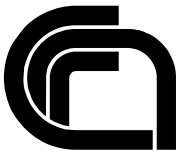

Consiglio Nazionale delle Ricerche

WLI: a Web application for Language Identification and evaluation of available tools

A. Marchetti, M. La Polla, C. Bacciu, M.Abrate

IIT TR-18/2012

Technical report

Dicembre 2012

Istituto di Informatica e Telematica

WLI: a Web application for Language Identification and evaluation of available tools

Andrea Marchetti, Mariantonietta N. La Polla, Clara Bacciu, Matteo Abrate

Institute of Informatics and Telematics
National Research Council (CNR), Pisa, Italy
firstname.surname@iit.cnr.it

Abstract. Web Language Identifier (WLI) is a service that, starting from the URL of a Web page or a plain text and exploiting a pool of language identification tools, returns a set of candidate languages with a confidence score. Currently embedded tools are Chromium Compact Language Detector, Lingua::Identify, and a simple one based on HTML attributes. The service can be exploited through a Web application or via an API. To globally evaluate the identifiers, we constructed a test set of Web pages extracted from 146 Wikipedia projects. This allows using WLI also as a service to compare language identification tools in terms of supported languages and precision of the results. The charts summarizing the comparison can be visualized in the WLI Web application. We plan to extend the service making it possible for the users to add their own identifier.

Keywords: Language identification, Web tools for languages,Multilingual Web, Web application

1 Introduction

In the context of Multilingual Web one of the most challenging task is the identification of the language of a Web page. Different tools, exploiting different identification techniques, are available on the Web.
Problems related to this plethora of tools comes from multiple aspects. First of all, the major part of available tools works only with text extracted from the considered Web page: typically, if a user wants to know which is the language of a Web page has to copy some text from the page itself, paste the text in the tool and run the evaluation. Detecting the language of a Web page given its URL can be more useful in some tasks, such as the crawling a certain amount of pages in a given language. In these situations, providing the URL can avoid the waste of time of downloading the pages without first knowing if they are useful or not. Secondly, it is not possible to know, without testing, which languages, among those that the tools declare to recognize, are really detected. Third, it is not possible, in nearly every case, to understand the precision of the results. This implies that we cannot compare the results obtained exploiting different

tools against the same text. To handle with some of these problems, it could be useful a service that works with both URLs and plain text and that collects some statistics about the results provided by different tools: this allows the comparison of tools for language identification. Web Language Identifier (WLI) is a service that allows users to detect the language of a Web page using a plain text or the URL of the page itself. The URL is used by WLI to automatically analyze the text, without the necessity, for the user, to crawl the page. Furthermore, WLI allows users to test different tools against a **unique** data set of languages on the Web. The objective of the testing is twofold: analyze how the tools work with languages that they declare to recognize and compare the results of different tools applied to the same data set. One of the key features of WLI is the test set: to perform the testing, we constructed a large dataset based on 146 different Wikipedia Projects.

The paper is structured as follow: Section 2 briefly discuss about the languages in the world and provides some data and statistics on the presence of languages on the Web. Section 3 briefly describes related works. Section 4 surveys the existing used practices, tools on language identification according to different metrics. Section 5 describes our service, WLI: we will describe how we built our test set, then we will present the experiments performed with some available tool for language identification and we discuss the results. Section 6 draws the conclusion and counts future work.

2 Languages and Web

The answer to the question "how many languages are there in the world" is far from being a simple one, since it heavily depends on the context in which the question is asked. The main problem is that it is hard to define what a language is. On one hand, theoretical linguistic studies tend to base the definition on internal structural criteria such as morpho-syntactic and lexical distance; often, historical linguistic reasons are also taken into account. On the other hand, socio linguists instead tend to rely more on external clues, such as domains of use, degree of standardization, existence of a written form, status of official language (being taught at school) in one or more countries. Ethnologue [1] is the most authoritative catalogue available for world languages. It currently lists 7,413 languages. The ISO 639-3:2007 Code for the representation of names of languages includes the current list from Ethnologue, plus some ancient and artificial language (7,589 entries). Moreover, according to an estimate from the Google Books project[1], the number of languages in which a book was ever published is only 480.

These data give us a very contradictory picture; published books appear only in highly standardized written languages, yet the number of partial Bible translations let us believe that a much larger number of languages may be written in other contexts, such as the Web [2].

The count of languages on the Web depends on the definition of language, but also on the definition of what kind of documents we look at. The fact that a

[1] http://books.google.com/

language is present on the Web does not make it automatically relevant for a Web language identifier. For instance of the 4,626 languages that are documented in Open Language Archives [3], 3,930 have online resources; the ODIN project[2] harvests examples of interlinear glossed text from linguistic papers, and has over 1,250 languages in its database. Yet, most of these languages may have be just present on one or two pages of grammatical descriptions or linguistic documentation. In a very interesting blog post on the subject[3] Kevin Scannell introduces the concept of primary text as opposed to language description. Primary texts are newspapers, blog posts, Wikipedia articles, Bible translations, etc. This definition of primary texts matches with the concept of language by development as introduced by Kloss in [4] and adopted by Ethnologue. We claim that a Web language recognizer should be able to deal with all languages present with primary texts on the Web. So far Scannell's web crawler has found and documented 1000 language that are present on the Web with primary texts and estimates an overall figure of around 1,500 languages present on the Web with such texts.

3 Related Work

3.1 Detecting language of a Web page

A good survey on existing techniques for language identification is in [5]. Typically, proposed solutions exploits some features that can be extracted from the page. In [6], authors perform language identification applying a variety of machine learning algorithms at three different data set in five languages: English, French, German, Spanish and Italian. A mapping between both training and data set to numerical features vectors is firstly proposed. For the extraction of the features three different methods are used: words as features, trigrams as features and custom-made features (such as Country code top-level domain - ccTLD).

In [7] an approach based on n-grams (see Section 4 for further details) is proposed. the n-gram based algorithm is complemented with heuristics and a similarity measure. Authors assign language labels to textual strings based on a statistical characterization of text in terms of its constituent n-grams. A tentative to adapt the language identification method, firstly proposed in [8], for use on a Web corpus is in [9].

3.2 Comparison of language identification tools

In literature we can find some attempts of comparison of language identification techniques or algorithms. For instance, in [10] two experiments are presented that compare different language identification algorithms. Authors perform an evaluation of the results retrieved from a two-step process: the generation of a document and a language model, that exploits different approaches (e.g. n-gram,

[2] http://linguistlist.org/projects/odin.cfm
[3] http://indigenoustweets.blogspot.it/2011/12/1000-languages-on-web.html

etc.) classification method, and the language identification on the basis of the language model. Another example of comparison is presented in [11]. Authors focus their attention on three different statistical language identification methods: Markov Models, Trigram Frequency Vectors, and n-gram text categorization. The objective of the comparison is the study of the influence on those systems of some basic parameters such as the size of the train set, the amount of text to classify and the languages the system is able to distinguish. Corpora for six different languages have been used in the experiments. Language identification that exploits Markov models is studied also in [12]. The studied approaches both deal with the incoming text at the character level. The final goal of the study is to define the precision of the results in language identification task of selected methods and to compare them. Experimental evaluation was based on large-scaled Multilingual Reuters Corpus with various European and Slavic languages. Interesting is also the comparison presented in [13]. In this case the focus is on the length of the analyzed text. Authors compare the performance of some typical approaches for language detection on very short, query-style texts.

Limitations of above-mentioned solutions can be pointed out in different aspects:

- *language coverage*: the test set is often composed by "frequent languages", moreover corpora include a limited numbers of languages;
- *experiments environments*: to performs their analysis, authors manually run the different approaches;
- *nature of corpora*: solutions presented are typically developed using long and articulated documents. This is not often the case of document retrieved on the Web.

4 State of the Art

Language identification is a very important step in several Natural Language Processing applications [14]. In this section we will briefly summarize some common techniques used to identify the language of a text. Then we will present current tools for language identification. We discuss about language identifiers referring, as in [15], to software for the automatic recognition of the language of a document; in particular we refer to electronic documents.

Various approaches have been presented in literature. All approaches work in the same manner: they apply a language identification method, starting from a language model [10]. A language model is a collection of information about the languages to be identified that the algorithm compares with the text to be analyzed. The most used approaches to build a language model are:

- **short word-based**: uses words up to a specific length to construct the language model, independently from the particular word frequency.
- **frequent word-based**: generates a language model using a variable amount of words having the highest frequency of all words occurring in a text.

– **n-gram-based**: uses substring of n characters to provide an evidence of the language.

The n-gram approach is the most used. It is based on the idea that, in every language, there are n characters that are more frequent than others. The length of these substrings, called n-grams, can be variable, like in [8], or fixed, as in [2] and [16]. Moreover, instead of words, to determine the length of n-grams sequences, of bytes can be used [17]. In order to build an n-gram model for a given language, the frequencies of all n-grams are retrieved in large corpora of text. Then, to identify the language of a document, the n-gram profile is calculated and compared to language specific n-gram profiles. The language profile which has the smallest distance to sample text n-gram profile indicates the language.

When the language model is defined, different classification approaches can be used for language identification.

The statistical approach, introduced in [17], is one of the most important techniques in language identification. This approach is based on Markov Chains in combination with Bayesian Decision Rules. Markov Chains are used to construct the language model of every language. A transition matrix contains the probability of occurrence of each string for each model. Then the algorithm calculates the probability that a document derives from one of the existing language models. A variant of this method is proposed in [2]. The algorithm rely on a database in which a list of the highly frequent small words of a language is stored.

Another approach is using n-grams. The algorithm proposed in [8] exploits overlapping n-grams, with n=1-5. Initially the document is analyzed in order to eliminate all punctuation marks and each word is treated as a token delimited by white spaces. All tokens are scanned and n-grams are produced. The n-grams are stored in a hash and for each occurrence the counter for the n-gram in question is increased. The n-gram hashes constitute the n-gram profiles for each language.

The Vector Space Model is based on the similarity computation via the cosine distance between the training and test language model. An example of this approach is in [16] in which the numerical values within one vector are defined by a token's occurrence in the training set (times of its inverse document frequency). The proposed solution can identify also if a document is in two or more languages, without incurring any appreciable extra computational overhead.

The Monte Carlo technique, presented in [18], uses dynamic models to classify the documents language, instead of the necessary generation of language models. The models of the language are built by randomly selected features. The process of feature's selection is executed until an adequate amount of features to determine the entire documents language is reached. This amount is calculated using the standard error.

To classify a language, the approach proposed in [5], uses the Relative Entropy to compute the similarities between language models based on the amount necessary encoding information for a second language model given a first one. The language models describe probability distributions.

The Ad-Hoc Ranking method, first introduced in [8], is based on the comparison of two models ranked in descending frequency. Text features are extracted

into a document model from every unclassified document and into a language model from training data. Then, all features are sorted by their descending frequency (rank).

4.1 Web language identification tools:

Table 1 shows currently used tools for language identification, available on the Web. For our service we chose, among the above mentioned tools, Chromium Compact Language Detector and Lingua::Identify. We also developed a very simple identifier that exploits the "lang" HTML tag (for further details see Section 5).

Tools	Detected languages	Type of identification	License
Chromium CLD	160	4-gram	Open source
Language Detection	53	Naive Bayesian filter	Open source
Lingua::Identify	33	Small Word Technique Prefix Analysis Suffix Analysis n-gram Categorization	Open source
Language Identification for Python (LID)	n.a	3-gram	Open source
Xerox Language Identifier	46	n.a.	Open source
Rosette Language Identifier	55	morphological approach	Commercial
Lingua Systems LID	45	Statistical approach	Commercial
Sematext Language Identifier	n.a.	Statistical approach	Commercial
Alchemy API	97	n.a.	Commercial
Semiocast	61	Tokenization	Commercial

Table 1. Web language identification tools

Chromium Compact Language Detector: Google's Compact Language Detector is an open source library partially embedded in the Google Chrome browser to detect the language of any UTF-8 encoded content. Developed as separate Google code project, CLD is is distributed as an open source software[4] and can be used directly from any C++ code. CLD uses language models based on 4-grams. The training set was based on Wikipedia; their sources of language data used to fine-tune the algorithm included the BBC and Watchtower.org, the Web site of the Jehovah's Witnesses. Currently the Google's Compact Language Detector declares support for about 160 languages, out of which 52 are currently implemented in the Chrome browser.

[4] http://code.google.com/p/chromium-compact-language-detector/

Lingua::Identify: Lingua::Identify[5] is an open source identifier written in Perl. The identification of the language is performed using four different methods of language identification:

- *small words*: searches the most common words (e.g. articles, pronouns, etc.) of each active language in the text.
- *prefix analysis*: analysis of the common prefixes of each active language. The size of the prefix can be 1, 2, 3 and 4 characters.
- *suffix analysis*: analysis of the common prefixes of each active language. The size of the suffix can be 1, 2, 3 and 4 characters.
- *n-gram*: uses sequences of substring (including spaces) of size 1, 2, 3 and 4.

The identifier is a module, so it is possible to integrate it in any application. At the moment, Lingua::Identify can detect 33 languages.

HTML lang: We also embedded a simple script that returns the value of the first lang attribute of the page (if any). When the attribute is set properly, as it often is for example when the page comes from a CMS, it is obviously very reliable for language identification.

5 Web Language Identifier - WLI

We propose a system that handles with the above mentioned open problems in Web language identification. Our service, named WLI, has two main functionalities:

- detection of the language of a Web page
- comparison of language identification tools

5.1 Detection of the language of a Web page

Figure 1 shows the Web interface of WLI.

The first functionality works with both URLs and plain text and exploits three tools that are currently embedded in the service. In the main screen there is a field for entering the URL of the page that the user wants to analyze. Pressing the `Load` button, the HTML code is displayed in a text box below. Clicking on the `Identify` button, the page is sent to the three identifiers and their output is displayed: the number of languages that match HTML "lang" attribute (if any), the outcome of CLD, and the one of Lingua::Identify. Both CLD and Lingua::Identify return also a list of other "possible" languages, according to a confidence score. The user can also provide some text that she wants to analyze, using the text box: the text can be a plain text or HTML code. REST APIs that allow the identification are also available. Sending a GET request to `http://wafi.iit.cnr.it/multilingualweb/api/retrieve+identify/<URL>`

[5] http://search.cpan.org/ ambs/Lingua-Identify-0.51/lib/Lingua/Identify.pm

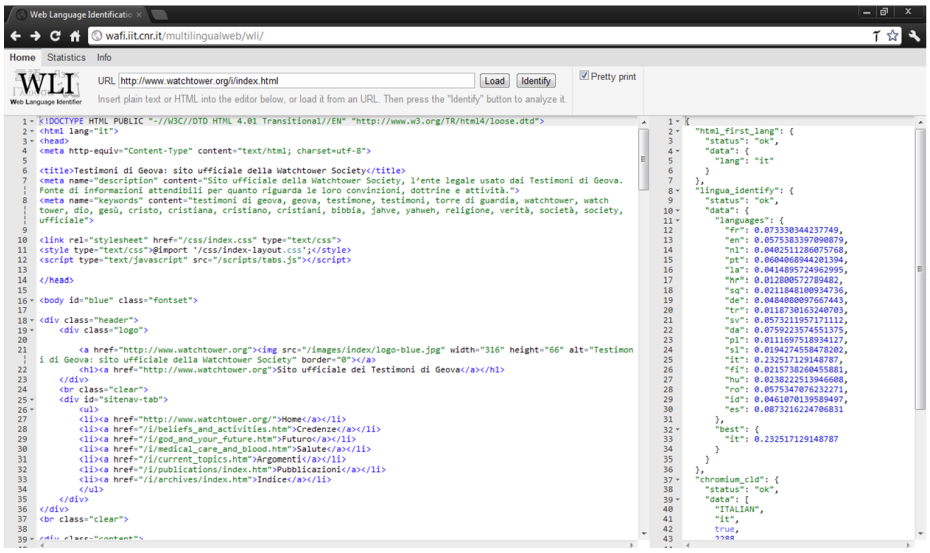

Fig. 1. WLI Web interface

or a POST at `http://wafi.iit.cnr.it/multilingualweb/api/identify` with the body containing the plain text or the HTML to be identified, WLI returns a JSON object which has a key for each used language identifier. Each key refers to an object containing the execution status of the identification (OK or ERROR), and if the status is ok, the key "data" points to an object reporting the output of the identifiers.

5.2 Comparison of language identification tools

The second functionality collects the results from the embedded language identification tools and provides statistic about their precision. One of the key features of our solution is the used test set: we built a multilingual corpus extracting pages from 146 different Wikipedia projects. We felt the need to globally evaluate the used language identifiers on a common set of web extracted pages that is vast enough in terms of number of pages per language and in terms of number of represented languages. Thus what we needed was a multilingual resource that

- is available online;
- covers as much languages as possible;
- contains a large amount of documents for each language.

We are aware of the presence online of many publicly available multilingual resources such as: the first article of the Universal Declaration of Human Rights[6],

[6] http://www.un.org/en/documents/udhr/

available online in 379 languages, the Bible translations[7], Watchtower[8](the Official Web site of Jehovah's Witnesses, in 366 languages) and Wikipedia[9] with 283 languages. We also analyzed The Rosetta Project[10], the Open Language Archives[11]and the Project Gutenberg[12] resources.

We also surveyed different recent multilingual Web corpora. Among these, we can mention,

- Corpus Factory: constructed in 2010, it contains 8 languages. [19]
- the Crúbadán Project, [20], containing texts in 487 languages in the version 1.0 and 1023 languages in version 2.0.
- I-X [21], that contains texts in 6 languages
- WaCky, [22], that contains texts in 3 languages.
- W2C that contains 120 languages [23].

To create our own corpus we focused our attention to the Wikipedia projects due several reasons such as the ease of access to the documents (web APIs), the number of represented languages, and fact that each document is given an identification string for its language. The string is saved in the "lang" HTML attribute of the page and is the same that is used as prefix for the Wikipedia project (e.g. en.wikipedia.org and en for English, it.wikipedia.org and it for Italian, etc.).

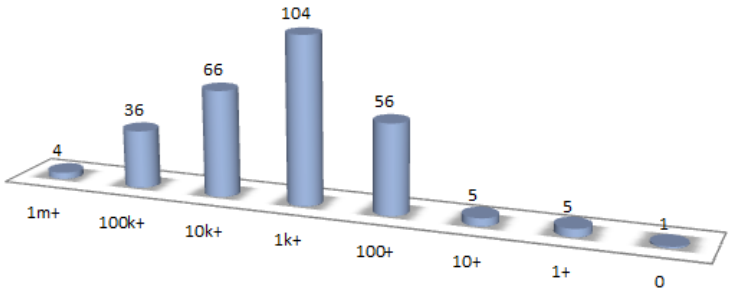

Fig. 2. Wikipedia articles per languages

A mapping table, showed in Figure 3, was then compiled, containing the following information:

- language prefix of the Wikipedia project (e.g. 'en' in en.wikipedia.org, 'pl' in pl.wikipedia.org)

[7] http://www.biblegateway.com/versions/
[8] http://www.watchtower.org
[9] http://www.wikipedia.org
[10] http://rosettaproject.org/
[11] http://www.language-archives.org/
[12] http://www.gutenberg.org/

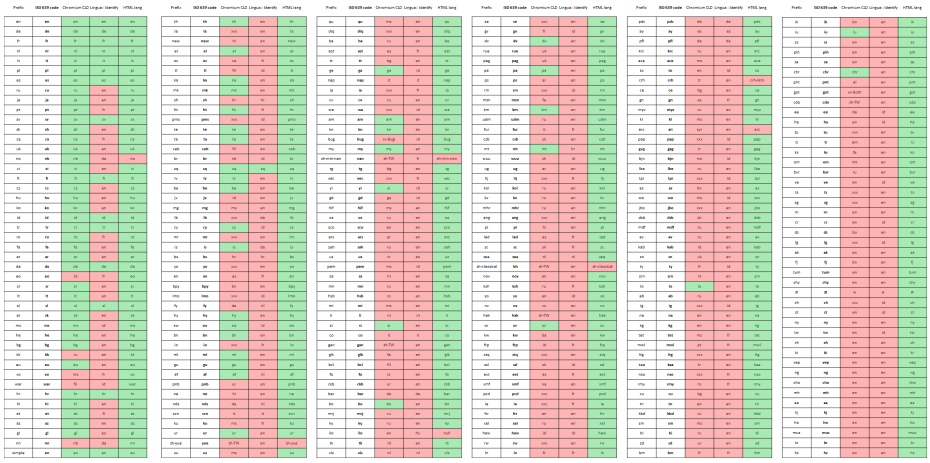

Fig. 3. Mapping table

- language name in English chosen by the Wikimedia community for the project (e.g. 'Polish')
- local language name chosen by the Wikimedia community for the project (e.g. 'Polski')
- code for the HTML "lang"attribute, extracted from the Wikipedia project's main page
- official ISO 639 code
- official ISO 639 reference name for the language

The mapping between the Wikipedia prefix and the ISO 639 code is not always straightforward, and we had to manually check each correspondence and correct inconsistencies. For example the prefix "simple" is for "simple English" and we mapped it to the ISO 639 code "eng". This table is available in the application (section Info) to be browsed in a human readable form, and can also be retrieved in JSON format. We needed this mapping because the identifiers we used return a ISO 639 code as response (in addition to the string containing the name of the language).

Not all 283[13] Wikipedia projects contain a sufficient number of articles: different projects are being closed or are very small in terms of contatined articles. We thus considered only Wikipedia projects having more than 3,000 articles (146 languages). For each of the selected projects, we acquired, using MediaWiki APIs, a pseudo-random sample of at least 1000 articles. Figure 2 shows the Wikipedia projects grouped by the number of articles.

To evaluate the embedded language identification tools, we run the identifiers on the pages of the corpus and we stored the output. Using the mapping table, we compared each output with the language associated to the Wikipedia. For

[13] http://meta.wikimedia.org/wiki/List_of_Wikipedias

Lingua::Identify the output can be correct, wrong or an exception (generally due to memory issues), while for CLD it can be correct, wrong, unknown[14] or not mappable (a non valid ISO code). For our extractor of the "lang" attribute the output can only be correct or wrong. Interactive charts showing the results are available in the application. They show the behavior of Lingua::Identify and CLD for each Wikipedia and the user can filter the results by type (correct, wrong, unknown, etc.) For a further analysis, for each type of answer of each language, the user can click the bar of the chart and visualize a table containing the analyzed pages showing the Wikipedia prefix, the title of the page and the output of the identifiers.

5.3 Results

Figure 6 shows the evaluation results of CLD and Lingua::Identify. Figure 4 is referred to CLD. If we consider that the result it is correct when at least 50% of Wikipedia pages for a given language are correctly categorized, as we can see from the figure, among the declared languages, almost the 46% is correct. Languages well detetected are those with a specific script, such as Armenian, Japanese or Hebrew. If we consider 10% of unknow value as limit, there is 34.4% of languages for which the identifier reports an `UNKNOW` value as result. If we augment the number of pages correctly categorized, the perfomances of CLD decrease: with the 90% of pages detected, just the 26,7% of languages are deteted.

For waht concerns Lingua::Identify, as show in Figure 6, the 8,3% of languages in the corpus is correctly identified with a precision of 50%. If we set the precision to the 90%, the result is correct in the 3.5% of cases. This lower percentages are related to the small number of languages with which Lingua::Identify works. Even if the number of declared languages is 33, just 19 languages are identified against the set (with different percentages of precision).

6 Conlusion and Future Works

In this work we presented WLI, a Web service for language identification and identifiers comparison. The service can be used as a service for language identification, providing the URL of the considered page, or the plain text. The service analyzes the text with three different identifiers: Chromium Language Detector, Lingua::Identify and HTML tags. The first two tools, are available online; the third one was implemented by our group in order to exploit meta HTML tags of Web pages. WLI can be also used as service of comparison for language identification tools. We tested the above mentioned tools against the same test: a Web corpus built on 146 different Wikipedia projects. At the moment, CLD and Lingua::Identify were embedded in WLI manually: we plan to extend the functionalities of the service in order to allow the automatic integration of others tools by users.

[14] `UNKNOWN_LANGUAGE` is returned if no language's internal reliability measure is high enough. This happens with non-text input such as the bytes of a JPEG, and also with some text in languages outside the set of supported languages

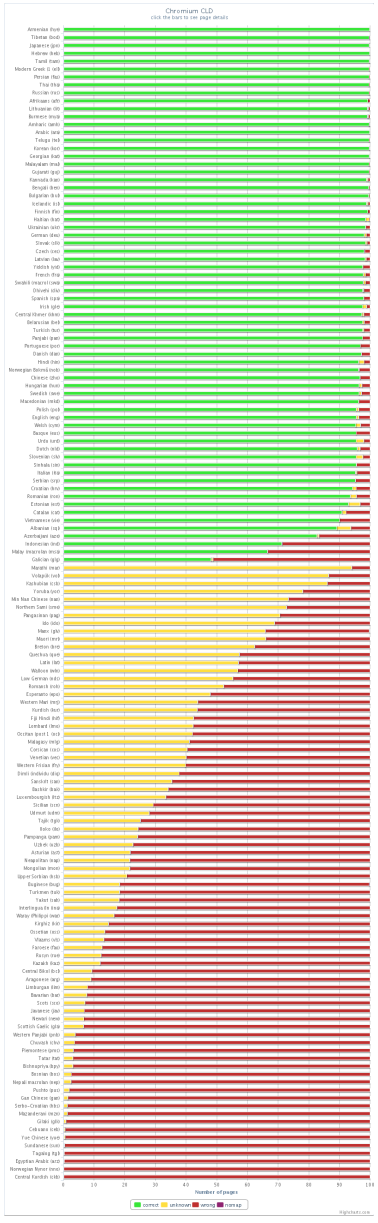

Fig. 4. Chromium Compact Language Detector results

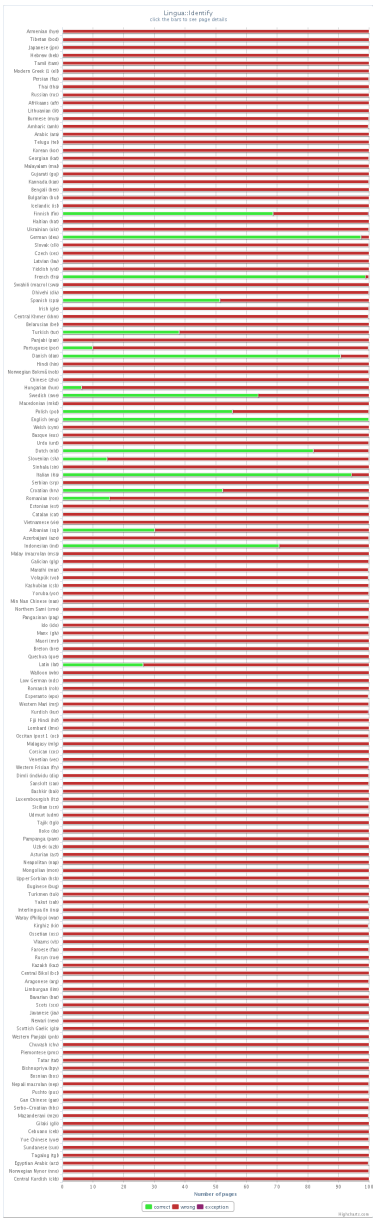

Fig. 5. Lingua::Identify results

Fig. 6. Evaluation Results

Acknowledgments

This work has been supported by EU FP7 Project CAPER (Grant Agreement no. FP7-261712) and by EU FP7 Project Multilingual Web (ICT PSP Grant Agreement No. 250500)

References

1. Lewis, M.P.: Ethnologue: Languages of the world (2009)
2. Grefenstette, G.: Comparing two language identification schemes. In: Third International Conference on the Statistical Analysis of Textual Data (JADT 1995). (1995)
3. G., S., S., B.: The open language archives community: An infrastructure for distributed archiving of language resources. Literary and Linguistic Computing **18**(2) (2003) 117–128
4. H., K.: Abstand languages and ausbau languages. Anthropological Linguistics **9**(7) (1967) 29–41
5. Sibun, P., Reynar, J.: Language identification: Examining the issues (1996)
6. Baykan, E., Henzinger, M., Weber, I.: Web page language identification based on urls. Proc. VLDB Endow. **1**(1) (August 2008) 176–187
7. Martins, B., Silva, M.J.: Language identification in web pages. In: Proceedings of the 2005 ACM symposium on Applied computing. SAC '05, New York, NY, USA, ACM (2005) 764–768
8. Cavnar, W.B., Trenkle, J.M.: N-gram-based text categorization. In: In Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval. (1994) 161–175
9. Hayati, K.: Language identification on the world wide web (2004)
10. Lena Grothe, E.W.D.L., Nrnberger, A.: A comparative study on language identification methods. In: Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08), Marrakech, Morocco, European Language Resources Association (ELRA) (may 2008)
11. Padro, M., Padro, L.: Comparing methods for language identification (2004)
12. Vojtek, P., Bielikov, M.: M.: Comparing natural language identification methods based on markov processes. In: In: Slovko, International Seminar on Computer Treatment of Slavic and East European Languages. (2007)
13. Gottron, T., Lipka, N.: A comparison of language identification approaches on short, query-style texts. In: Proceedings of the 32nd European conference on Advances in Information Retrieval. ECIR'2010, Berlin, Heidelberg, Springer-Verlag (2010) 611–614
14. S., K.: Evaluation of language identification methods (2006)
15. Langer, S., Gmbh, E.: Natural languages and the world wide web (2001)
16. Prager, J.: Linguini: Language identification for multilingual documents. In: Journal of Management Information Systems. (1999) 1–11
17. Dunning, T.: Statistical identification of language. Technical report (1994)
18. Poutsma, A.: Applying monte carlo techniques to language identification. In: In Proceedings of Computational Linguistics in the Netherlands (CLIN, Rodopi (2001) 179–189

19. Kilgarriff, A., Reddy, S., Pomiklek, J., PVS, A.: A corpus factory for many languages. In: Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10), Valletta, Malta, European Language Resources Association (ELRA) (may 2010)
20. Scannell, K.P.: The crbadn project: Corpus building for under-resourced languages (2007)
21. Sharoff, S.: Creating general-purpose corpora using automated search engine queries. In: WaCky! Working papers on the Web as Corpus. (2006)
22. crawled Corpora, W., Baroni, M., Bernardini, S., Ferraresi, A., Zanchetta, E.: The wacky wide web: A collection of very large linguistically processed (2009)
23. Majli, M., abokrtsk, Z.: Language richness of the web. In: Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12), Istanbul, Turkey, European Language Resources Association (ELRA) (may 2012)