Asiya Web Service – User Guide

1.Introduction

The Asiya Web Service (AsiyaWS) is a RESTful web service to access the Asiya framework, an online tool to evaluate automatic translations. This web service allows for using Asiya from any remote client running on any platform. The service is asynchronous, i.e., the client launches an evaluation request and gets a token that is used to retrieve the results once they are available.

This document is a user guide to access the AsiyaWS available at UPC facilities[1]. It describes the service requirements and how to handle the possible errors.

2. User guide

Following the REST design model, the AsiyaWS uses HTTP methods to access the two services exposed. The first service, used to launch an evaluation request to Asiya, uses the POST method. The second service, used to download the results of the evaluation, uses a GET method.

2.1. Executing Asiya: `/asiyaws/main`

This service allows the user a) to upload the files that constitute the evaluation test-bed (e.g., source, reference and candidate translations), and b) to choose the parameters to execute Asiya (e.g., metric names, input/output format)[2]. The service executes Asiya and generates a file of results and an error file giving a description about what went wrong, in case the results are not as expected.

This operation consumes a single *HttpServletRequest* object having all the parameters and files required (see the list below) and returns a *String* as a response. This *String* contains either a

[1] http://asiya-faust.lsi.upc.edu.

[2] Consult the technical manual of Asiya for a complete list of the execution parameters. http://asiya.lsi.upc.edu/Asiya_technical_manual_v3.0.pdf

token identifier (which can be used later to retrieve the results) or an error message. All the errors start with the keyword “ERROR”.

Note that the service is asynchronous due the run-time that Asiya needs to finish the execution, especially in those cases where parsing is required. For this reason, the use of the token allows the clients to get their results later, as an independent HTTP request (see the next section for further details).

An error message will appear, instead of a *token*, in case that something goes wrong during the web service execution (instead, notice that errors in the Asiya execution are given in a error file along with the results).

The **input parameters** that the user must or may include in the *HttpServletRequest* object are listed below:

- **username** and **password**: The service requires authentication. To sign up to the service, they should contact asiya@lsi.upc.edu.

- **src**, **ref**, **trans**: The source, reference and candidate translations are required in raw format or XML (NIST format). They all must have the same number of lines. The source file (“src”) contains the text in the original language. If not source file is available, the user must provide a file having the same number of empty lines than the translations. At least one reference file must be provided (parameter name: “ref”). It contains a translation made by a human. At least one candidate file (parameter name: “trans”) must be also provided. This contains the automatic translations. Any number of reference and candidate translation files can be sent using always the same parameter name.

- **assessments**, **mod and tarscores**: Optionally, the user can also provide an assessments file (.csv file in NIST format) which can be used for meta-evaluation purposes or to learn a new evaluation model.
  The model learnt can be used in other evaluations. To upload the file it is used the parameter name “mod”.
  The scores calculated during the evaluation process are needed during the meta-evaluation task. In order to upload those scores (and avoid the burden of metric scores recalculation) the user can provide a .tar.gz file containing the scores obtained in a previous evaluation execution (parameter name: “tar”).

- **configFile**: It is also required that the user provides to the service the following parameters: the input format of the text files mentioned above, the source and target languages and the user preference about the use of case sensitive treatment. Optionally, the user can also specify one or more sets of metrics, systems or references. All these parameters can be provided either in a .config file (parameter name:

“configFile”) or directly along with the others parameters. In case of conflict, the last parameter uploaded will have higher priority. It is recommended that the config file goes first, so further updates of any parameter have preference over the config file. The next table presents the name of the parameters when they are passed directly through the *HttpServletRequest*:

PARAMETER	**PARAMETER NAME**
Input type	“input”
Source language	“srclang”
Source case treatment	“srccase”
Translation language	“trglang”
Translation case treatment	“trgcase”
Metric group name	“gName”
Metric	“metric”
System set name	“gName”
System	“system”
Reference set name	“gName”
Reference	“reference”

The first 5 parameters (input type, source language, source case treatment, translation language and translation case treatment) are unique, i.e., the user can only specify them once.

In contrast, the user can have several groups of metrics, systems and references, each having several metric, system and reference names respectively. The user can specify up to one group per type (otherwise the Asiya execution will fail). For each group, the user must first indicate a group name (parameter name: “gName”) and then indicate all the metrics, systems or references for the group. For instance gName=mymetrics; metric=BLEU; metric=NIST. To create a new group, the user has to simply add a new group name (gName), and then the previous group will be closed and the new one will be composed of the next names indicated.

- **tardata:** Optionally, you can upload a tarball containing all the source, references and translation files. In this case, the config file must include also the declaration of these documents (parameters “**src**”, “**ref**” and “**sys**”), and the user should omit to use the input parameters **src**, **ref**, **trans mentioned above.**

- **arg**: Finally, the service needs all the other Asiya options for the execution (parameter name: “arg”). These are the parameters that will define the execution on the Asiya framework. It works exactly as the Asiya technical manual says, although, the user should remember that it is executed remotely. In consequence, there is no need for using options like “-data_path”. Also, there is no need to write either the command name (“Asiya.pl”) nor the configuration file. Instead, say just the execution options, for instance: *-eval single,ulc -g all -m BLEU,-TER*

The **output parameters** of this method are the following:

- **token identifier**: If the AsiyaWS execution finishes correctly, the service will return a *String* with a token identifier.

- **authentication error**: If the authentication failed then the service will return a *String* informing the user about the issue.

- **other errors**: For any other type of error, the service will return a *String* informing about the issue. In that case, contact asiya@lsi.upc.edu and inform of the error and describe the execution parameters that you tried including the time of the execution.

2.2 Retrieving results: `asiyaws/main/query` and `asiyaws/main/download`

This second service is simpler than the previous one. Once a user has started an execution will get it is token as it is explained above. That token can be used to retrieve the results of that execution.

There are two different operations for this service. Both of them consume the same parameters, but while the one found at `asiyaws/main/query` returns a *String*, the one at `asiyaws/main/download` returns a *Response* object. This is made to simplify the possible development of a web client. The query service will check the authentication of the user and if the execution is done before presenting an html download link where it can be downloaded the .zip file with the results. The download service instead, will return a *Response* that will contain either an error message (starting with the “ERROR” keyword) or the .tar.gz file.

This service requires just three input parameters (all of them *Strings*):

- **username** and **password**: Give the same authentication than in the request.
- **token**: Give the token identifier obtained after the request.

The service returns the following output parameters:

- **authentication error:** If the authentication failed, the service returns a *String* informing about it.

- **execution not finished**: In case that Asiya has not finished its execution, the service will return a *String* message indicating that the execution is still not done

- **execution done**: After Asiya has finished, the service returns a *String* informing so. In that case, you can download the results.

- **results**: The service returns an Response containing an InputStream having the results. This is a .tar.gz that contains all the data and the results of the Asiya execution. Its content may depend on the type of operation requested:
    - In case of *evaluation or meta-evaluation*, the container will have a results file, in the format requested by the user and the scores folder.
    - If a *learn* operation was requested, then the result file will contain the model generated.
    - In both cases, the result file will contain also an error file showing if something went wrong during the execution of Asiya. Sometimes, the user will get a result even though there were errors. For that reason it is highly recommended that the error file is always checked to make sure the results are reliable.

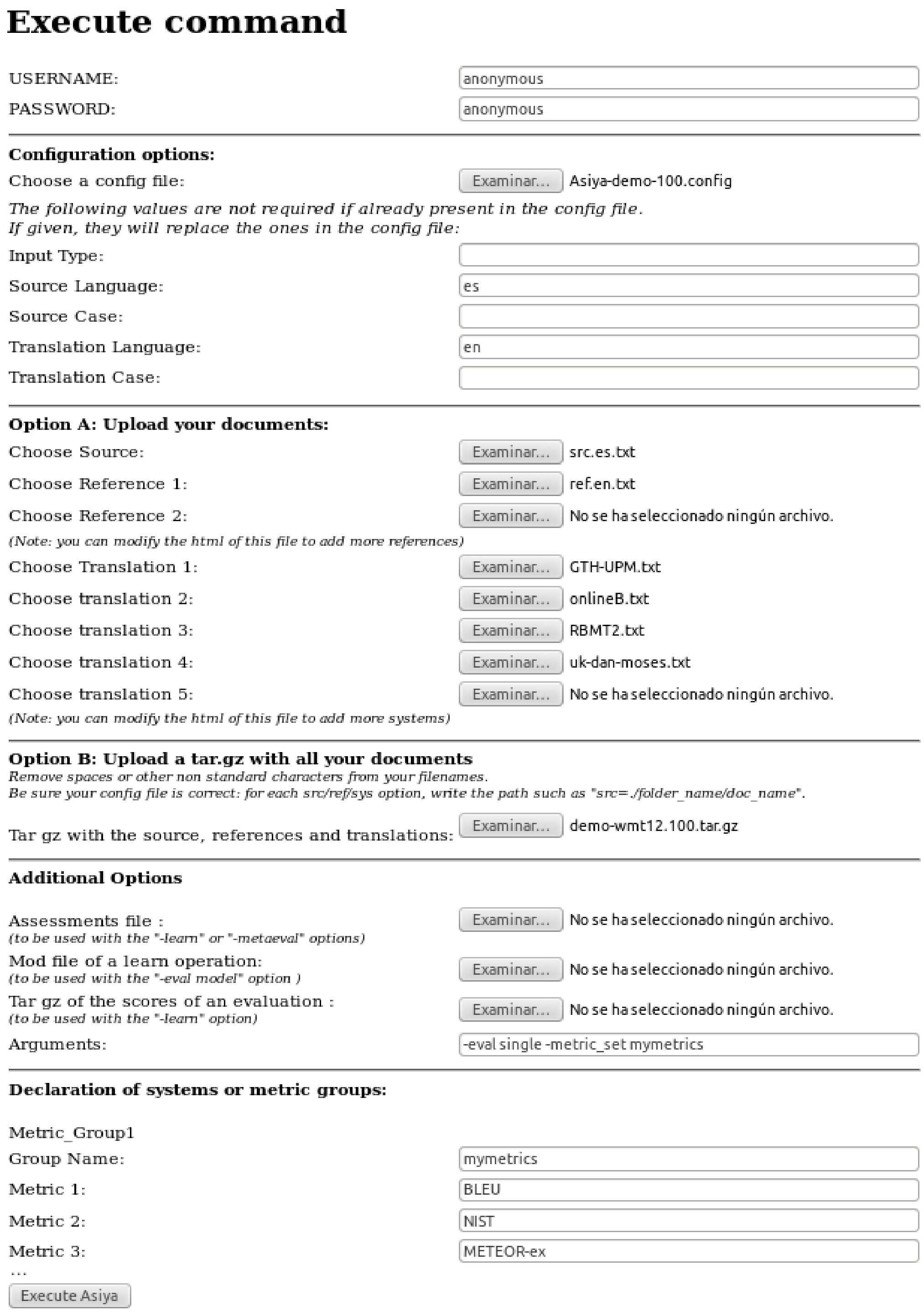
Note: once an execution is done, it will remain in the system to be retrieved for a week. After that, the system will delete it.

3. Example

The following example illustrates a simple evaluation execution using the data and the simple HTML client provided.

The folder “sample data” contains five text files: the source file in Spanish, a reference file in English and three candidate translations. Each file has three sentences, each in separated lines.

In order to evaluate these translations, open the file `clientUpload.html` with your web browser and fill the form as shown in the "Execute Command" figure below.



Execute command

USERNAME:	anonymous
PASSWORD:	anonymous

Configuration options:

Choose a config file:	Examinar... Asiya-demo-100.config

The following values are not required if already present in the config file.
If given, they will replace the ones in the config file:

Input Type:	
Source Language:	es
Source Case:	
Translation Language:	en
Translation Case:	

Option A: Upload your documents:

Choose Source:	Examinar... src.es.txt
Choose Reference 1:	Examinar... ref.en.txt
Choose Reference 2:	Examinar... No se ha seleccionado ningún archivo.

(Note: you can modify the html of this file to add more references)

Choose Translation 1:	Examinar... GTH-UPM.txt
Choose translation 2:	Examinar... onlineB.txt
Choose translation 3:	Examinar... RBMT2.txt
Choose translation 4:	Examinar... uk-dan-moses.txt
Choose translation 5:	Examinar... No se ha seleccionado ningún archivo.

(Note: you can modify the html of this file to add more systems)

Option B: Upload a tar.gz with all your documents
Remove spaces or other non standard characters from your filenames.
Be sure your config file is correct: for each src/ref/sys option, write the path such as "src=./folder_name/doc_name".

Tar gz with the source, references and translations:	Examinar... demo-wmt12.100.tar.gz

Additional Options

Assessments file : *(to be used with the "-learn" or "-metaeval" options)*	Examinar... No se ha seleccionado ningún archivo.
Mod file of a learn operation: *(to be used with the "-eval model" option)*	Examinar... No se ha seleccionado ningún archivo.
Tar gz of the scores of an evaluation : *(to be used with the "-learn" option)*	Examinar... No se ha seleccionado ningún archivo.
Arguments:	-eval single -metric_set mymetrics

Declaration of systems or metric groups:

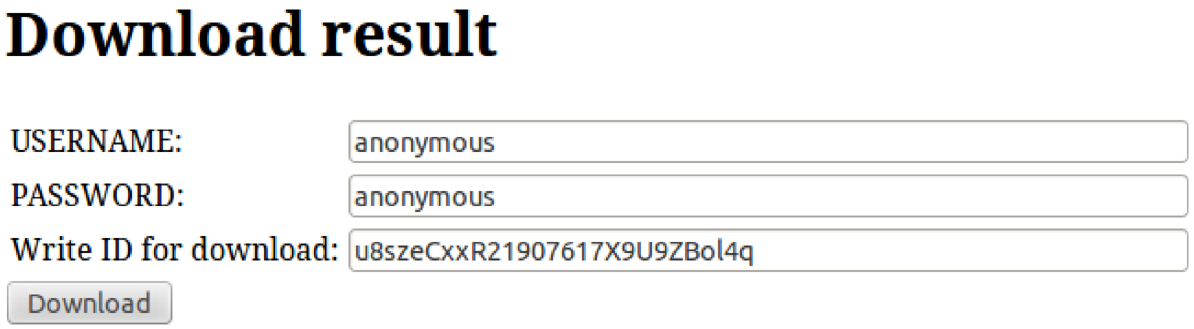
Metric_Group1

Group Name:	mymetrics
Metric 1:	BLEU
Metric 2:	NIST
Metric 3:	METEOR-ex

...

The execution will return the following message: `Your download token: u8szeCxxR21907617X9U9ZBol4q`

Then, in order obtain the results, open the file `clientQuery.html` with your web browser and fill in the form with the following data (change the token id with yours).

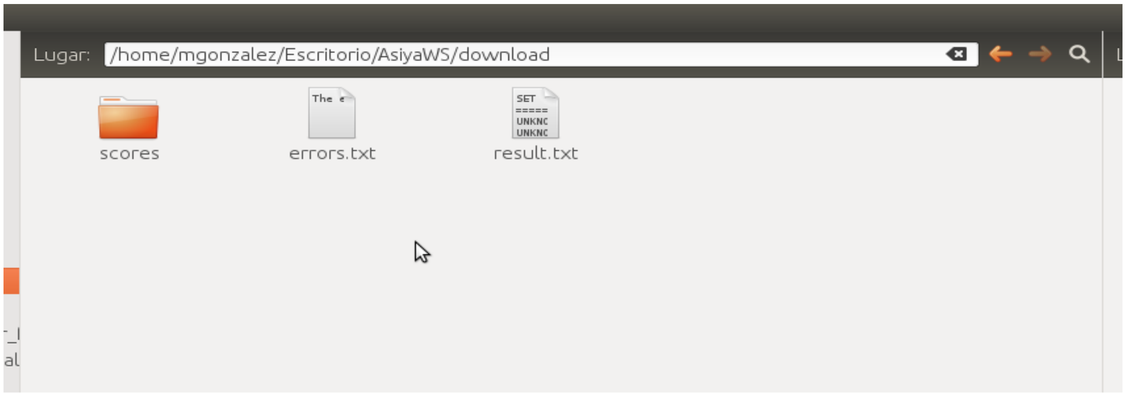


Download result

USERNAME:	anonymous
PASSWORD:	anonymous
Write ID for download:	u8szeCxxR21907617X9U9ZBol4q

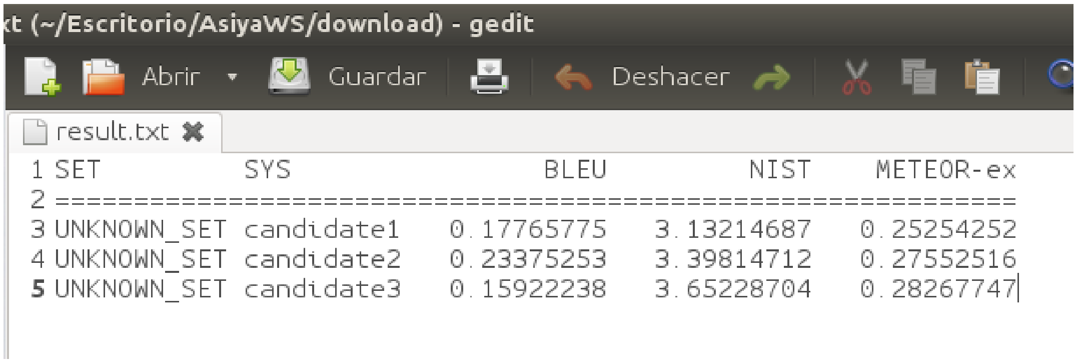
While Asiya is calculating you will receive the following message: `ERROR: The execution is not done yet.`

When all the evaluation results are available, you will receive the following message: `Your download link:` Download . Follow the link and save the file as `download.tar.gz`. This file has the following content (tar -xcvf download.tar.gz):



Lugar: /home/mgonzalez/Escritorio/AsiyaWS/download

If everything is alright, the errors.txt file contains the message: The execution had no errors. The results.txt file contains the metric scores:



```
SET          SYS                  BLEU        NIST     METEOR-ex
================================================================
UNKNOWN_SET  candidate1     0.17765775   3.13214687   0.25254252
UNKNOWN_SET  candidate2     0.23375253   3.39814712   0.27552516
UNKNOWN_SET  candidate3     0.15922238   3.65228704   0.28267747
```

And the scores folder contains all the scores calculated for all the systems, references and segments in XML format.