieved by the indexing engine. The official runs sent by our group during our participation in the Test Batch number 3 were created using the following configurations.

iria1. Representation of MEDLINE articles using unigrams, bigrams and trigrams extracted from noun phrase chunks identified by means of Genia Tagger.

As described at the end of section 2.1 only articles from year 2000 onwards were indexed, discarding terms appearing in 5 or less abstracts and term used in more than 50% of total documents.

The predicted number of Check Tags and regular descriptors to be returned is increased a 10% in order to ensure slightly better values in recall related measures.

iria2. Representation of MEDLINE articles using terms extracted using standard English stop-words removal and stemming. All other parameter are identical to *iria1*.

iria3. Representation of MEDLINE articles using lemmas extracted with ClearNLP tools after PoS tag filtering. All other parameter are identical to *iria1*

iria4. Using the Lucene index created for *iria2* this set of runs employs the *Iterative k-NN vote* approach described in section 3, using a two step k-NN method.

iria-mix. This was a "control" set of runs employed to measure how close were our methods to MTI baselines.

In test sets 1,2,3 and 4 *iria-mix* was simply a weighted mix of our results in *iria-2* run with the MTI-DEF and MTI-FLI results distributed by BioASQ organization each week. Weight assigned to each one of these three lists was the respective official MiF values obtained in the previous week. Every descriptor in *iria-2*, MTI-DEF and MTI-FLI accumulates the weight of the descriptors list where it was included. The final list of descriptors is ranked according to these accumulated scores and the n top-most descriptors are returned as candidates, being n the number of Check Tags and regular descriptors originally predicted by *iria-2* run.

In test set 5, *iria-mix* used the Lucene index created for *iria-2* to test a different k-NN search. In this case, a more complex type of query to find similar documents was evaluated. This query was constituted by the index terms extracted from the abstract to be annotated, like in *iria-2* case, but it also included the descriptors assigned in the MTI-DEF results distributed by BioASQ organization that week. That is, in this case the similarity query searches for articles sharing index terms with the abstract being annotated and also with real MeSH descriptors included in the MTI-DEF prediction.

The results of our participation in the third edition of the BioASQ biomedical semantic indexing challenge are far from the results of the most competitive teams and our particular objective, try to reach performance levels similar to MTI baselines, was not achieved. As positive aspects of our participation, we have shown that k-NN methods backed by conventional textual indexers like Lucene are a viable alternative for this kind of large scale problems, with minimal computational requirements and not so bad results. We also have performed an exhaustive evaluation of the performance of several alternatives to index term extraction, ranging from simple ones, based on stemming rules, to more complex ones were natural language processing is required.

Our *a priori* main contribution, the proposed methods to improve initial k-NN predictions, has not obtained real performance improvements, except in the case of training a per-article multilabel classifier. More work needs to be done in this case and also in the use of taxonomy based similarity measures, like Lin's measure, since we still think that is a promising alternative to include hierarchical information on flat categorization approaches.

Acknowledgements

Research reported in this paper has been partially funded by ”Ministerio de Economía y Competitividad” and FEDER (under projects FFI2014-51978-C2-1 and TIN2013-42741-P) and by the Autonomous Government of Galicia (under projects R2014/029 and R2014/034).

References

1. 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.
2. 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
3. A.S. Schwartz, M.A. Hearst. Algorithm for Identifying Abbreviation Definitions in Biomedical Text. Pacific Symposium on Biocomputing 8:451-462(2003)
4. 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.
5. U.S. National Library of Medicine. MEDLINE Indexing Online Training Course. http://www.nlm.nih.gov/bsd/indexing/training *(online, 5th june, 2015)*
6. Yoshimasa Tsuruoka, Yuka Tateishi, Jin-Dong Kim, Tomoko Ohta, John. McNaught, Sophia Ananiadou, and Jun’ichi Tsujii. Developing a Robust Part-of-Speech Tagger for Biomedical Text, Advances in Informatics, 10th Panhellenic Conference on Informatics, LNCS 3746, pp. 382-392, 2005
7. Dekang Lin. An Information-Theoretic Definition of Similarity. Proceedings of the Fifteenth International Conference on Machine Learning (ICML 1998), Madison, Wisconsin, USA, July 24-27, 1998.
8. Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. Okapi at TREC-3. In Proceedings of the Third Text REtrieval Conference (TREC 1994). Gaithersburg, USA, November 1994.
9. Sparck Jones, K. A Statistical Interpretation of Term Specificity and Its Application in Retrieval. Journal of Documentation 28: 11–21. 1972
10. J.G. Mork, A. Jimeno Yepes, A.R. Aronson. The NLM Medical Text Indexer System for Indexing Biomedical Literature. 2013. http://ii.nlm.nih.gov/Publications/Papers/MTI_System_Description_Expanded_2013_Accessible.pdf *(online, 5th june, 2015)*
11. Jesse Read, Bernhard Pfahringer, Geoff Holmes and Eibe Frank. Classifier Chains for Multi-label Classification. Machine Learning Journal. Vol. 85(3), pp. 333–359. 2011.

Table 2. Official results for BioASQ batch 3.

week 1, labeled documents: 2530/3902																		
	flat											hier.						
system	rank	MiF	EBP	EBR	EBF	MaP	MaR	MaF	MiP	MiR	Acc.	rank	LCA-F	HiP	HiR	HiF	LCA-P	LCA-R
best	1/35	0.6320	0.6910	0.6041	0.6247	0.6430	0.5025	0.5000	0.6909	0.5824	0.4693	1/35	0.5181	0.8091	0.7081	0.7316	0.5773	0.4978
def. MTI	13/35	0.5805	0.6002	0.5836	0.5732	0.5536	0.5292	0.4962	0.5957	0.5661	0.4164	13/35	0.4916	0.7546	0.7107	0.7098	0.5265	0.4891
iria-2	19/35	0.4869	0.4275	0.5756	0.4780	0.3961	0.4346	0.3853	0.4311	0.5593	0.3260	19/35	0.4306	0.6033	0.7301	0.6430	0.4031	0.4896
iria-3	20/35	0.4868	0.4256	0.5770	0.4773	0.3926	0.4302	0.3796	0.4295	0.5618	0.3253	20/35	0.4297	0.6002	0.7343	0.6428	0.4007	0.4919
iria-1	21/35	0.4727	0.5024	0.4695	0.4673	0.4113	0.3096	0.3014	0.5024	0.4463	0.3184	21/35	0.4149	0.6814	0.6042	0.6150	0.4612	0.4045
iria-4	23/35	0.4164	0.3730	0.5038	0.4117	0.2738	0.4065	0.3435	0.3617	0.4905	0.2699	22/35	0.3887	0.5460	0.7075	0.5942	0.3574	0.4611
iria-mix	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

week 2, labeled documents: 2256/4027																		
	flat											hier.						
system	rank	MiF	EBP	EBR	EBF	MaP	MaR	MaF	MiP	MiR	Acc.	rank	LCA-F	HiP	HiR	HiF	LCA-P	LCA-R
best	1/39	0.6397	0.6847	0.6222	0.6331	0.6284	0.5144	0.5060	0.6820	0.6023	0.4783	1/29	0.5250	0.7960	0.7172	0.7318	0.5745	0.5127
def. MTI	18/39	0.5822	0.6056	0.5842	0.5743	0.5452	0.5128	0.4792	0.6002	0.5653	0.4184	18/39	0.4914	0.7464	0.7039	0.6997	0.5288	0.4895
iria-mix	20/39	0.5730	0.5527	0.6057	0.5636	0.5125	0.5315	0.4854	0.5617	0.5847	0.4061	19/39	0.4862	0.6968	0.7392	0.6977	0.4919	0.5076
iria-2	25/39	0.4922	0.4442	0.5636	0.4833	0.4056	0.4070	0.3693	0.4490	0.5446	0.3310	25/39	0.4330	0.6136	0.7100	0.6381	0.4145	0.4812
iria-3	26/39	0.4871	0.4256	0.5788	0.4776	0.3855	0.4199	0.3723	0.4301	0.5614	0.3257	26/39	0.4296	0.5948	0.7282	0.6353	0.4000	0.4923
iria-4	27/39	0.4700	0.5675	0.4235	0.4635	0.4271	0.3147	0.3089	0.5588	0.4056	0.3167	27/39	0.3988	0.7053	0.5484	0.5853	0.4814	0.3681
iria-1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

week 3, labeled documents: 1519/3162																		
	flat											hier.						
system	rank	MiF	EBP	EBR	EBF	MaP	MaR	MaF	MiP	MiR	Acc.	rank	LCA-F	HiP	HiR	HiF	LCA-P	LCA-R
best	1/42	0.6496	0.6919	0.6313	0.6420	0.6429	0.5293	0.5228	0.6892	0.6144	0.4875	1/42	0.5363	0.8082	0.7266	0.7439	0.5850	0.5235
def. MTI	17/42	0.5970	0.6202	0.5994	0.5897	0.5644	0.5346	0.5049	0.6123	0.5824	0.4329	15/42	0.5039	0.7651	0.7249	0.7202	0.5407	0.5029
iria-mix	20/42	0.5826	0.5609	0.6151	0.5727	0.5264	0.5466	0.5049	0.5679	0.5981	0.4147	17/42	0.4966	0.7098	0.7529	0.7115	0.4995	0.5205
iria-2	24/42	0.5011	0.4524	0.5726	0.4927	0.4163	0.4122	0.3771	0.4557	0.5566	0.3394	24/42	0.4396	0.6229	0.7226	0.6501	0.4218	0.4861
iria-3	27/42	0.4894	0.4277	0.5814	0.4806	0.3965	0.4214	0.3779	0.4309	0.5662	0.3283	27/42	0.4331	0.5965	0.7355	0.6402	0.4029	0.4964
iria-4	28/42	0.4868	0.7394	0.3754	0.4771	0.6789	0.2560	0.2733	0.7408	0.3625	0.3285	30/42	0.3874	0.8561	0.4581	0.5674	0.5832	0.3095
iria-1	29/42	0.4811	0.4359	0.5455	0.4721	0.3978	0.3817	0.3515	0.4402	0.5304	0.3217	28/42	0.4242	0.6094	0.6978	0.6314	0.4095	0.4667

week 4, labeled documents: 1097/3621																		
	flat											hier.						
system	rank	MiF	EBP	EBR	EBF	MaP	MaR	MaF	MiP	MiR	Acc.	rank	LCA-F	HiP	HiR	HiF	LCA-P	LCA-R
best	1/40	0.6190	0.6758	0.5961	0.6139	0.6272	0.5108	0.5024	0.6716	0.5739	0.4577	1/40	0.5128	0.8045	0.6998	0.7259	0.5657	0.4963
def. MTI	17/40	0.5662	0.5959	0.5674	0.5612	0.5422	0.5129	0.4830	0.5875	0.5464	0.4049	16/40	0.4854	0.7586	0.6947	0.7024	0.5247	0.4807
iria-mix	19/40	0.5577	0.5487	0.5828	0.5509	0.5169	0.5190	0.4823	0.5543	0.5610	0.3940	18/40	0.4817	0.7149	0.7262	0.7019	0.4940	0.4956
iria-3	23/40	0.4837	0.4390	0.5468	0.4745	0.4065	0.4146	0.3772	0.4425	0.5334	0.3232	24/40	0.4304	0.6254	0.7044	0.6442	0.4154	0.4725
iria-2	24/40	0.4831	0.4397	0.5461	0.4746	0.4065	0.4122	0.3760	0.4433	0.5308	0.3232	23/40	0.4305	0.6303	0.7044	0.6472	0.4158	0.4715
iria-1	25/40	0.4647	0.4263	0.5201	0.4559	0.3942	0.3893	0.3582	0.4297	0.5059	0.3075	25/40	0.4170	0.6186	0.6797	0.6282	0.4073	0.4511
iria-4	26/40	0.4453	0.4757	0.4468	0.4401	0.3477	0.3476	0.3258	0.4625	0.4293	0.2952	26/40	0.3954	0.6440	0.6124	0.6006	0.4229	0.4022

week 5, labeled documents: 896/3842																		
	flat											hier.						
system	rank	MiF	EBP	EBR	EBF	MaP	MaR	MaF	MiP	MiR	Acc.	rank	LCA-F	HiP	HiR	HiF	LCA-P	LCA-R
best	1/43	0.6512	0.6861	0.6405	0.6444	0.6326	0.5564	0.5464	0.6822	0.6229	0.4893	1/43	0.5367	0.8015	0.7363	0.7461	0.5759	0.5305
def. MTI	19/43	0.5985	0.6121	0.6079	0.5908	0.5519	0.5649	0.5313	0.6048	0.5922	0.4342	18/43	0.5069	0.7607	0.7352	0.7238	0.5353	0.5104
iria-2	23/43	0.5221	0.5369	0.5190	0.5121	0.5080	0.3811	0.3738	0.5451	0.5009	0.3588	25/43	0.4465	0.7019	0.6467	0.6510	0.4830	0.4399
iria-3	24/43	0.5217	0.5353	0.5186	0.5114	0.5017	0.3760	0.3688	0.5438	0.5013	0.3575	24/43	0.4469	0.7098	0.6501	0.6563	0.4824	0.4414
iria-mix	25/43	0.5134	0.4552	0.6033	0.5052	0.4134	0.4804	0.4370	0.4576	0.5847	0.3508	23/43	0.4494	0.6223	0.7480	0.6605	0.4217	0.5101
iria-1	26/43	0.4905	0.4348	0.5761	0.4824	0.3918	0.4385	0.3977	0.4367	0.5595	0.3312	26/43	0.4337	0.6045	0.7270	0.6412	0.4070	0.4927
iria-4	27/43	0.4834	0.5125	0.4860	0.4794	0.3620	0.3839	0.3585	0.5002	0.4678	0.3297	27/43	0.4177	0.6506	0.6301	0.6135	0.4449	0.4249