Intelligent Information Management
Targeted Competition Framework
ICT-2011.4.4(d)

Project **FP7-318652 / BioASQ**

Deliverable **D5.1**

Distribution **Public**

http://www.bioasq.org

Technology Overview Report 1

Ioannis Partalas and Eric Gaussier

Status: Final-revised (Version 1.1)

October 2013

Project

Project ref.no.	FP7-318652
Project acronym	BioASQ
Project full title	A challenge on large-scale biomedical semantic indexing and question answering
Porject site	http://www.bioasq.org
Project start	October 2012
Project duration	2 years
EC Project Officer	Martina Eydner

Deliverable

Deliverabe type	Report
Distribution level	Public
Deliverable Number	D5.1
Deliverable title	Technology Overview Report 1
Contractual date of delivery	M13 (October 2013)
Actual date of delivery	October 2013
Relevant Task(s)	WP5/Task 5.1
Partner Responsible	UJF
Other contributors	
Number of pages	27
Author(s)	Ioannis Partalas and Eric Gaussier
Internal Reviewers	Sergios Petridis
Status & version	Final-revised
Keywords	BioASQ, technology overview, results analysis

Executive Summary

This deliverable reviews the systems that participated during the first BIOASQ challenge and performs an analysis of the results. More specifically, in the deliverable a short description of each system is given providing also the key technologies that have been used. The objective of this deliverable is to identify the most promising approaches and to point out the progress made with the state-of-the-art.

The challenge comprised two tasks: a) large-scale online biomedical indexing (Task 1a) and b) introductory biomedical semantic QA (Task 1b). Both tasks run in three consecutive batches.

In Task 1a 11 teams participated using 46 registered systems. The systems were evaluated in several performance measures and compared against two baseline systems. Most of them were able to cope with the large scale of the problem while two of them achieved to systematically outperform the state-of-the-art baseline (Medical Text Indexer). A variety of methods have been used like machine learning approaches or search-based ones and hierarchical or flat ones.

In Task 1b 4 teams participated in total in the two phases. In phase A 4 systems were submitted while in phase B the teams submitted 7 systems. In phase A the systems have not achieved better results than the baselines while in phase B the were able to obtain a superior performance.

Contents

List of Figures

List of Tables

1

Introduction

This deliverable reviews the systems that participated during the first BIOASQ challenge and performs an analysis of the results. More specifically, in the deliverable a short description of each system is given providing also the key technologies that have been used. The objective of this deliverable is to identify the most promising approaches and to point out the progress made with the state-of-the-art.

The reminder of the deliverable is as follows:

- Chapter 1 describes briefly the BIOASQ challenge providing also details of the evaluation procedure along with the corresponding time plans. Additionally, for each of the two tasks of the challenge, the total numbers of the participating systems and teams are reported.
- Chapter 2 reviews, for the two tasks, the systems that participated in the challenge. This review is based on the available descriptions provided by the participants. For each system, we present the key points of the proposed methods.
- Chapter 3 presents the results of the evaluation procedure available from the BIOASQ evaluation platform[1].
- Chapter 4 presents the prizes awarded to the winners of each task.
- Chapter 5 concludes this deliverable by commenting on the advancement of the state-of-the-art in the biomedical semantic indexing and question answering domain. Also, it discusses the potential impact of the technologies on specialized search engines.

1.1 Challenge Description

The goal of the BIOASQ challenge is to push the state-of-the-art technologies in the domain of biomedical semantic indexing and question answering (Tsatsaronis et al., 2012). The challenge comprises two tasks: a) a large-scale semantic indexing task (Task 1a) and b) a question answering task (Task 1b).

[1] http://bioasq.lip6.fr

Large-scale online biomedical semantic indexing. In Task 1a the goal is to classify documents from the PubMed[2] digital library, which indexes biomedical articles form MEDLINE, onto concepts of the MeSH[3] hierarchy. On a daily basis, new articles which are not yet annotated are stored in the database of PubMed. These articles are used as test sets for the evaluation of the participating systems. As soon as the annotations are available from the PubMed curators, the performance of each system is calculated using standard evaluation measures as well as new ones partly developed during the project.

In order to provide an on-line and large-scale scenario, the task were divided in three independent batches, where in each batch 6 test sets of biomedical articles were released consecutively. Each of these test sets were released on a weekly basis and the participants had 23 hours to provide their answers. Figure 1.1 presents the time plan of Task 1a.

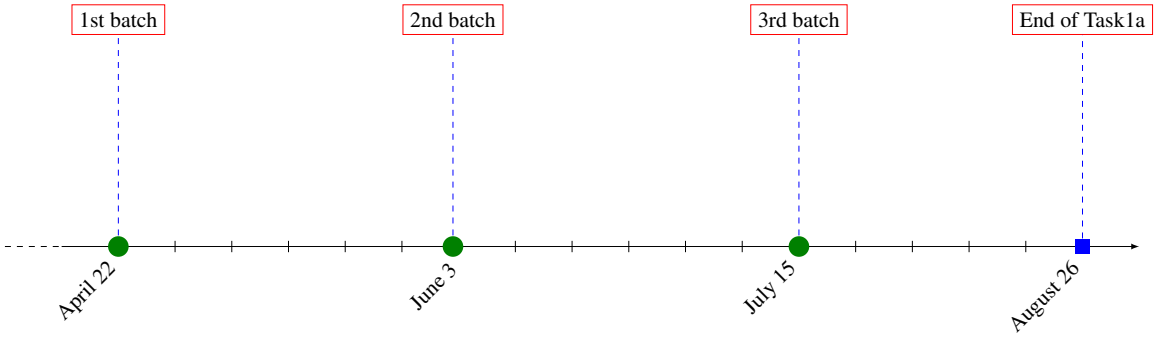

Figure 1.1: The time plan of Task 1a.

The participants were evaluated using several flat and hierarchical measures. The winners of each batch were decided based on their performance in the Micro F-measure (MiF) from the family of flat measures (Tsoumakas et al., 2010), and the Lowest Common Ancestor F-measure (LCA-F) from the family of hierarchical measures (Kosmopoulos et al., 2013). For completeness, several other flat and hierarchical measures were reported (Balikas et al., 2013).

Introductory biomedical semantic QA. Task 1b comprised two phases and the goal was to provide a large-scale question answering challenge where the systems should be able to cope with all the stages of a question answering task (as the retrieval of relevant concepts and articles) and provide natural language answers.

During phase A of the task, BIOASQ released questions in English from the benchmark datasets and the participants had to respond with concepts (from specific terminologies and ontologies), snippets extracted from the retrieved articles and RDF triples (from specific ontologies).

In the second phase of the task, the released questions contained also the correct answers for the elements (concepts, articles, snippets and RDF triples) of the first phase. The participants had to answer with *exact* answers (this varies according to the type of the question) and *ideal* answers which are paragraph-sized summaries in natural language.

The task has been split in three independent batches, as in Task 1a. The two phases for each batch were run with a time gap of 24 hours and for each of them the participants had 24 hours to submit their answers.

For evaluating the performance of the systems in phase A, well established information retrieval measures have been used as the mean precision, mean recall, mean F-measure, mean average precision

[2] http://www.ncbi.nlm.nih.gov/pubmed
[3] http://www.ncbi.nlm.nih.gov/mesh

(MAP) and geometric MAP (GMAP). The winners were selected based on MAP. The phase B of the task has been based on a manual evaluation of the ideal answers provided by the systems, by the BIOASQ experts. For reasons of completeness, automatic evaluation measures have been also reported using ROUGE (Lin, 2004).

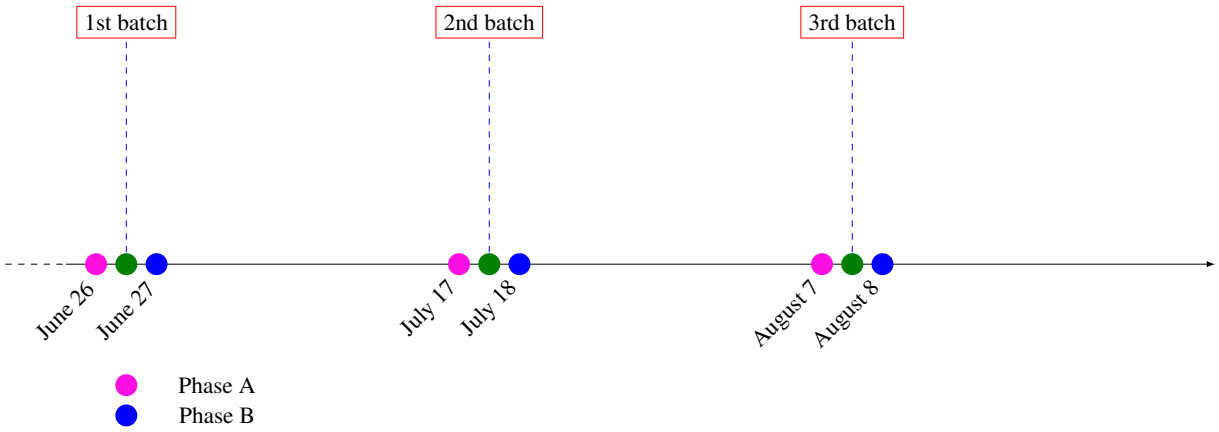

Figure 1.2: The time plan of Task 1b. The two phases for each batch run in consecutive dates.

2

Technology Overview

2.1 Task 1a

2.1.1 Background and Related Work

Background. Task 1a deals with the semantic indexing of biomedical documents with concepts from the MeSH hierarchy. Typically the problem is tackled like a classification one where one should build classification models that assign classes from the designated hierarchy to documents. Under this setting, the training set can be represented by $S = \{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{m}$. In the context of text classification, $\boldsymbol{x}^{(i)} \in \mathcal{X}$ denotes the vector representation of the i-th document in the input space $\mathcal{X} \subseteq \mathbb{R}^n$. Assuming that there are K classes denoted by the set $\mathcal{Y} = \{y_1 \dots y_K\}$, the label $y^{(i)} \in \mathcal{Y}$ represents the class associated with the instance $\boldsymbol{x}^{(i)}$. In text classification the features (or terms) of the vector representation are the distinct words that occur in the training data. Each element x_k of the vector representation can be either a binary value (0/1), expressing the absence or the presence of the specific word in the document, or a real value calculated by statistical techniques. A simple approach (term frequency) is to calculate the number of occurrences of each word in the document. The most popular scheme is the $tf * idf$ (term-frequency inverse document frequency) where the tf is the term frequency of a specific term t and $idf = \ln \frac{m}{df_t}$ is the logarithm of the number of the documents in the collection divided by the number of documents that contain the term. The idf is a measure of the importance of a specific term in the collection. For example, very common words will have a low idf value. A standard chain for producing the vectors is the following: tokenization, stemming/lemmatization and stop-word removal.

Related work. There have been proposed several approaches for large-scale classification which either leverage the hierarchy information (a simple tree hierarchy is presented in Figure 2.1) by taking into account the parent-child relations among the classes (*hierarchical methods*) or they totally ignore this information (*flat measures*). Hierarchical methods suffer from the fact that the errors made at an upper level of the hierarchy are unrecoverable. On the other hand, flat methods are very slow in terms of training and testing compared to hierarchical methods (Babbar et al., 2013).

Some of the earlier works on exploiting hierarchy among target classes for the purpose of text classification has been studied in (Koller and Sahami, 1997). Parameter smoothing for Naive Bayes classifier

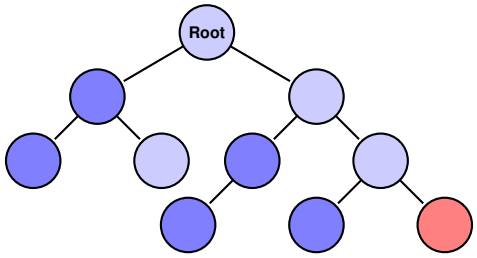

Figure 2.1: A simple tree hierarchy.

along the root to leaf path was explored by (McCallum et al., 1998). Maximum margin based approaches have been proposed in (Cai and Hofmann, 2004; Dekel et al., 2004), where the degree of penalization in mis-classification depends on the distance between the true and predicted class in the hierarchy tree. However, these approaches were applied to the datasets in which the number of categories were limited to a few hundreds. (Liu et al., 2005) applied hierarchical SVM to the scale with over 100,000 categories in Yahoo! directory. More recently, other techniques for large scale hierarchical text classification have been proposed. Prevention of error propagation by applying *Refined Experts* trained on a validation was proposed in (Bennett and Nguyen, 2009). In this approach, bottom-up information propagation is performed by utilizing the output of the lower level classifiers in order to improve the classification of top-level classifiers. Deep Classification (Xue et al., 2008) proposes hierarchy pruning to first identify a much smaller subset of target classes. Prediction of a test instance is then performed by re-training Naive Bayes classifier on the subset of target classes identified from the first step. More recently, Bayesian modelling of large scale hierarchical classification has been proposed in (Gopal et al., 2012) in which hierarchical dependencies between the parent-child nodes are modelled by centering the prior of the child node at the parameter values of its parent.

Hierarchy simplification by flattening entire layer in the hierarchy has been studied from an empirical view-point in (Wang and Lu, 2010; Malik, 2009). These strategies for taxonomy adaptation by flattening do not provide any theoretical justification for applying this procedure. Moreover, they offer no clear guidelines regarding which layer in the hierarchy one should flatten. Most of the existing approaches to large scale classification have focussed on the two extremes of flat or hierarchical classification. An approach based on taxonomy embedding has been proposed in (Weinberger and Chapelle, 2009), but this has been restricted to only small scale problems, wherein the target classes are of the order of few hundreds.

Apart from accuracy, other important factors while evaluating the classification strategies for large scale classification are training and prediction speed. The comparison of training time complexity for flat and hierarchical classification in the context of large taxonomies has been studied in (Liu et al., 2005). Learning the hierarchy tree from large number of classes in order to make faster prediction has also attained significance as explored in the recent works such as (Bengio et al., 2010; Beygelzimer et al., 2009; Gao and Koller, 2011). The aim in these approaches is to achieve better prediction speed while maintaining the same classification accuracy as flat classification. On the other end of the specturm are flat classification techniques such as employed in (Perronnin et al., 2012) which ignore the hierarchy structure. These strategies are likely to perform well for balanced hierarchies with sufficient training instances per target class and not so well in large scale taxonomies which suffer from the problem of rare classes.

2.1.2 Systems Overview

The participating systems in the semantic indexing task of the BIOASQ challenge adopted a variety of approaches, like hierarchical and flat methods or search-based systems that rely on information retrieval techniques. In the rest of this section we describe the proposed systems by stressing on their key points.

In (Ribadas et al., 2013) the authors proposed two hierarchical approaches. The first approach, referred to as *Hierarchical Annotation and Categorization Engine* (HACE), follows a top-down hierarchical classification scheme (Silla and Freitas, 2011) where, for each node of the hierarchy, a binary classifier is trained. For constructing the positive training examples for each node, the authors employ a random method that selects a fixed amount of examples from the descendants of the current node and a method that is based on k-means to choose the k closest examples to the centroid of the node. In both approaches the selected examples are fixed in order to create manageable datasets especially in the upper levels of the hierarchy. The second system (*Rebayct*) that has participated in the challenge was based on a Bayesian network which models the hierarchical relations as well as the training data (that is the terms in the abstracts ant titles). A major drawback of this system is that it cannot scale well to large classification problems with thousands of classes and millions of documents. For this reason, the authors reduced drastically the training data keeping only 10% of the data split in 5 disjoint parts in order to train five different models. During the testing phase, the models are aggregated through a simple majority voting.

In (Tsoumakas et al., 2013) (*AUTH*) a flat classification approach has been employed which trains a binary SVM for each label in the training data (Tang et al., 2009). In order to reduce the complexity of the problem the authors kept only the training data that belong to the journals (1806 in total) from which the test sets were sampled during the testing phase of the challenge. The journal filtering reduced the training data to approximately 4 millions of documents from the initial 11 millions documents. The features that were used to represent each article were unigrams and bigrams (word as unit) extracted from the title and abstract of each article. The systems that were introduced in the challenge use a meta-model (called MetaLabeler (Tang et al., 2009)) for predicting the number of labels (N) of a test instance. During the prediction all the SVM classifiers are queried and the labels are sorted according to the corresponding confidence value. Finally, the system predicts the N top labels. While the proposed approach is relative simple, it requires processing power for both the training and the testing procedure. Furthermore, it has large storage requirements (as reported from the authors, the size of the models for one of the systems was 406Gb).

In (Zhu et al., 2013), the authors follow two different approaches: a) one that relies in the results provided by the MetMap tool described in Aronson and Lang (2010) and b) one that is based on the search engine Indri[1]. In the MetaMap based approach, for each test instance, the MetaMap system is queried for both the title and the abstract of the article. The returned results contain concepts and their corresponding confidence scores. The system calculates a final score, weighting differently the concepts that are obtained for the title and the abstract and filtering the ones exceeding a predefined threshold for the confidence score. Finally, the system proposes the m top-ranked concepts, where m is a free parameter. In the search based approach the authors index the training data using the engine Indri. For each test article a query q is formed and a score is calculated for each document d in the index. The concepts of the m top-ranked documents are assigned to the test article.

In the *Wishart* system (Liu, 2013) a typical flat classification approach as well as a k-NN are used. In the flat approach, a binary SVM is trained for each label present in the training data using as features unigrams, bigrams and trigrams extracted from the abstracts of the training data. In the k-NN based approach, for each test article, the k-NN method is invoked in order to retrieve documents from a local

[1] http://www.lemurproject.org/indri.php

index. Additionally, the NCBI Entrez system is queried in order to retrieve extra documents along with their labels. All the abstracts are ordered (first N - empirically set to 100) according to their distance and the top M (empirically set to 10) labels are retained. For the final prediction, the two systems are combined by keeping the common predicted labels; the other labels are ordered according to their confidence scores. The system predicts 10-15 labels for each test article.

A learning to rank method has been used in the NCBI team (Mao and Lu, 2013). More specifically, the systems follow a three stage approach: i) first the k-nearest neighbours of the test article are retrieved from the MEDLINE database, ii) next the labels are ordered using a learning to rank algorithm and iii) finally a cut-off method prunes the ordered list. It is interesting to note that in the definition of the features for the learning to rank problem, the authors use the results of the MTIFL baseline system (see next paragraph). More specifically, a binary feature indicates whether a specific label is observed in the results of MTIFL.

Table 2.1 summarizes the main technologies that were employed by the participating systems; it also indicates whether a hierarchical or a flat approach has been followed. Additionally, the last column shows what features were used from each team for the representation of the documents. It is clear that the majority of the participants followed flat methods to tackle the problem using a variety of technologies from the machine learning and information retrieval areas. Not surprisingly, the machine learning approaches used SVM classifiers which are powerful schemes in text classification tasks (Tsoumakas et al., 2013; Liu, 2013). In the contrary, these flat systems have large processing and storage requirements in both training and inference stages. In order to reduce the complexity of the problem in (Ribadas et al., 2013), the authors leveraged the hierarchy information by employing the classifiers in a top-down manner. In (Zhu et al., 2013) and (Mao and Lu, 2013) the authors follow a two stage approach, thus reducing the complexity, where they first retrieve relevant articles using search engines or following a k-nearest neighbors approach on local indexes of the training data.

Reference	Approach	Technologies	Features
Tsoumakas et al. (2013)	flat	SVMs, MetaLabeler (Tang et al., 2009)	unigrams, bigrams
Ribadas et al. (2013)	hierarchical	SVMs, Bayes networks	unigrams, bigrams
Zhu et al. (2013)	flat	MetaMap (Aronson and Lang, 2010), information retrieval, search engines	unigrams
Liu (2013)	flat	k-NN, SVMs	unigrams, bigrams, trigrams
Mao and Lu (2013)	flat	k-NN, learning-to-rank	unigrams

Table 2.1: Technologies used in Task 1a from the participating systems along with the feature representation of the documents.

Baselines. During the first challenge, two systems were used as baseline systems. The first one, called BioASQ_Baseline, follows an unsupervised approach to tackle the problem; it is thus expected that the systems developed by the participants will outperform it. More specifically, the baseline implements Attribute Alignment Annotator (Doms, 2010). It is an unsupervised method, based on the Smith-Waterman sequence alignment algorithm (Smith and Waterman, 1981) and can recognizes terms from MeSH and Gene Ontology in a given text passage. The annotator first pre-processes both the ontology terms and the text by tokenizing them, removing the stop words and stemming the remaining terms (an in-house stop word list that is specific to the domain is used). Then the term stems are mapped onto the text stems

using the local sequence alignment algorithms (Smith and Waterman, 1981). Insertions, deletions and gaps are penalized. The information value of terms calculated over the whole ontology is also taken into account during the alignment process, in a similar manner as the inverse document frequency score is used for the tf-idf weighting of terms.

The second baseline is a state-of-the-art method called Medical Text Indexer (Mork et al., 2013) developed by the National Library of Medicine[2]. It is a classification system for articles of MEDLINE. MTI is used by curators in their annotation process. It is worth to note also that MTI is used in a few journals to fully automate the annotation process. So, it is expected to be a strong baseline.

2.2 Task 1b

In the second task of the BioASQ challenge a total of three teams participated in both phases with 11 systems. Only two descriptions were available from these systems (Liu, 2013; Zhu et al., 2013).

For phase A of Task 1b the Wishart system (Liu, 2013) makes use of query processing and document ranking techniques. More specifically, each test question in natural language form is converted by extracting the noun phrases and reference them using a thesaurus of biomedical entities. Then the question is expanded by adding synonyms and relevant biomedical entities using the PolySearch tool[3]. The entities found by PolySearch are used to rank the retrieved set of concepts, articles, triples and snippets. In phase B of the task a similar approach to phase A is used in order to augment the set of given concepts. Extracted sentences from the retrieved documents are ranked according to the cosine similarity with respect to the augmented concepts. The top-ranked sentences are concatenated in order to provide an *ideal* answer.

The MCTeam system participated (Zhu et al., 2013) only in phase A. In order to form an appropriate query the system first uses the test question to query MetaMap, which responds with concept-related words. These words were used to form a query. In case where no concepts were returned by MetaMap, the final query formed by removing the stopwords from the test question. This query was used to retrieve the appropriate information from the BIOASQ web services and also from a local index of PubMed full-text articles[4]. The two lists of the retrieved results were then merged and formed the final results.

Baselines. Two baselines were used in phase A, respectively returning the list of the top-50 and the top-100 entities that may be retrieved using the keywords of the input question as a query to the BIOASQ services. As a result, two lists for each of the main entities (concepts, documents, snippets, triples) are produced, of a maximum length of 50 and 100 items respectively.

For the creation of a baseline approach in Phase B, three approaches were created that address respectively the answering of factoid and lists questions, summary questions, and yes/no questions (Weissenborn et al., 2013). The three approaches were combined into one system, and constitute the BIOASQ baseline for this phase of Task 1B. The baseline approach for the list/factoid questions utilizes and ensembles a set of scoring schemes that attempt to prioritize the concepts that answer the question by assuming that the type of the answer aligns with the lexical answer type (type coercion). The baseline approach for the summary questions uses a multi-document summarization method using Integer Linear Programming and Support Vector Regression.

[2] http://ii.nlm.nih.gov/MTI/index.shtml
[3] http://wishart.biology.ualberta.ca/polysearch/
[4] The Indri search engine has been used for indexing the documents.

3

Setup and Results

3.1 Task 1a

3.1.1 Data and Setup

During the evaluation phase of Task1a, the participants were submitting each week their results in the online evaluation platform of the challenge[1]. The evaluation period was divided in three batches containing 6 test sets each. 11 teams participated in the task with a total of 40 systems. For measuring the classification performance of the systems, several evaluation measures were used (Balikas et al., 2013). The micro F-measure (MiF) and the Lowest Common Ancestor F-measure (LCA-F) were used to assess the systems and choose the winners for each batch (Kosmopoulos et al., 2013).

Table 3.1 presents the statistics of the training data that were provided to the participants while Table 3.2 presents the number of articles in each test set of each batch of the challenge along with the number of articles that had been annotated sofar. The articles were provided to the participants in their raw format (plain text) as well as in a pre-processed one (in a vectorized format) under the Apache Lucene framework[2]. Lucene is an open-source library[3] dedicated to text search. Figure 3.1 presents an example of two articles extracted from the BIOASQ benchmark training data.

Articles	10,876,004
Total labels	26,563
Labels per article	12.55
Size in GB	22

Table 3.1: Properties of the training data for Task1a: total number of articles, labels present in the data, average labels per article and size of the data in GB.

[1] http://bioasq.lip6.fr
[2] http://lucene.apache.org/
[3] Under the Apache Licence: http://www.apache.org/licenses/LICENSE-2.0.html

```
{
    "abstractText":"From the above it is seen that the [...]
    scientific guidance of which lies wholly
     in the hands of scientists.",
    "journal":"Science (New York, N.Y.)",
    "meshMajor":["Biomedical Research"],
    "pmid":"17772322",
    "title":"New Horizons in Medical Research.",
    "year":"1946"
},
{
    "abstractText":"1. T antigens of group A hemolytic
     streptococci have been [...] T antigen in the intact
      streptococcus from which it was derived.",
    "journal":"The Journal of experimental medicine",
    "meshMajor":["Antibodies","Antigens",
    "Immunity","Streptococcal Infections","Streptococcus"],
    "pmid":"19871581",
    "title":"THE PROPERTIES OF T ANTIGENS EXTRACTED
    FROM GROUP A HEMOLYTIC STREPTOCOCCI.",
    "year":"1946"
}
```


Figure 3.1: An extract from the training data of Task1a.

3.1.2 Results

Table 3.3 presents the correspondence of the systems for which a description was available and the submitted systems in Task 1a. The systems MTIFL, MTI and bioasq_baseline were the baseline systems used throughout the challenge. MTIFL and MTI refer to the NLM Medical Text Indexer system (Mork et al., 2013). Systems that participated in less than 4 test sets in each batch are not reported in the results[4].

According to (Demsar, 2006) the appropriate way to compare multiple classification systems over multiple datasets is based on their average rank across all the datasets. On each dataset the system with the best performance gets rank 1.0, the second best rank 2.0 and so on. In case that two or more systems tie, they all receive the average rank.

Tables 3.4 presents the average rank (according to MiF and LCA-F) of each system over all the test sets for the corresponding batches. Note, that the average ranks are calculated for the 4 best results of each system in the batch according to the rules of the challenge[5]. The best ranked system is highlighted with bold typeface. We can observe that during the first batch the MTIFL baseline achieved the best performance in terms of MiF measure, exhibiting a state-of-the-art performance which is also evident in the other two batches. During the first batch, RMAIP and system3 have the best performances in both measures. Interestingly, the ranking of the RMAIP according to the LCA-F measure is better than the

[4] According to the rules of BioASQ, each system had to participate in at least 4 test sets of a batch in order to be eligible for the prizes.

[5] http://bioasq.lip6.fr/general_information/Task1a/

Batch	Articles	Annotated Articles	Labels per article
1	1,942	1,543	10.00
	845	701	11.56
	793	706	10.87
	2,408	586	10.27
	6,742	4,194	11.70
	4,556	2,503	11.67
Subtotal	17,286	10,233	11.01
2	5,012	1,658	12.39
	5,590	1,658	11.48
	7,349	2,100	12.93
	4,674	1,552	12.37
	8,254	2,556	12.18
	8,626	2,284	13.20
Subtotal	39,505	11,808	12.42
3	7,650	2,002	12.58
	10,233	2,880	13.07
	8,861	2,274	12.44
	1,986	1,118	10.81
	1,750	1,024	10.70
	1,357	530	11.14
Subtotal	31,792	9,828	11.79
Total	88,628	31,869	12.01

Table 3.2: Statistics on the test datasets of Task1a.

Reference	Systems
Tsoumakas et al. (2013)	system1, system2, system3, system4, system5
Ribadas et al. (2013)	cole_hce1, cole_hce2, utai_rebayct, utai_rebayct_2
Zhu et al. (2013)	mc1, mc2, mc3, mc4, mc5
Liu (2013)	Wishart-*
Mao and Lu (2013)	RMAI, RMAIP, RMAIR, RMAIN, RMAIA
Baselines	MTIFL, MTI, bioasq_baseline

Table 3.3: Correspondence of reference and submitted systems for Task1a.

one based on MiF which shows that RMAIP is able to give answers in the neighborhood (as designated by the hierarchical relations among the classes) of the correct ones.

In the other two batches the systems proposed in (Tsoumakas et al., 2013) ranked as the best performed ones occupying the first two places (system3 and system2 for the second batch and system1 and system 2 for the third batch). Recall, that these systems follow a simple machine learning approach which uses SVMs and the problem is treated as flat.

We note here the good performance of the learning to rank systems (RMAI, RMAIP, RMAIR, RMAIN, RMAIA). We are not aware of similar attempts with learning to rank approaches in rel-

evant large scale classification challenges like the LSHTC challenge series (http://lshtc.iit.demokritos.gr/). Learning to rank methods are usually used in information retrieval tasks for ranking the retrieved results.

According to the available descriptions, the only systems that made use of the MeSH hierarchy were the ones introduced by (Ribadas et al., 2013). The top-down hierarchical systems, cole_hce1 and cole_hce2, achieved mediocre results while the utai_rebayct systems had poor performances. For the systems based on a Bayesian network, this behavior was expected as they cannot scale well to large problems. On the other hand the question that arises is whether the use of the MeSH hierarchy can be helpful for classification systems as the labels that are assigned by the curators to the PubMed articles do not follow the rule of the most specialized label. That is, an article may have been assigned a specific label in a deeper level of the hierarchy and in the same time a label in the upper hierarchy that is ancestor of the most specific one. In this case the system that predicted the more specific label will be punished by the flat evaluation measures for not predicting the most general label, which is implied by the hierarchical relations.

System	Batch 1		Batch 2		Batch 3	
	MiF	LCA-F	MiF	LCA-F	MiF	LCA-F
MTIFL	**1.25**	**1.75**	2.75	2.75	4.0	4.0
system3	2.75	2.75	**1.0**	**1.0**	2.0	2.0
system2	-	-	1.75	2.0	3.0	3.0
system1	-	-	-	-	**1.0**	**1.0**
MTI	-	-	-	-	3.25	3.0
RMAIP	2.50	**1.75**	5.0	4.5	5.25	5.5
RMAI	3.25	3.0	5.0	4.5	8.5	7.25
RMAIR	6.25	6.0	4.5	3.25	6.25	6.25
RMAIA	5.75	5.5	4.0	5.25	7.25	5.75
RMAIN	4.50	3.25	6.0	5.0	6.5	6.25
Wishart-S3-NP	8.75	9.0	14.25	15.0	-	-
Wishart-S1-KNN	8.75	9.25	12.25	12.5	-	-
Wishart-S5-Ensemble	9.5	8.0	9.50	10.25	-	-
mc4	14.75	14.25	21.0	21.0	21.5	21.25
mc3	11.0	11.25	19.75	19.75	22.0	21.5
mc5	11.25	10.0	15.0	14.75	17.0	17.0
cole_hce2	9.25	9.5	11.25	9.25	12.75	12.0
bioasq_baseline	14.0	14.0	17.75	16.75	20.75	
cole_hce1	13.5	13.5	14.75	14.0	16.0	14.75
mc1	8.75	8.25	13.75	13.25	13.0	13.5
mc2	11.25	11.5	17.75	18.25	14.25	15.75
utai_rebayct	15.5	16.0	16.75	17.5	19.25	21.5
Wishart-S2-IR	9.75	10.75	8.5	9.25	-	-
Wishart-S5-Ngram	-	-	10.5	9.75	-	-
utai_rebayct_2	-	-	-	-	18.25	18.5
TCAM-S1	-	-	-	-	11.25	12.25
TCAM-S2	-	-	-	-	12.25	12.25
TCAM-S3	-	-	-	-	12.5	12.5
TCAM-S4	-	-	-	-	12.0	12.75
TCAM-S5	-	-	-	-	12.75	12.0
FU_System	-	-	-	-	24.0	23.25

Table 3.4: Average ranks for each system across the batches of Task 1a for the measures MiF and LCA-F. A hyphenation symbol (-) is used whenever the system participated in less than 4 times in the batch.

3.2 Task 1b

Phase A. Table 3.5 presents the statistics of the training and test data provided to the participants. Figure 3.2 presents the format of the training data for Task 1b. As in Task 1a the evaluation included three test batches. For phase A of Task 1b the systems were allowed to submit responses to any of the corresponding categories, that is documents, concepts, snippets and RDF triples. For each category, we ranked the systems according to the Mean Average Precision (MAP) measure (Balikas et al., 2013). The final ranking for each batch is calculated as the average of the individual rankings in the different categories. The detailed results for Task 1b phase A can be found in http://bioasq.lip6.fr/results/1b/phaseA/.

Batch	Size	# of documents	# of snippets	# of concepts	# of triples
training	29	10.31	14.00	4.82	3.67
1	100	14.89	19.89	8.30	21.87
2	100	14.66	20.24	7.58	5.56
3	82	14.47	17.06	6.24	4.50
total	311	14.28	18.70	7.11	9.00

Table 3.5: Statistics on the training and test datasets of Task 1b: numbers of documents, snippets, concepts and triples refer to averages.

Table 3.6 presents the average ranking of each system in each batch of Task 1b phase A. It is evident from the results that the participating systems did not manage to perform better than the two baselines that were used in phase A. Note also that the systems did not respond to all the categories. For example, the MCTeam systems did not submit snippets throughout the task. Focusing on the specific categories, like concepts, for the Wishart system we observe that it achieves to have a balanced behavior with respect to the baselines (Table 3.7). This is evident from the F-measure which is superior to the values of the two baselines. This can be explained by the fact that the Wishart-S1 system responded with short lists while the baselines returned always long lists (50 and 100 items respectively). Similar observations hold also for the other two batches.

Phase B. In phase B of Task 1b the systems were asked to report exact and ideal answers. The systems were ranked according to the manual evaluation of ideal answers by the BioASQ experts (Balikas et al., 2013). For reasons of completeness, we report also the results of the systems for the exact answers.

System	Batch 1	Batch 2	Batch 3
Top 100 Baseline	**1.0**	**1.875**	**1.25**
Top 50 Baseline	2.5	2.375	1.75
MCTeamMM	3.625	4.5	3.5
MCTeamMM10	3.625	4.5	3.5
Wishart-S1	4.25	3.875	-
Wishart-S2	-	4.125	-

Table 3.6: Average ranks for each system for each batch of phase A of Task 1b. The MAP measure was used to rank the systems. A hyphen (symbol -) is used whenever the system did not participate in the corresponding batch.

```
{ "questions": [
        {
                "id": "the ID",
                "body": "the question?",
                "type": "the type of the question",
                "concepts": [
                        "c1",
                        "c2",
                        ...
                        "cn"
                ],
                "documents": [
                        "d1",
                        "d2",
                        ...
                        "dn"
                ],
                "exact_answer": [
                        "ea1",
                        "ea2",
                        ...
                ],
                "ideal_answer": "the ideal answer",
                "snippets":[
                        {
                                "document": "dk",
                                "beginSection": "sections. #b",
                                "endSection": "sections.#e",
                                "offsetInBeginSection": number,
                                "offsetInEndSection": number,
                                "text": "the snippet"
                        }
                ],
                "triples": [
                        {
                                "o": "object",
                                "p": "predicate",
                                "s": "subject"
                        },
                        ...
                ]
        },
        ...
]
}
```

Figure 3.2: The format of the training data of Task1b.

System	Mean precision	Mean recall	Mean F-measure	MAP	GMAP
Top 100 Baseline	0.080	0.858	0.123	0.472	0.275
Top 50 Baseline	0.121	0.759	0.172	0.458	0.203
Wishart-S1	0.464	0.429	0.366	0.342	0.063
MCTeamMM	0.000	0.000	0.000	0.000	0.000
MCTeamMM10	0.000	0.000	0.000	0.000	0.000

Table 3.7: Results for batch 1 for concepts in phase A of Task1b.

To do so, we average the individual rankings of the systems for the different types of questions, that is

System	Batch 1	Batch 2	Batch 3
Wishart-S1	**2.0**	**1.0**	-
Wishart-S2	**2.0**	-	-
Wishart-S3	**2.0**	-	-
Baseline1	4.66	**2.33**	**2.33**
Baseline2	4.33	4.0	2.66
main system	6.0	4.33	3.0
system 2	-	5.33	3.33
system 3	-	5.5	3.66
system 4	-	5.5	-

Table 3.8: Average ranks for each system and each batch of phase B of Task 1b. The final rank is calculated across the individual ranks of the systems for the different types of questions. A dash symbol (-) is used whenever the system did not participate to the corresponding batch.

System	Batch 1	Batch 2	Batch 3
Wishart-S1	**3.94**	**4.23**	-
Wishart-S2	**3.94**	-	-
Wishart-S3	**3.94**	-	-
Baseline1	2.86	-	**3.19**
Baseline2	2.73	-	3.17
main system	3.35	3.39	3.13
system 2	-	3.34	3.07
system 3	-	3.34	2.98
system 4	-	3.34	-

Table 3.9: Average scores for each system and each batch of phase B of Task 1b for the ideal answers. The final score is calculated as the average of the individual scores of the systems for the different evaluation criteria. A hyphenation symbol (-) is used whenever the system did not participate in the corresponding batch.

Yes/No, factoids and list.

Table 3.8 presents the average ranks for each system for the exact answers. In this phase we note that the Wishart system was able to outperform the BioASQ baselines.

Table 3.9 presents the average scores[6] of the biomedical experts for each system across the batches. Note that the scores are between 1 and 5 and the higher it is the better the performance. According to the results, the systems were able to provide comprehensible answers, and in some cases like in the second batch, highly readable ones. For example Table 3.10 presents the answer of the Wishart-S1 system along with the golden answer to the question: *Which drug should be used as an antidote in benzodiazepine overdose?* Of course the quality of the answer depends on the difficulty of the question. This seems to be the case in the last batch were the average scores are lower with respect to the other batches. Also, the calculated measures using ROUGE seem to be consistent with the manual scores in the first two batches while the situation is inverted in the third batch.

[6]Please consult the description of the evaluation measures used in the challenge for more information .

Wishart-S1	golden answer
Benzodiazepine (BZD) overdose (OD) continues to cause significant morbidity and mortality in the UK. Flumazenil is an effective antidote but there is a risk of seizures, particularly in those who have co-ingested tricyclic antidepressants. (PMID: 21785147) Flumazenil is a benzodiazepine antagonist. It is widely used as an antidote in comatose patients suspected of having ingested a benzodiazepine overdose. (PMID: 19500521)	Flumazenil should be used in all patients presenting with suspected benzodiazepine overdose. Flumazenil is a potent benzodiazepine receptor antagonist that competitively blocks the central effects of benzodiazepines and reverses behavioral, neurologic, and electrophysiologic effects of benzodiazepine overdose. Clinical efficacy and safety of flumazenil in treatment of benzodiazepine overdose has been confirmed in a number of rigorous clinical trials. In addition, flumazenil is also useful to reverse benzodiazepine induced sedation and to and to diagnose benzodiazepine overdose.

Table 3.10: The ideal answers returned from the system Wishart-S1 along with the golden one.

4

Prizes

Tables 4.1 presents the prizes that were awarded to the winners of each task.

Participant	**Task**	**Place**	**Prize** (in Euros)
Tsoumakas G.	1a	1st and 2nd	2,750
Dongqing Zhu	1b (Phase A)	1st	600
Mayo Clinic	1b (Phase A)	1st	1,000
Liu Yifeng (University of Alberta)	1b (Phase A)	2nd	900
Liu Yifeng (University of Alberta)	1b (Phase B)	1st	1,600
Kota Makise (Toyota Institute)	1b (Phase B)	2nd	900
Total			**7,750**

Table 4.1: Prizes of Task1a and Task1b.

Liu Yifeng has also received the additional best overall contribution award sponsored by Transinsight. For more information please consult the corresponding Web page http://www.bioasq.org/participate/prizes.

5

Conclusions and Potential Impact

5.1 Task 1a

In the first task of BIOASQ a sufficient number of teams participated submitting a large number of systems. The majority of the systems were able to successfully cope with both the large scale of the problem as well as the on-line evaluation procedure. From the results, we can draw three main conclusions:

- The majority of the systems were able to achieve good performance being able to outperform the weak baseline throughout the batches.
- The best systems were able to outperform the strong baseline (MTIFL), thus pushing the state-of-the-art. We regard this as a very important achievement towards the goal of developing accurate classification systems for large-scale problems.
- A variety of methods have been used by the participants like pure machine learning approaches, search-based approaches and learning-to-rank approaches. The different technologies that were used allowed us to asses them on a very large-scale scenario. More specifically, simple machine learning approaches like the ones used in (Tsoumakas et al., 2013) are able to achieve state-of-the-art results. Additionally, the learning-to-rank approach followed in (Mao and Lu, 2013) showed that such systems can be effective for large-scale classification tasks. On the other hand, the hierarchical approach employed in (Ribadas et al., 2013) achieved moderate results revealing the fact that the MeSH hierarchy may not be appropriate for classification tasks.

5.2 Task 1b

Only a few systems participated in the second task of the BIOASQ challenge, so that we cannot draw safe conclusions. Additionally, in phase A the participating systems were not able to outperform the baselines. As the systems seem to follow well principled ways to construct queries, we cannot conclude whether their low performance can be attributed to the use of low performing methods. Other factors should also be considered, like whether the systems were able to retrieve appropriate responses from the designated resources.

Concerning phase B of the task, the participating systems were able to obtain better performance than that of the baselines. Again, the low participation does not allow to make any safe conclusions. Interestingly, the automatic measures that were used to asses the ideal answers seem to be in accordance with the manual scores assigned by the BIOASQ experts in the first two batches of the task, while in the third one the measure had a different behavior.

5.3 Potential Impact of New Technologies

From the analysed approaches above, we distinguished the best two, i.e. the best rated system presented in (Tsoumakas et al., 2013) as well as the second ranked one (Mao and Lu, 2013), in order to discuss their impact on specialized search engines (e.g. GoPubMed).

The top rated systems which were able to improve substantially over the MTI baseline follow different approaches. The best rated system presented in (Tsoumakas et al., 2013) followed a pure machine learning approach employing SVMs while the second ranked one followed a hybrid approach mixing an information retrieval phase and a learning-to-rank procedure (Mao and Lu, 2013). While the former approach is able to provide better results the latter enjoys faster training and inference times (very crucial for on-line search engines like GoPubMed). So, potentially both technologies could be used in order to boost the prediction capabilities of a search engine where the first can be employed in an off-line scenario for improving the annotations of the articles in the database. The technologies of the latter system can be integrated in the front-end of the search engine in order to provide accurate and fast results to the users. In addition, the approaches developed and submitted in the framework of Task 1b, may be used as a basis to develop Q&A expansions of GoPubMed. Based on this observation, GoPubMed could be among the first search engines to launch a fully fledged Q&A for the biomedical domain in the search engine market. More details on the potential impact of the proposed approaches in BioASQ challenges will be presented in the deliverable D2.11 (Exploitation and dissemination plan).

A

Appendix

(a) MiF

System	1	2	3	4	5	6
MTIFL	1	1	1	2	2	2
mc4	2	22	19	21	17	21
mc3	3	12	13	20	16	20
RMAI	4	3	3	4	3	4
RMAIP	-	2	2	3	4	3
RMAIR	-	5	6	7	7	7
RMAIA	-	6	5	6	6	6
RMAIN	-	4	4	5	5	5
Wishart-S3-NP	5	8	9	14	-	13
Wishart-S1-KNN	6	10	7	12	-	12
Wishart-S5-Ensemble	7	13	10	11	-	10
system3	8	11	-	1	1	1
mc5	9	14	11	15	11	16
bioasq_baseline	10	16	16	18	14	19
cole_hce2	11	15	14	8	10	8
cole_hce1	12	17	15	16	12	15
mc1	13	9	-	9	8	9
mc2	14	19	12	10	9	17
utai_rebayct	15	21	17	17	13	18
Wishart-S2-IR	16	7	8	13	-	11

(b) LCA-F

System	1	2	3	4	5	6
MTIFL	1	6	1	3	2	5
mc4	4	19	18	20	16	21
mc3	3	13	14	18	15	20
RMAI	2	3	3	5	4	4
RMAIP	-	1	2	2	3	2
RMAIR	-	4	6	7	7	7
RMAIA	-	5	5	6	6	6
RMAIN	-	2	4	4	5	3
Wishart-S3-NP	6	11	8	15	-	11
Wishart-S1-KNN	7	8	10	12	-	13
Wishart-S5-Ensemble	5	14	7	10	-	10
system3	8	12	-	1	1	1
mc5	9	10	11	14	10	16
bioasq_baseline	10	16	16	17	14	18
cole_hce2	13	15	13	8	9	8
cole_hce1	14	18	15	16	11	14
mc1	12	7	-	9	8	9
mc2	11	17	12	11	12	17
utai_rebayct	15	22	17	19	13	19
Wishart-S2-IR	16	9	9	13	-	12

Table A.1: Detailed ranks for each system in batch 1 of Task 1a for the MiF and LCA-F measures respectively.

(a) MiF

System	1	2	3	4	5	6
MTIFL	3	3	3	3	2	3
mc4	22	22	20	22	20	-
mc3	18	21	22	21	19	-
RMAIP	4	5	5	7	6	7
Wishart-S3-NP	14	15	17	17	13	15
RMAI	5	7	6	6	3	8
RMAIR	8	6	7	5	1	6
RMAIN	6	8	8	8	5	5
RMAIA	7	4	4	4	4	4
Wishart-S1-KNN	13	13	15	13	11	12
system2	2	1	2	2	22	2
Wishart-S4-Ngram	11	11	12	11	9	11
Wishart-S5-Ensemble	10	10	10	10	8	10
system3	1	2	1	1	21	1
mc5	19	18	13	14	15	-
bioasq_baseline	17	20	19	20	18	17
cole_hce2	12	12	11	12	10	13
cole_hce1	15	16	16	16	14	14
mc1	21	14	14	15	12	-
mc2	20	17	21	18	16	-
utai_rebayct	16	19	18	19	17	16
Wishart-S2-IR	9	9	9	9	7	9

(b) LCA-F

System	1	2	3	4	5	6
MTIFL	2	3	3	3	3	6
mc4	22	22	20	22	20	-
mc3	18	21	22	21	19	-
RMAIP	4	7	4	7	5	5
Wishart-S3-NP	15	16	17	17	14	15
RMAI	6	4	8	6	2	8
RMAIR	8	5	7	4	1	3
RMAIN	5	8	6	5	4	7
RMAIA	7	6	5	8	6	4
Wishart-S1-KNN	13	14	15	13	12	12
system2	3	2	2	2	22	2
Wishart-S4-Ngram	9	11	12	12	8	11
Wishart-S5-Ensemble	11	10	11	11	10	10
system3	1	1	1	1	21	1
mc5	21	17	13	14	15	-
bioasq_baseline	16	19	18	18	16	17
cole_hce2	12	12	9	9	7	13
cole_hce1	14	15	16	16	13	14
mc1	20	13	14	15	11	-
mc2	19	18	21	19	17	-
utai_rebayct	17	20	19	20	18	16
Wishart-S2-IR	10	9	10	10	9	9

Table A.2: Detailed ranks for each system in batch 2 of Task 1a for the MiF and LCA-F measures respectively.

(a) MiF

System	1	2	3	4	5	6
MTIFL	4	5	5	5	5	2
MTI	-	4	4	4	4	1
mc4	20	20	21	27	25	25
mc3	19	19	26	26	24	26
RMAIP	5	6	6	6	-	4
RMAI	9	10	9	8	9	8
RMAIR	8	8	10	10	6	3
RMAIN	7	7	7	7	7	5
RMAIA	6	9	8	9	10	6
system2	3	3	3	3	3	27
system3	2	2	2	2	2	17
mc5	15	16	20	17	26	21
bioasq_baseline	18	18	25	25	23	24
cole_hce2	12	11	13	18	17	15
cole_hce1	13	14	18	20	19	19
mc1	10	12	23	22	16	14
mc2	16	13	22	16	15	13
utai_rebayct	17	17	24	23	21	22
TCAM-S1	-	-	16	14	8	7
TCAM-S2	-	-	15	12	12	10
TCAM-S3	-	-	11	13	14	12
TCAM-S4	-	-	17	11	11	9
TCAM-S5	-	-	12	15	13	11
utai_rebayct_2	14	-	19	21	20	20
FU_System	-	-	27	24	22	23
system1	1	1	1	1	1	18

(b) LCA-F

System	1	2	3	4	5	6
MTIFL	4	9	5	5	9	2
MTI	-	4	4	3	4	1
mc4	20	20	21	27	24	24
mc3	19	19	26	25	23	26
RMAIP	6	5	6	7	-	5
RMAI	8	10	10	6	8	7
RMAIR	9	8	9	10	5	3
RMAIN	7	6	8	8	6	6
RMAIA	5	7	7	9	7	4
system2	3	3	3	4	3	27
system3	2	2	2	2	2	18
mc5	14	16	20	18	26	22
bioasq_baseline	16	17	24	23	22	23
cole_hce2	11	11	13	16	15	13
cole_hce1	12	13	17	20	19	17
mc1	10	12	23	21	18	14
mc2	17	14	22	19	16	16
utai_rebayct	18	18	25	26	25	25
TCAM-S1	-	-	16	14	11	8
TCAM-S2	-	-	15	12	12	10
TCAM-S3	-	-	11	13	14	12
TCAM-S4	-	-	18	11	13	9
TCAM-S5	-	-	12	15	10	11
utai_rebayct_2	15	-	19	22	20	20
FU_System	-	-	27	24	21	21
system1	1	1	1	1	1	19

Table A.3: Detailed ranks for each system in batch 3 of Task 1a for the MiF and LCA-F measures respectively.

Bibliography

A. R. Aronson and F.-M. Lang. An overview of metamap: historical perspective and recent advances. *Journal of the American Medical Informatics Association*, 17:229–236, 2010.

R. Babbar, I. Partalas, E. Gaussier, and M.-R. Amini. On flat versus hierarchical classification in large-scale taxonomies. In *NIPS*, sep 2013.

G. Balikas, I. Partalas, A. Kosmopoulos, S. Petridis, P. Malakasiotis, I. Pavlopoulos, I. Androutsopoulos, N. Baskiotis, E. Gaussier, T. Artieres, and P. Gallinari. Evaluation framework specifications. Project deliverable D4.1, 05/2013 2013. URL sites/default/files/PublicDocuments/BioASQ_D4.1-EvaluationFrameworkSpecification_final.pdf.

S. Bengio, J. Weston, and D. Grangier. Label embedding trees for large multi-class tasks. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 163–171. 2010.

P. N. Bennett and N. Nguyen. Refined experts: improving classification in large taxonomies. In *Proceedings of the 32nd annual International ACM SIGIR conference on Research and development in information retrieval*, pages 11–18, 2009.

A. Beygelzimer, J. Langford, Y. Lifshits, G. Sorkin, and A. Strehl. Conditional probability tree estimation analysis and algorithms. In *Proceedings of the Twenty-Fifth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-09)*, pages 51–58, Corvallis, Oregon, 2009. AUAI Press.

L. Cai and T. Hofmann. Hierarchical document categorization with support vector machines. In *CIKM*, pages 78–87. ACM, 2004.

O. Dekel, J. Keshet, and Y. Singer. Large margin hierarchical classification. In *Proceedings of the twenty-first international conference on Machine learning*, ICML '04, pages 27–35, 2004.

J. Demsar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

A. Doms. *GoPubMed: Ontology-based literature search for the life sciences*. Phd thesis, Technische Universität Dresden, 2010.

T. Gao and D. Koller. Discriminative learning of relaxed hierarchy for large-scale visual recognition. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2072–2079, 2011.

S. Gopal, Y. Yang, B. Bai, and A. Niculescu-Mizil. Bayesian models for large-scale hierarchical classification. In *Neural Information Processing Systems*, 2012.

D. Koller and M. Sahami. Hierarchically classifying documents using very few words. In *Proceedings of the Fourteenth International Conference on Machine Learning*, ICML '97, 1997.

A. Kosmopoulos, I. Partalas, E. Gaussier, G. Paliouras, and I. Androutsopoulos. Evaluation measures for hierarchical classification: a unified view and novel approaches. *CoRR*, abs/1306.6802, 2013. URL http://arxiv.org/pdf/1306.6802v2.

C.-Y. Lin. ROUGE: A package for automatic evaluation of summaries. In *Proceedings of the ACL workshop 'Text Summarization Branches Out'*, pages 74–81, Barcelona, Spain, 2004.

T.-Y. Liu, Y. Yang, H. Wan, H.-J. Zeng, Z. Chen, and W.-Y. Ma. Support vector machines classification with a very large-scale taxonomy. *SIGKDD Explor. Newsl.*, pages 36–43, 2005.

Y. Liu. Bioasq system descriptions (wishart team). Technical report, 2013.

H. Malik. Improving hierarchical svms by hierarchy flattening and lazy classification. In *1st Pascal Workshop on Large Scale Hierarchical Classification*, 2009.

Y. Mao and Z. Lu. Ncbi at the 2013 bioasq challenge task: Learning to rank for automatic mesh indexing. Technical report, 2013.

A. McCallum, R. Rosenfeld, T. M. Mitchell, and A. Y. Ng. Improving text classification by shrinkage in a hierarchy of classes. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 359–367, 1998.

J. Mork, A. Jimeno-Yepes, and A. Aronson. The nlm medical text indexer system for indexing biomedical literature, 2013.

F. Perronnin, Z. Akata, Z. Harchaoui, and C. Schmid. Towards good practice in large-scale learning for image classification. In *Computer Vision and Pattern Recognition*, pages 3482–3489, 2012.

F. Ribadas, L. de Campos, V. Darriba, and A. Romero. Two hierarchical text categorization approaches for bioasq semantic indexing challenge. In *1st BioASQ Workshop: A challenge on large-scale biomedical semantic indexing and question answering*, 2013.

C. N. Silla, Jr. and A. A. Freitas. A survey of hierarchical classification across different application domains. *Data Mining Knowledge Discovery*, 22:31–72, 2011.

T. F. Smith and M. S. Waterman. Comparison of biosequences. *Advances in Applied Mathematics*, 2(4): 482 – 489, 1981.

L. Tang, S. Rajan, and V. K. Narayanan. Large scale multi-label classification via metalabeler. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 211–220, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-487-4. doi: 10.1145/1526709.1526738. URL http://doi.acm.org/10.1145/1526709.1526738.

G. Tsatsaronis, M. Schroeder, G. Paliouras, Y. Almirantis, I. Androutsopoulos, E. Gaussier, P. Gallinari, T. Artieres, M. R. Alvers, M. Zschunke, et al. Bioasq: A challenge on large-scale biomedical semantic indexing and question answering. In *2012 AAAI Fall Symposium Series*, 2012. URL sites/default/files/PublicDocuments/2012-Tsatsaronis-BioASQ.pdf.

G. Tsoumakas, I. Katakis, and I. Vlahavas. Mining Multi-label Data. In O. Maimon and L. Rokach, editors, *Data Mining and Knowledge Discovery Handbook*, pages 667–685. Springer US, 2010.

G. Tsoumakas, M. Laliotis, N. Markontanatos, and I. Vlahavas. Large-scale semantic indexing of biomedical publications. In *1st BioASQ Workshop: A challenge on large-scale biomedical semantic indexing and question answering*, 2013.

X. Wang and B.-L. Lu. Flatten hierarchies for large-scale hierarchical text categorization. In *Fifth IEEE International Conference on Digital Information Management*, pages 139–144, 2010.

K. Q. Weinberger and O. Chapelle. Large margin taxonomy embedding for document categorization. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1737–1744. 2009.

D. Weissenborn, G. Tsatsaronis, and M. Schroeder. Answering factoid questions in the biomedical domain. In *1st BioASQ Workshop: A challenge on large-scale biomedical semantic indexing and question answering*, 2013.

G.-R. Xue, D. Xing, Q. Yang, and Y. Yu. Deep classification in large-scale text hierarchies. In *Proceedings of the 31st annual International ACM SIGIR conference on Research and development in information retrieval*, pages 619–626, 2008.

D. Zhu, D. Li, B. Carterette, and H. Liu. An incemental approach for medline mesh indexing. In *1st BioASQ Workshop: A challenge on large-scale biomedical semantic indexing and question answering*, 2013.