# Language Identifier Web Service

## Table of Contents

# 1 BASIC INFORMATION

### 1.1 Tool name

The tool is called **LangId** which is the short for “Language Identifier”.

### 1.2 Overview and purpose of the tool

LangId identifies the language of a written raw text. It is developed in C# and it is encapsulated as a SOAP web service which is WSDL described at:

http://www.racai.ro/webservices/LangId.asmx?WSDL

LangId is able to identify 54 commonly known languages (*Afrikaans*, *Alemannic German*, *Arabic*, *Azerbaijani*, *Bavarian*, *Belarusian*, *Bosnian*, *Breton*, *Bulgarian*, *Catalan*, *Chinese* (*Standard*), *Croatian*, *Czech*, *Danish*, *Dutch*, *English*, *Esperanto*, *Estonian*, *Filipino*, *Finnish*, *French*, *Galician*, *German*, *Greek*, *Hebrew*, *Hungarian*, *Indonesian*, *Irish Gaelic*, *Italian*, *Japanese*, *Korean*, *Latin*, *Latvian*, *Lithuanian*, *Maltese*, *Norwegian*, *Occitan*, *Polish*, *Portuguese*, *Romanian*, *Russian*, *Serbian*, *Serbo-Croatian*, *Sicilian*, *Slovak*, *Slovene*, *Spanish*, *Swedish*, *Thai*, *Turkish*, *Ukrainian*, *Volapük*, *Welsh*, *Yiddish*) and 3 rare languages (*Aweti*, *Beaver*, *Teop*).

### 1.3 A short description of the algorithm

The algorithm and the web service are thoroughly described in Tufiş et al. (2008) and Ştefãnescu (2010). The algorithm implements a statistical solution which requires for training raw balanced reference texts for each of the languages one intends to identify. In the training phase, the application builds a model for each language of interest, which contains the distribution of affixes and small words in the reference texts corresponding to that language. In the prediction phase, when a new, unseen text is presented, the tool generates a similar model for it and then compares this model with the reference models learned in the training phase. The application computes a similarity score for each language and the language code corresponding to the highest score is returned as output.
In the training phase, for each of the 57 languages, we used Wikipedia documents containing 3 to 10 Mb of raw text.
For the moment, the web service is a quite slow since it is necessary to compare the models for 57 language pairs in order to identify the language. In the near future we plan to optimize it, in order to run faster.

# 2 TECHNICAL INFORMATION

### 2.1 Software dependencies and system requirements

This tool is available as a web service and therefore all computations are performed on the machine hosting the service. Consequenly, there are no system requirements for the end-user.

## 2.2 Installation

There is no installation required for this tool.

## 2.3 Execution instructions

After adding the Web Service as a web reference, identifying the language of a text can be done using the following lines of code (this is a C# example; see *Program.cs*):

```
LangIdWebService langIdService = new LangIdWebService();
string text = DataStructReader.readWholeTextFile("input.txt", Encoding.UTF8);
LangIDResult result = langIdService.IdentifyLanguage(text, true, false);
```

## 2.4 Input / Output data formats

The input should be a UTF-8 raw text string. The output is a *LangIDResult* object containg:

- the native name of the identified language (.*LanguageNative*)
- the English name of the identified language (.*LanguageEn*)
- ISO 639-1 code of the identified language (.*Language*)
- the confidence score for the prediction, which is a *double* value between 0 and 100 (.*Confidence*)

## 2.5 Integration with external tools

LangId is fully self-contained.

# 3 CONTENT INFORMATION

This Web Service can be tested by using the example provided in the ‘`test/`’ directory, or by using the web application available at:

http://www.racai.ro/webservices/LangId.aspx

For testing using the local provided executable (in ‘`test/`’ directory), it is necessary to have .Net Framework 4 installed.

*Program.cs* contains a C# example for using the web service.

## 3.1 Test input files

See the testing kit in the ‘`test/`’ directory. The input file is ‘`input.txt`’ which contains a raw Latin UTF-8 text.

In case of using the web application, the input text should be UTF-8 encoded and it should be inserted into the ‘*Input text:*’ textbox.

### 3.2 Output files

There is no output file. The test application will display results at Standard Output.

### 3.3 Running times

LangId Web Service is hosted on a 4-core Intel(R) Xeon(R) x86 CPU @ 3.20GHz and 2 GB of RAM. The usual running time for identifiying the language of a text (as in the provided example) is about 13.5 seconds.

## 4 ADMINISTRATIVE INFORMATION

### 4.1 Contact

For further information, please contact Dan ŞTEFĂNESCU (http://www.racai.ro/~danstef/; danstef@racai.ro).

## 5 REFERENCES

Tufiş, D., Ion, R., Ceauşu, A., Ştefănescu, D. (2008). RACAI's Linguistic Web Services. In Proceedings of the 6th Language Resources and Evaluation Conference - LREC 2008, 28-30 May 2008, Marrakech, Morocco

Ştefănescu, D. (2010). Intelligent Information Mining from Multilingual Corpora. PhD thesis (in Romanian). Romanian Academy, Bucharest.