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

Project **FP7-318652 / BioASQ**

Deliverable **D4.9**

Distribution **Public**

http://www.bioasq.org

Evaluation infrastructure software for future oracle use

Georgios Balikas, Ioannis Partalas, Nicolas Baskiotis, Thierry Artieres, Eric Gaussier, Patrick Gallinari, Ioannis Vetsikas, George Paliouras and Aris Kosmopoulos

Status: Final-revised (Version 1.1)

January 2015

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	D4.9
Deliverable title	Evaluation infrastructure software for future oracle use
Contractual date of delivery	M24 (September 2014)
Actual date of delivery	January 2015
Relevant Task(s)	WP4/Task 4.2
Partner Responsible	UPMC
Other contributors	UJF, NCSR “D”
Number of pages	19
Author(s)	Georgios Balikas, Ioannis Partalas, Nicolas Baskiotis, Thierry Artieres, Eric Gaussier, Patrick Gallinari, Ioannis Vetsikas, George Paliouras and Aris Kosmopoulos
Internal Reviewers	Prodromos Malakasiotis
Status & version	Final-revised
Keywords	BioASQ, platform, challenge operation

Executive Summary

The deliverable describes the procedure and the functional details that users of the BIOASQ Participants Area[1] should follow in order to use the BIOASQ oracle.[2]

The BIOASQ oracle is an online application which was developed as part of the BIOASQ Participants Area. It provides its users the opportunity to test their systems using past BIOASQ datasets. Both tasks of the BIOASQ challenge are available in the oracle. When submitting results for a past test dataset the oracle returns the scores for the corresponding BIOASQ evaluation measures along with the ranking of the submission compared with the official challenge submissions and other publicly available oracle submissions. In this way, participants can continue improving their systems by using the datasets and the infrastructure generated during the BIOASQ challenge to evaluate their progress in an off-challenge mode.

The BIOASQ oracle will remain active after the end of the project. This deliverable presents the details of the use of the oracle from a user point-of-view. It can be also seen as guidelines or reference point for the oracle functionality.

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

Contents

List of Figures

1

Introduction

BIOASQ[1] initiated a series of challenges on biomedical semantic indexing and question answering. One of the goals of the project consortium towards the end of the project is to provide a sustainable way of evaluating off-challenge system submissions for the released BIOASQ datasets. That way, the BIOASQ infrastructure and datasets will continue aiding researchers to improve their systems and evaluate them.

To address those requirements the BIOASQ oracle[2] was developed and integrated in the BIOASQ Participants Area.[3] Using the oracle, participants can submit results for already released test datasets and receive as feedback:

1. The scores of the corresponding for each task evaluation measures, and
2. the ranking of the submission compared to the systems that participated in the official part of the challenge and other public submission that were made in the oracle.

Recall, that during the challenge the evaluation was performed incrementally for both tasks. For the semantic indexing task (Task A) the measures were updated as MeSH heading were becoming available and for the question answering task (Task B) indicative scores of the evaluation measures were calculated initially and then the measures were updated after the end of the assessment task. However, the feedback described above is provided in real-time; the oracle uses the updated and refined versions of the golden datasets created during the challenge. For more information on the BIOASQ evaluation measures for each task please consult Balikas et al. (2013b) and Kosmopoulos et al. (2013). For the engineering and development decisions behind the oracle consult Balikas et al. (2014). Finally, for more information on the process used during the creation of the BIOASQ benchmark datasets, consult Malakasiotis et al. (2013) and Ngonga Ngomo et al. (2013).

All tests released during the official part of the challenge for both tasks are available, so that participants can submit their results for them to the oracle. This translates to 33 test datasets for Task A and 8 datasets for Task B.

The rest of the document is organised as follows:

[1] http://bioasq.org
[2] http://bioasq.lip6.fr/oracle/
[3] http://bioasq.lip6.fr

- *Chapter 2* presents the home page of the oracle and explains the fields of the Oracle submission form,
- *Chapter 3* presents the guidelines for using the oracle for submissions for Task A,
- *Chapter 4* presents the guidelines for using the oracle for submissions for Task B, and
- *Chapter 5* explains the way that the oracle results are displayed.

2

The Oracle submission form

Figure 2.1 depicts the home page of the oracle. One can reach the page either by typing the URL http://bioasq.lip6.fr:/oracle/ or by visiting the BioASQ Participants Area[1] and selecting "Submit results" under the "Oracle" option on the horizontal navigation menu, also depicted in Figure 2.1. Note that there are five options under the "Oracle" tag: "Get Data", "Submit results", "Results-Task A", "Results-Task B-Phase A" and "Results-Task B-Phase B". By clicking "Get Data" one can access the test datasets of the challenge. The way that the results are displayed is explained in Chapter 5.

The home page can be decomposed in two elements:

1. the form for submitting guidelines, and
2. the instructions and the explanations of the Oracle functionality.

In Figure 2.1 the fields of the form that provide the information of what is expected for each field are highlighted in red ellipses. These are:

1. *Task*: The user is requested to specify the task she intends to submit results for. She can select one between three options: "Task A", "Task B-phase A" and "Task B-phase B".
2. *Test*: The user is requested to specify the test dataset she submits results for. The field is auto-populated with all the available test datasets, given the task one has selected in the field *Task*.
3. *Your system*: Users participating in the BioASQ challenge are allowed to participate with a maximum of 5 systems because they are usually testing similar or different methods simultaneously. The Oracle auto-populates the field with the systems a user has registered. The username during authentication is used for the systems to be identified. Systems can be added at: http://bioasq.lip6.fr/profile/
4. *Your system results*: The field corresponds to a typical "Browse" button, clicking on it, a window asking the user to identify the results file (for the selected task and test dataset) in his file-system opens.

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

After filling the aforementioned required information, the user can submit the results by clicking on the "Submit" button, which is also highlighted in the Figure 2.1. The Oracle processes the submitted results and returns the scores of the evaluation measures. The way they are rendered depends on the "Task" the user has submitted results for since, the selected evaluation measures are different for each task. More information and examples concerning the evaluation measures and the rendering of the scores is provided in the next chapters.

The format of the data submitted for each task should follow the guidelines of the respective task. One can find them by clicking on "Guidelines" to the horizontal navigational menu, or follow the links:

- http://bioasq.lip6.fr/general_information/Task2a/ for Task A, and
- http://bioasq.lip6.fr/general_information/Task2b/ for Task B. Note that here one can find the guidelines for both phases of the task.

In case the format is different from the expected, informative messages are returned to the user. When an unexpected error is caused, the process ends in the 500 error page.

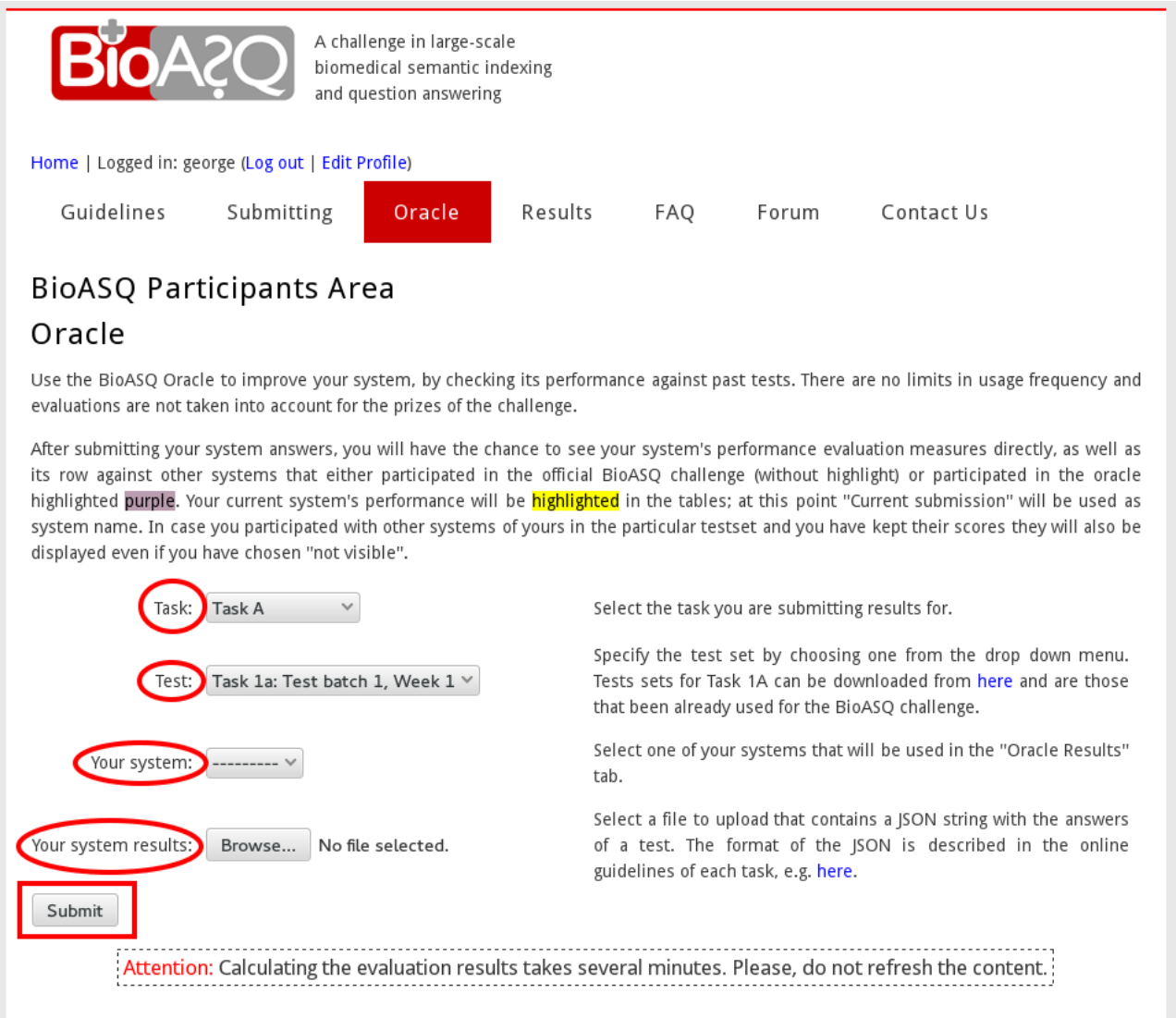

Figure 2.1: The home page of the Oracle, available at http://bioasq.lip6.fr

3

Submitting results for Task A

When submitting results for Task A the format of the submitted file should follow the guidelines at http://bioasq.lip6.fr/general_information/Task2a/. In the auto-populated field of the submission form "Test" the test datasets of the first and the second year are displayed. They are discriminated by the flag "Task 1a" for the first year's datasets and the flag "Task 2a" for the second year's datasets. The oracle's response when submitting results in the correct format can be decomposed in:

1. a form with a boolean flag asking for confirmation to save the scores and another boolean flag to keep the results visible. The form is depicted in Figure 3.1.

2. a sorted table populated with the scores of the flat evaluation measures, depicted in Figure 3.2. The table is sorted using the MiF (Micro F-measure) column. The submission of the user is highlighted with a yellow background and named "Current Submission".

3. a sorted table populated with the scores of the hierarchical evaluation results, depicted in Figure 3.3. The table is sorted using the LCA-F (Least Common Ancestor- F-measure) column. Again, the submission of the user is highlighted with a yellow background and named "Current Submission".

Note, that a user can maintain only one saved version of results per system per test dataset. When the "Save my score" flag in the form is enabled the new scores will replace any previous scores of the particular system for the particular test dataset of the submission. To apply those changes the user should click the "Submit" button, also illustrated in the form of Figure 3.1.

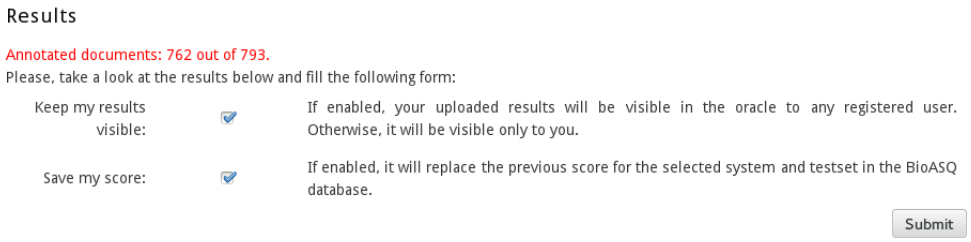

Figure 3.1: The menu for saving results, displayed with the scores of the evaluation measures.

Flat Measures

System	MiF	Acc.	EBP	EBR	EBF	MaP	MaR	MaF	MiP	MiR
MTI First Line Index	0.5575	0.3979	0.6166	0.5372	0.5498	0.5318	0.4831	0.4638	0.6045	0.5172
MeSH Indexing	0.5230	0.3689	0.5072	0.5966	0.5224	0.4590	0.4630	0.4352	0.4865	0.5653
MeSH Indexing Pre	0.5230	0.3689	0.5072	0.5966	0.5224	0.4590	0.4630	0.4352	0.4865	0.5653
MeSH Indexing New	0.5103	0.3588	0.4723	0.6196	0.5115	0.4290	0.4867	0.4483	0.4516	0.5864
MeSH Indexing Add	0.5016	0.3414	0.4319	0.6324	0.4943	0.4085	0.4937	0.4480	0.4319	0.5982
MeSH Indexing Ref	0.5008	0.3407	0.4311	0.6314	0.4934	0.4082	0.4939	0.4482	0.4311	0.5972
Wishart-S1	0.4738	0.3171	0.5446	0.4646	0.4664	0.5079	0.3354	0.3213	0.5136	0.4398
Wishart-S2	0.4735	0.3161	0.5089	0.4893	0.4661	0.4805	0.3624	0.3417	0.4830	0.4644
Wishart-S3	0.4715	0.3141	0.4912	0.5012	0.4643	0.4657	0.3746	0.3490	0.4665	0.4765
Wishart-S5	0.4706	0.3210	0.4903	0.4910	0.4716	0.4984	0.3638	0.3492	0.4909	0.4520
MCTeamMM	0.4439	0.2962	0.4222	0.4927	0.4374	0.4644	0.3207	0.3100	0.4222	0.4679
MCTeamMR24	0.4352	0.2884	0.4502	0.4507	0.4329	0.4980	0.3053	0.3063	0.4483	0.4228
MCTeamMM10	0.4282	0.2805	0.3687	0.5339	0.4199	0.3886	0.3586	0.3270	0.3687	0.5107
MCTeamSR	0.3945	0.2507	0.3041	0.5827	0.3855	0.3044	0.4051	0.3420	0.3041	0.5616
cole_hce2	0.3929	0.2470	0.2851	0.6570	0.3844	0.2416	0.5038	0.3964	0.2851	0.6317
cole_hce1	0.3334	0.2023	0.2419	0.5678	0.3280	0.1668	0.4240	0.3302	0.2419	0.5361
BioASQ_Baseline	0.2767	0.1627	0.2743	0.2994	0.2645	0.3263	0.3726	0.3229	0.2577	0.2988
Current Submission	0.2674	0.1627	0.2809	0.2994	0.2681	0.3255	0.3713	0.3215	0.2501	0.2873
utai_rebayct	0.2506	0.1446	0.1819	0.4331	0.2478	0.2306	0.3410	0.2932	0.1819	0.4029
UCD-CMgg	0.2450	0.1433	0.1944	0.3650	0.2432	0.3367	0.4231	0.3664	0.1878	0.3524
MCTeamSR8	0.2196	0.1338	0.2817	0.2009	0.2254	0.4368	0.2342	0.2328	0.2810	0.1802
UCD-CMr	0.2109	0.1233	0.1543	0.4170	0.2145	0.2408	0.4889	0.3972	0.1426	0.4046
UCD-CMd	0.1427	0.0860	0.5070	0.0945	0.1492	0.4351	0.0910	0.0908	0.4272	0.0857
UCD-CMp	0.1427	0.0860	0.5070	0.0945	0.1492	0.4351	0.0910	0.0908	0.4272	0.0857
multifactorial	0.1191	0.0772	0.7047	0.0772	0.1370	0.7047	0.0003	0.0003	0.7047	0.0651

Figure 3.2: The part of the returned results with the scores of the flat measures.

Hierarchical Measures

System	LCA-F	HiP	HiR	HiF	LCA-P	LCA-R
MTI First Line Index	0.4609	0.7477	0.6454	0.6670	0.5198	0.4427
MeSH Indexing	0.4539	0.6780	0.7011	0.6628	0.4618	0.4827
MeSH Indexing Pre	0.4539	0.6780	0.7011	0.6628	0.4618	0.4827
MeSH Indexing New	0.4490	0.6452	0.7282	0.6589	0.4359	0.5005
MeSH Indexing Add	0.4423	0.6104	0.7527	0.6529	0.4111	0.5134
MeSH Indexing Ref	0.4414	0.6107	0.7506	0.6520	0.4109	0.5117
Wishart-S5	0.4064	0.6737	0.6380	0.6330	0.4239	0.4173
Wishart-S3	0.3975	0.6693	0.6364	0.6168	0.4187	0.4167
Wishart-S2	0.3965	0.6854	0.6216	0.6151	0.4277	0.4065
MCTeamMM	0.3904	0.6181	0.6046	0.5888	0.4007	0.4060
Wishart-S1	0.3887	0.7127	0.5911	0.6070	0.4381	0.3854
MCTeamMR24	0.3877	0.6628	0.5544	0.5777	0.4335	0.3753
MCTeamMM10	0.3863	0.5576	0.6603	0.5837	0.3626	0.4417
cole_hce2	0.3745	0.4515	0.8021	0.5603	0.2988	0.5414
MCTeamSR	0.3679	0.4833	0.7203	0.5594	0.3117	0.4826
cole_hce1	0.3351	0.4068	0.7551	0.5115	0.2651	0.4918
BioASQ_Baseline	0.3094	0.5404	0.5391	0.5090	0.3424	0.3096
utai_rebayct	0.2884	0.3518	0.6811	0.4471	0.2290	0.4216
MCTeamSR8	0.2776	0.5587	0.4044	0.4473	0.3667	0.2352
UCD-CMgg	0.2621	0.3989	0.6159	0.4659	0.2165	0.3610
UCD-CMr	0.2618	0.3284	0.6884	0.4280	0.2056	0.4011
Current Submission	0.2202	0.4078	0.4124	0.3792	0.2326	0.2435
UCD-CMd	0.1620	0.6877	0.1547	0.2279	0.3809	0.1102
UCD-CMp	0.1620	0.6877	0.1547	0.2279	0.3809	0.1102
multifactorial	0.0923	0.7203	0.0594	0.1082	0.2841	0.0565

Figure 3.3: The part of the returned results with the scores of the hierarchical measures.

4

Submitting results for Task B

The process of submitting results for Task B is similar with Task A. One should first take into account the information provided in the online guidelines and then create a file in the described format. The interaction with the form begins by selecting the Task: “Task B-phase A” and “Task B-phase B”. Provided that the step is completed, the test datasets of the two years are discriminated by the “Task 1B”, “Task 2B” in the beginning of the name of the test dataset.

4.1 Oracle’s response for Phase A

The oracle’s response after submitting results in the correct format can be decomposed in:

1. a form with a boolean flag asking for confirmation to save the scores and another boolean flag to keep the results visible. The form is the same with the one described in Chapter 3 and is depicted in Figure 3.1.
2. a table with the scores of the evaluation measures for the retrieved documents (Figure 4.1),
3. a table with the scores of the evaluation measures for the retrieved snippets (Figure 4.2),
4. a table with the scores of the evaluation measures for the retrieved concepts (Figure 4.3), and
5. a table with the scores of the evaluation measures for the retrieved RDF triples (Figure 4.4).

In every one of those tables the results are sorted using the MAP measure. The results of the submission are highlighted with a yellow background and named “Current Submission”.

4.2 Oracle’s response for Phase B

The Oracle’s response after submitting results for Phase B of Task B consists of the following:

1. a form with a boolean flag asking for confirmation to save the scores and another boolean flag to keep the results visible. The form is the same with the one described in Chapter 3 and is depicted in Figure 3.1.

2. the scores of the evaluation measures for the exact answers (Figure 4.5),
3. the scores of the evaluation measures for the ideal answers (Figure 4.6).[1]

Documents

System Name	Mean precision	Recall	F-Measure	MAP	GMAP
Top 100 Baseline	0.1578	0.2867	0.1569	0.1090	0.0018
Top 50 Baseline	0.1577	0.2307	0.1482	0.0968	0.0015
Current Submission	0.1466	0.2326	0.1416	0.0958	0.0012
MCTeamMM	0.1121	0.1375	0.0867	0.0522	0.0002
MCTeamMM10	0.0275	0.1375	0.0433	0.0522	0.0002
Wishart-S1	0.0819	0.1118	0.0772	0.0382	0.0002
Wishart-S2	0.0819	0.1118	0.0772	0.0382	0.0002

Figure 4.1: The table with the scores of the evaluation measures for the retrieved documents.

Snippets

System Name	Mean precision	Recall	F-Measure	MAP	GMAP
Wishart-S1	0.0564	0.0772	0.0491	0.0360	0.0002
Wishart-S2	0.0564	0.0772	0.0491	0.0360	0.0002
Top 100 Baseline	0.0871	0.0858	0.0605	0.0337	0.0003
Top 50 Baseline	0.0855	0.0640	0.0515	0.0272	0.0002
Current Submission	0.0682	0.0614	0.0415	0.0196	0.0002
MCTeamMM	0.0000	0.0000	0.0000	0.0000	0.0000
MCTeamMM10	0.0000	0.0000	0.0000	0.0000	0.0000

Figure 4.2: The table with the scores of the evaluation measures for the retrieved snippets.

Concepts

System Name	Mean precision	Recall	F-Measure	MAP	GMAP
Top 100 Baseline	0.0561	0.7833	0.0939	0.4634	0.2148
Current Submission	0.0902	0.7551	0.1359	0.4625	0.1967
Top 50 Baseline	0.0974	0.7399	0.1455	0.4555	0.1989
Wishart-S1	0.4023	0.4663	0.3742	0.4108	0.0937
Wishart-S2	0.3966	0.4794	0.3788	0.3653	0.0979
MCTeamMM	0.0000	0.0000	0.0000	0.0000	0.0000
MCTeamMM10	0.0000	0.0000	0.0000	0.0000	0.0000

Figure 4.3: The table with the scores of the evaluation measures for the retrieved concepts.

[1]The set of the selected evaluation measures for the ideal answers of the oracle is subject to change. More evaluation measures may be added in the future.

RDF Triples

System Name	Mean precision	Recall	F-Measure	MAP	GMAP
Top 100 Baseline	0.0167	0.3404	0.0309	0.0844	0.0013
Top 50 Baseline	0.0311	0.3265	0.0540	0.0841	0.0013
Current Submission	0.0311	0.3265	0.0540	0.0841	0.0013
MCTeamMM	0.0222	0.4403	0.0413	0.0763	0.0033
MCTeamMM10	0.0222	0.4403	0.0413	0.0763	0.0033
Wishart-S1	0.0000	0.0000	0.0000	0.0000	0.0000
Wishart-S2	0.0000	0.0000	0.0000	0.0000	0.0000

Figure 4.4: The table with the scores of the evaluation measures for the retrieved RDF triples.

Exact Answers

	Yes/No	Factoid			List		
System Name	Accuracy	Strict Acc.	Lenient Acc.	MRR	Mean precision	Recall	F-Measure
Wishart-S1	0.9615	0.2500	0.3000	0.3000	0.4060	0.3127	0.3336
Current Submission	0.9615	0.2500	0.3000	0.3000	0.3748	0.3152	NaN
BioASQ Baseline 2	0.5000	-	-	-	0.0612	0.2062	0.0789
main system	0.4231	-	-	-	0.0603	0.1040	0.0680
system 2	0.4231	-	-	-	0.0437	0.0445	0.0414
system 3	0.4231	-	-	-	0.0644	0.0440	0.0488
system 4	0.4231	-	-	-	0.0644	0.0440	0.0488
BioASQ_Baseline	0.2692	-	-	-	0.0612	0.2062	0.0789

Figure 4.5: The table with the scores of the evaluation measures for the exact answers.

Ideal Answers

	Automatic scores	
System Name	Rouge-2	Rouge-SU4
Wishart-S1	0.1035	0.1123
Current Submission	0.1011	0.1103
system 2	0.0565	0.0618
system 3	0.0555	0.0611
system 4	0.0555	0.0611
main system	0.0546	0.0625
BioASQ_Baseline	0.0360	0.0428
BioASQ Baseline 2	0.0357	0.0424

Figure 4.6: The table with the scores of the evaluation measures for ideal answers.

5

Browsing the oracle results

The scores of the evaluation measures for each task of the challenge are accessible under the menu "Oracle" (depicted also in Figure 2.1). There are three options relevant to the scores of the evaluation measures i.e. "Results-Task A", "Results-Task B-Phase A" and "Results-Task B-Phase B". By following the links one can browse over tables that contain the scores of the systems that participated in the official part of the challenge and the scores of the systems that used the oracle and selected to keep their results publicly available. In the latter case, the results are highlighted purple. For example, in the Figure 5.1, the system "testing2" is a system that submitted results using the oracle and kept the results publicly available.

Wishart-S3	0.4706	0.5381	0.4393	0.4568	0.5295	0.3330	0.3212	0.5198	0.4300	0.3077
Wishart-S2	0.4705	0.5083	0.4616	0.4564	0.5021	0.3555	0.3372	0.4908	0.4518	0.3067
testing2	0.4705	0.5083	0.4616	0.4564	0.5021	0.3555	0.3372	0.4908	0.4518	0.3067
Wishart-S4	0.4665	0.5648	0.4173	0.4516	0.5582	0.3105	0.3054	0.5453	0.4077	0.3038
MCTeamMM	0.4598	0.4977	0.4419	0.4492	0.5103	0.2824	0.2831	0.4977	0.4273	0.3059

Figure 5.1: The results of Task A of the official challenge and the oracle submissions.

6

BioASQ Participants Area installation

The chapter discusses how one can install the BioASQ Participants Area in his local machine. Following the provided instructions, a working copy of the software, fully functional, will be running in the local machine. The platform is available at https://github.com/balikasg/BioASQParticipantsArea. After copying the contents of the git project to a directory the proposed process includes the following steps:

1. Create an isolated Python environment.
2. Install the platform dependancies.
3. Add your path to the `setting.py` file.
4. Synchronize your database, and,
5. Launch the application in a local server.

For the rest of the chapter assume that the extraction was in the directory : `\local\balikasg\platformInstall\`

6.1 Dependencies

In order to install the platform we will first create an isolated Python environment and then we will install a few module dependacies that are required.

Creating an isolated Python environment

The basic problem being addressed is the problem of dependencies and versions, and indirectly permissions. Imagine you have an application that needs version 1 of Module1, but another application requires version 2 of this same module. To be able to use both applications the best solution is to create an environment that has its own installation directories. More generally, we want to be able to install in

a computer the platform and leave it functional without causing problem or meshing with the already existing modules. To create this isolated environment we use `virtualenv`.[1]

Having a working installation of `virtualenv`, the only thing that remains is to create a fully isolated environment:

```
virtualenv --no-site-packages myENV
```

There is no output for for this command. Instead a folder `myENV` will be created in your working directory and its structure will be:

```
|-- myEnv
|-- requirements.txt
`-- webexample
```

Note that currently "requirements.txt" is in the github directory of the project. The version of `virtualenv` that was used during the preparation of those guidelines is `1.11.4`. Since we used Python 2.7 during the developement of the platform problems may occur if you try to deploy it with Python 3.X.[2] To overcome this, note that `virtualenv` supports choosing the Python version to be used in the isolated environment. In that case the command for creating the isolated environment would be:

```
virtualenv -p /usr/bin/python2.6 --no-site-packages myENV
```

In the above command `/usr/bin/python2.6` is the path to your Python installation.

Having created the isolated environment you can activate it by:

```
source myENV/bin/activate
```

You can deactivate it by simple typing:

```
deactivate
```

Database

Django supports many databases. In the deployment server we use MySQL while when developing it we used SQLite. For the purpose of this tutorial we propose to the readers to use SQLite [3] for reasons of simplicity. At the same time using this database requires no extra parametrization.

Install Python dependancies

We use `pip`[4] in order to install the dependancies of the platform. Having activated the isolated environment, type the following commands:

```
pip install requests==1.1.0
pip install decorator==3.4.0
pip install django-common==0.1.51
pip install django==1.4.5
pip install django_widget_tweaks==1.1.2
pip install djangorestframework==2.2.1
pip install south==0.7.6
```

[1] More information and an installation guide for `virtualenv` can be found in http://www.virtualenv.org/en/latest/virtualenv.html#usage.

[2] To find the Python version you use press in the terminal `python -V`.

[3] More information about SQLite is available at https://sqlite.org/

[4] http://www.pip-installer.org/en/latest/

The same result can be achieved by typing:

```
pip install -r requirements.txt
```

where `requirements.txt` contains the above list with packages and is provided in the "BioASQ_Area.zip". Each command installs a specific version of the modules, e.g. `pip install requests==1.1.0` will install the version 1.1.0 of the module `requests`.

Modifying the `settings.py` file

In this last step before launching the application in a python server we need to do a modification in the `setting.py` file available at `webexample\webexample\settings.py`. The file contains information about different settings concerning the platform e.g. paths to be considered when searching for templates/static files, database configuration etc. In this step you only have to modify the `myPath` variable in line 6, to indicate the absolute path of the webexample directory (e.g. myPath='/local/balikasg/platformInstall/webexample/'). At this point, every detail has been set. The only remaining thing is to initialise the database that the application will use. Note that in the `settings.py` file we provide two configurations: (i) a configuration with SQLite database and another one, commented, with MySQL. The configuration with the SQLite database is provided because of its simplicity for the illustrative purposes of the tutorial. In a deployment environment its use is discouraged for scaling and safety reasons.

Synchronizing the database

In order to built the database tables required by the application we need to execute:

```
python manage.py syncdb
```

This will create the required database tables along with a superuser account. Below you can see a sample output of the command:

```
python manage.py syncdb

Syncing...
Creating tables ...
Creating table auth_permission
Creating table auth_group_permissions
Creating table auth_group
Creating table auth_user_user_permissions
Creating table auth_user_groups
Creating table auth_user
Creating table django_content_type
Creating table django_session
Creating table django_site
Creating table django_admin_log
Creating table registration_registrationprofile
Creating table uploads_document
Creating table Test_detail
Creating table Test_article
Creating table Test_bioasq_baseline
```

```
Creating table Test_user_profile
Creating table Test_system
Creating table Test_test_result
Creating table Test_test_result_file
Creating table Test_upload_information
Creating table Test_eval_meas
Creating table forum_category
Creating table forum_forum
Creating table forum_topic
Creating table forum_post
Creating table south_migrationhistory
Creating table task1b_detail1b
Creating table task1b_golden_question_1b
Creating table task1b_user_results_1b
Creating table task1b_upload_information_for_1b
Creating table task1b_evaluation_measures_1b
Creating table task1bb_user_results_1bb
Creating table task1bb_evaluation_measures_1bb
Creating table oracle_eval_meas_oracle
Creating table oracle_log_oracle

You just installed Django's auth system, which means
you don't have any superusers defined.
Would you like to create one now? (yes/no): <--type "yes"
Username (leave blank to use 'balikasg'): test <--type a username
E-mail address: geompalik@hotmail.com example@mail.com <--
type your mail account
Password:  <--type a password but nothing will appear
Password (again): <--retype the password here, still nothing appears
Superuser created successfully.
Installing custom SQL ...
Installing indexes ...
Installed 0 object(s) from 0 fixture(s)

Synced:
 > django.contrib.auth
 > django.contrib.contenttypes
 > django.contrib.sessions
 > django.contrib.sites
 > django.contrib.messages
 > django.contrib.staticfiles
 > django.contrib.admin
 > registration
 > uploads
 > Test
 > webservice
 > rest_framework
```

```
 > forum
 > widget_tweaks
 > south
 > task1b
 > task1bb
 > oracle

Not synced (use migrations):
 -
(use ./manage.py migrate to migrate these)
```

Now we are ready in terms of the database. In the last step we will launch a local server to confirm that the application works.

Launching the application

You can start the application (platform) by typing:

```
python manage.py runserver
```

Output:

```
Validating models...

0 errors found
Django version 1.4.5, using settings 'webexample.settings'
Development server is running at http://127.0.0.1:8000/
Quit the server with CONTROL-C
```

Opening from a browser the http://127.0.0.1:8000/ will lead you to the homepage of the platform. The administator interface is at http://127.0.0.1:8000/admin/.

6.2 Modifying and adapting the BioASQ Participants Area

The BioASQ Participants Area was built to be generic and easily configurable so that it can be adapted to other domains without significant effort. Recall that the BioASQ Participants Area has been developed following the Model-View-Controller design pattern and as a result the Model (database), the View (graphical interface) and the Controller (functionality) of the software are loosely coupled and can be handled independently. Also, the different parts of the Controller are implemented in different subsystems of the software. Recall that there are seven subsystems implemented in the platform:

- Task A
- TaskB-PhaseA
- TaskB-PhaseB

} core functionality

- Registration
- Forum

- Contact Form
- Django-Admin

The first three implement the evaluation infrastructure and the result submission functionality. The rest offer additional functionality, mainly enabling the communication between the challenge participants and the BioASQ team as well as the administration of the challenge by the organisers. For more information on the design decisions for the development of the platform and a discussion on the advantages of those choices, please consult Balikas et al. (2013a).

Adapting the platform for a challenge in another domain requires minor changes, mainly to update the texts of the platform. This is due to the platform being domain agnostic and as a result the evaluation infrastructure and the results submission mechanisms do not need domain specific information. In detail, if one wants to start a challenge in a different domain, four of the subsystems, namely "Registration", "Forum", "Contanct Form" and "Django-Admin", don't have to be modified as their functionality is generic for each challenge.

The rest of the subsystems that implement the core functionality can also be used as is, if the structure of the other challenge remains the same. The evaluation measures for both tasks are domain agnostic. The scores are calculated by comparing the results submitted by the participants with the golden responses, which are stored in the platform database. Consequently, scoring with the same evaluation measures is feasible without any modifications. The same applies for the submission of results. Only the sanity checks performed on the submitted data may need to be modified. For example, in the semantic indexing task, where the challenge participants submit categories from a specified taxonomy, the platform should make sure the results belong indeed to the taxonomy.

Finally, changes may be required in the guidelines that describe the new challenge. Their current version reflects the specifications of the BioASQ challenge. Furthermore, the logos and the look-and-feel of the platform can be easily modified since they are written using HTML.

Bibliography

G. Balikas, I. Partalas, N. Baskiotis, E. Gaussier, T. Artieres, and P. Gallinari. Evaluation Infrastructure. Technical report, 2013a.

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. Technical report, 2013b.

G. Balikas, I. Partalas, N. Baskiotis, E. Gaussier, T. Artieres, and P. Gallinari. Evaluation infrastructure software for the challenges 2nd version. BioASQ Deliverable D4.7, 2014.

A. Kosmopoulos, I. Partalas, E. Gaussier, G. Paliouras, and I. Androutsopoulos. Evaluation Measures for Hierarchical Classification: a unified view through two generic frameworks. 2013.

P. Malakasiotis, I. Androutsopoulos, Y. Almirantis, D. Polychronopoulos, and I. Pavlopoulos. Tutorials and Guidelines. BioASQ Deliverable D3.7, 2013.

A.-C. Ngonga Ngomo, N. Heino, R. Speck, T. Ermilov, and G. Tsatsaronis. Annotation Tool. BioASQ Deliverable D3.3, 2013.