Proceedings of SIGUL2022 @LREC2022, pages 112–121
Marseille, 24-25 June 2022

CoSwID, a Code Switching Identification Method Suitable for Under-Resourced Languages

Laurent Kevers
UMR CNRS 6240 LISA, Università di Corsica - Pasquale Paoli
Avenue Jean Nicoli, 20250 Corte, France
kevers_l@univ-corse.fr

Abstract

We propose a method for identifying monolingual textual segments in multilingual documents. It requires only a minimal number of linguistic resources – word lists and monolingual corpora – and can therefore be adapted to many under-resourced languages. Taking these languages into account when processing multilingual documents in NLP tools is important as it can contribute to the creation of essential textual resources. This language identification task – code switching detection being its most complex form – can also provide added value to various existing data or tools. Our research demonstrates that a language identification module performing well on short texts can be used to efficiently analyse a document through a sliding window. The results obtained for code switching identification – between 87.29% and 97.97% accuracy – are state-of-the-art, which is confirmed by the benchmarks performed on the few available systems that have been used on our test data.

Keywords: language identification, code switching, under-resourced languages, Corsican

1. Introduction

Identifying the language of a document is a task that generally gives very good results. However, there are still various situations where performance tends to lower. Hughes et al. (2006) and Jauhiainen et al. (2018) note, more than ten years apart, that the processing of multilingual documents and the support of under-resourced languages are among the aspects that are not yet fully mastered[1]. In this paper, we focus on these two specific issues by studying the possibility of segmenting a multilingual document – potentially including under-resourced languages – into monolingual sequences whose languages would be identified.

The presence of several languages in a document may have multiple reasons and take different forms. It may be intentional, as in the case of a document which provides the same content in different languages, or even organises language alternation throughout the text[2]. It can also be unintentional – unpremeditated – as in the case of texts transcribing oral interviews where code-switching situations occur.

The localisation of monolingual segments and the identification of languages within a multilingual document is important in many ways, especially for under-resourced languages. Building corpora for these languages is a major challenge that can benefit from fine-grained language identification in order to produce the cleanest possible linguistic resources. This objective can be achieved either by excluding multilingual documents from the corpus or by splitting or annotating sections according to their language. Further linguistic processing of these documents can then take advantage from the language information linked to each word. For example, morphosyntactic annotation – whether done (semi-)manually to create reference data, or automatically using a tagger – can be facilitated and enhanced by this information. At a higher applicative level, search engine indexes or machine translation can also be improved if monolingual segments are correctly located and identified in the documents.

By definition, under-resourced languages often lack the resources and tools to identify them, especially in a multilingual context. We believe that this difficulty must be addressed. Most languages are, as a matter of fact, under-resourced (Joshi et al., 2020), even though they represent a significant number of speakers and constitute a cultural richness whose survival should be encouraged by the development and implementation of appropriate resources and tools.

This paper will first describe the context and objectives of our work (section 2), followed by an overview of the state of the art (section 3). In light of these elements, a solution is then presented (section 4) and evaluated (section 5). Finally, we will discuss the results and outline a method for code switching identification in the context of under-resourced languages (section 6).

2. Context, scope and objectives

In this work on the identification of monolingual sequences within multilingual documents, we are interested both in documents where language diversity is structured and in documents where code switching occurs. In the first case, we can expect fairly homogeneous and well-defined areas, for example a text available in two languages and structured in two columns,

[1] Other issues such as open language detection, the effects of preprocessing, the support of a large number of languages, the distinction between close languages and dialects, as well as language identification for short texts are also identified as problematic.

[2] For example, a switch of language every paragraph if several official or used languages coexist.

or a text where language changes at each paragraph. In the second case, the language switches and the size of the linguistically homogeneous segments may vary greatly and be much more irregular. This is of course the most challenging and interesting problem and will be our main objective.

While many texts will involve only a limited number of languages – for example two or three, possibly known in advance – we also consider, more generally, the situation where it may be higher.

Our approach aims at making the identification of monolingual segments accessible to under-resourced languages. We therefore consider that there is not necessarily an annotated corpus available to describe the phenomenon and to carry out specific and massive machine learning. However, we assume a minimum of resources are available, i.e. an assumed monolingual raw corpus and a word list for each language to include.

Among the under-resourced languages, we are particularly interested in Corsican, for which we have started to develop resources and tools in recent years (Kevers and Retali-Medori, 2020). We have been able to obtain interesting results for the identification of this language at the scale of a whole document (Kevers, 2021), but the treatment of multilingual documents remains unsatisfactory. The presence of words or parts in a language other than the one globally detected is not currently handled, which can be a drawback in some cases. Our concerns for Corsican include the manual or semi-automatic constitution of raw or annotated corpora – in particular morphosyntactically – which can greatly benefit from a language-aware processing. Moreover, we are working on the *Banque de Données Langue Corse* project (BDLC), which is developing a database featuring ethnotexts in Corsican[3]. These texts include segments in French and code switching identification would therefore bring a real added value.

3. State of the art

Language identification has been the focus of much research. While the problem can to some extent be considered solved, there are various situations where this is not the case. In his survey, Jauhiainen et al. (2018) provide an overview of the field and identify open questions. Based on their work – enriched with a few additional references – we take it up again from the angle of the analysis of multilingual documents, as this remained relatively marginal in his original work. Our presentation is necessarily more synthetic, we therefore refer to the original paper for further details.

Some research, such as Prager (2000) or Lui et al. (2014), have addressed the analysis of multilingual documents with the aim of identifying – or even quantifying – the languages they contain, but without locating them precisely. This approach, although it may be useful, is not sufficient for our purposes.

Regarding the identification and characterisation of monolingual segments within a document, some systems introduce a limitation on the number of languages involved. For example, Mandl et al. (2006) or Singh and Gorla (2007) postulate the presence of two or three languages maximum among those recognised by the system. The limitation can also concern one or more specific language pairs, such as English-Spanish (Lignos and Mitch, 2013), Turkish-Dutch (Nguyen and Doğruöz, 2013), English-Spanish and English-Dutch (Chang and Lin, 2014), Hindi-English (Jhamtani et al., 2014), Irish-English, Welsh-English and Breton-English (Minocha and Tyers, 2014), English-Spanish and MSA[4]-Egyptian (Samih et al., 2016), or English-Dutch (Dongen, 2017).

The granularity of the identified units is also variable. While the sentence level is sometimes chosen (Stensby et al., 2010; Lavergne et al., 2014), most methods operate on words or tokens. Some approaches, however, extend the analysis to the character level (Pethő and Mózes, 2014; Kocmi and Bojar, 2017).

Although many approaches could meet our needs (Hammarström, 2007; Rehurek and Kolkus, 2009; Ullman, 2014; Giwa and Davel, 2014; Lavergne et al., 2014; King et al., 2015), we have to note that the availability of tools, in the form of source code or reusable modules, is very limited. Some functional solutions, discussed below, have nevertheless been identified and present the advantage of being comparable and benchmarkable against common data.

With *SegLang*, Yamaguchi and Tanaka-Ishii (2012) propose the division of multilingual documents into monolingual segments for 222 languages, including Corsican. The system, which uses either training data from the *Universal Declaration of Human Rights* or from *Wikipedia*, can also be re-trained with new data. There is no pre-built model as the loading and processing of reference data is done at initialization. The volume of the training set and the data to be processed might be limited according to the available memory[5]. The principle of the analysis is to optimise the segmentation of the text by minimising the *Description Length*. The source code[6] is not ready to use as is, but the authors provide us with a working demo version[7].

[3]The Corsican Language Database project, `https://bdlc.univ-corse.fr`, collects linguistic data on know-how and cultural traditions throughout Corsica by means of oral interviews with native speakers. The lexical surveys are sometimes extended by semi-directed interviews which allow the collection of authentic accounts in Corsican, which once transcribed are integrated into the database as *ethnotexts*.

[4]Modern Standard Arabic.

[5]We did not perform extensive tests to define this limitation, but we were not able to use our full learning data set, even by allocating up to 6 GB of memory to Java. We ended up using about 500 KB of data per language.

[6]`https://github.com/hiroshi-cl/seglang-core`, under BSD license.

[7]We would like to thank them for their help!

King and Abney (2013) present *LangId*, a word-level language identification system designed for the processing of bilingual documents[8]. There are 30 languages initially proposed, Corsican is not among them. It is however possible to generate a model from new data. Again, the size of the data to be processed – for training and to be analysed – depends on the memory that can be allocated. Several classification methods are available, CRF being the one offering the best reported results. The Java code is made available and can be used without any particular dependency[9].

The third available tool is *Codeswitchador* (Lignos and Mitch, 2013), which allows code switching detection by identifying the language at word level. With this system, it is not possible to consider more than two languages, those initially defined being Spanish and English. However, it is possible to generate language models, based on word appearance probabilities. This procedure allows adaptation to other languages. Context-sensitive heuristics are also implemented. The Python code is available[10] and needs *numpy*.

A last system, *LanideNN*[11] (Kocmi and Bojar, 2017), based on neural networks and supporting 131 languages – including Corsican – could have completed this trio. Unfortunately, the installation of the required dependencies[12] could not be achieved. A migration to a more recent framework seems to be envisaged, which could make *LanideNN* usable in the future.

Given these previous works, we highlight the small number of open and reusable tools, as well as some of their limitations. We therefore propose a solution for the identification of monolingual segments within multilingual documents, easily adaptable to different use cases, in particular the simultaneous presence of many languages, possibly under-resourced, as well as code switching situations.

4. A solution to code switching identification : CoSwID

The approach explored is in line with our previous work on **language identification** (Kevers and Retali-Medori, 2020; Kevers, 2021). The experiments carried out have been encouraging and have allowed us to highlight a tool – *ldig*[13] (Nakatani, 2012) – which, in addition to good results at the document level, has shown its ability to efficiently process small texts. We assume that this type of tool can be used to analyse a document through small sequences using a sliding window, and thus determine locally the language of each token.

Unlike most other approaches that extract n-grams of different lengths – for example, ranging from two to four – the analysis performed by *ldig* is based on the concepts of *infinity gram* and *maximal substring* (Okanohara and Tsujii, 2009). The principle consists of extracting a set of character substrings with a priori undefined and variable length. Given their potentially large number, these are aggregated and represented by maximal substrings[14].

The decision to use *ldig* does not imply any exclusive dependency between this component and our developments. Any other language identification module, with similar characteristics, could be used instead.

For this work, we chose to train the language identification module for Corsican as well as for eight other European languages[15]. The limitation to only nine languages was motivated by the need to gather minimal resources (monolingual corpora, dictionaries), as well as by the assumption that very highly multilingual documents are relatively rare. With Corsican in mind, we took care to select, French and Italian, which are languages that are close historically, linguistically and in real-life situations. The other languages have been added in order to be able to handle a slightly larger number of them. Even if the integration of many languages is not a priority, it remains interesting in the perspective of having a more generic approach.

In addition to the language identification module, a set of **monolingual dictionaries**[16] have been collected. Their utility in the analysis process is not fundamental, but they constitute an interesting tool when there is an indecisive decision on some words.

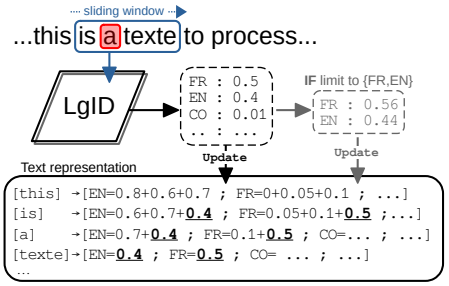

Figure 1: Three tokens wide sliding window parsing

The **parsing** general principle is to split the text into tokens and to analyse them progressively by means of a sliding window which is moved forward from token to token until the end of the text (Figure 1). Besides

[8] The extension to documents containing a larger number of languages is however possible.

[9] `http://www-personal.umich.edu/~benking/resources/langid_release.tar.gz`. Java version 8 is required. There is no user license specified.

[10] `https://github.com/ConstantineLignos/Codeswitchador`, under BSD license.

[11] `https://github.com/kocmitom/LanideNN`

[12] A Python 3.4 version, which is no longer available in recent Linux distributions, and a rather old version of TensorFlow (0.8).

[13] `https://github.com/shuyo/ldig`

[14] For example 'abracadabra' gives the maximal substrings 'a', 'abra' and 'abracadabra' (Nakatani, 2012).

[15] English, French, German, Italian, Dutch, Portuguese, Romanian and Spanish.

[16] These are more precisely word lists, grammatical or inflectional information not being used.

the current token, the window contains a number of additional units taken from the left and right context, and has therefore always an odd size, with a minimum length of one. The resulting snippet is sent to the language identification module, which returns the set of probabilities for all languages known by the system. If we wish to limit the analysis to a subset of languages, irrelevant ones are eliminated from the result – regardless of their importance in this one – and the remaining probabilities are readjusted to keep the sum to 100%. The results obtained for the different languages on a snippet are recorded for each token by adding the probabilities to any scores already produced by a previous segment[17]. Once this has been done for the whole text, each token has a score for each language of the system. In principle, the language with the highest probability is assigned to the token. However, this analysis can be questioned if the difference between the first and subsequent languages does not seem to be sufficient. This is judged according to a configurable margin[18] (*indecision gap*). In this case, a threshold is applied using the same margin[19]. Once this initial selection has been made, various approaches are possible to choose between the candidate languages: consultation of monolingual dictionaries, use of the language identification module on the token alone, or a combination of these two methods. In all cases, if these additional investigations are not successful, the initial scores are maintained and the language that ranks first is selected.
The CoSwID Python code is available[20], as well as an updated version of *ldig*[21].

5. Tests and evaluation

5.1. Data

First of all, regarding the training data needed to create a language identification model with *ldig*, we mainly used, for the eight European languages, sentence corpora from the *Tatoeba* collaborative platform[22]. For Corsican, which is only marginally represented in this source, we used three corpora made available by the BDLC[23]: *Wikipedia*, the *Bible* and *A Piazzetta*, a local news blog in Corsican. Table 1 (*Base* column) gives an overview of the available data.

Language		*Base*	*Filter*	*Filter2*
eng	en	67 948 293	64 653 131	58 824 526
ita	it	32 022 121	22 815 185	20 813 785
deu	de	29 987 665	29 106 064	28 126 760
fra	fr	22 482 372	19 062 318	17 833 313
por	pt	17 399 633	13 688 730	13 054 128
spa	es	15 437 547	12 294 945	11 069 048
cos	co	11 868 620	10 483 557	10 402 975
nld	nl	5 968 644	5 455 944	5 034 300
ron	ro	1 045 723	862 135	782 957

Table 1: Number of characters in the training data (base corpus or after one or two filtering processes)

All these training documents were slightly preprocessed: normalisation to lower case, punctuation removal and space normalisation. As Corsican documents from the *Wikipedia* and *A Piazzetta* corpora may sometimes contain substantial passages in other languages, they were filtered out by means of keywords language detection[24]. Finally, the training data was presented in lines of maximum about 200 characters[25].
The monolingual dictionaries are mainly derived from the lexical resources made available by *Unitex*[26], with the exception of those for Dutch (*OpenTaal*[27]), Romanian (*ELRC*[28]) and Corsican (*BDLC*). The number of items in each dictionary is shown in Table 2.

Language			*Items*
English	eng	en	398 417
Italian	ita	it	95 038
German	deu	de	8 277
French	fra	fr	794 286
Portuguese	por	pt	890 193
Spanish	spa	es	477 976
Corsican	cos	co	43 051
Dutch	nld	nl	401 575
Romanian	ron	ro	19 946

Table 2: Dictionaries data

Finally, we have created several evaluation corpora. The first is a set of synthetic multilingual documents based on the *Universal Declaration of Human Rights* (UDHR)[29] and involving all nine languages. Sev-

[17] If the window size is three, each token will receive three scores which will be summed language by language before being normalised.

[18] With a margin of 10% and the probabilities [Lg1=0.25, Lg2=0.22, Lg3=0.18, Lg4=0.10, Lg5=0.10, Lg6=0.10, Lg7=0.05], the differences between Lg1 and Lg2 (3%) and Lg1 and Lg3 (7%), are not sufficient to choose Lg1 directly.

[19] In the previous example, all languages with a probability greater than or equal to 15% are kept (Lg1, Lg2 and Lg3), the others are eliminated.

[20] `https://github.com/lkevers/coswid`

[21] `https://github.com/lkevers/ldig-python3`

[22] `https://tatoeba.org`, under CC BY 2.0 FR license.

[23] `https://bdlc.univ-corse.fr/tal/index.php?page=res`

[24] Based on a document-wide count of language-specific keyword occurrences. Keyword lists are from Lucene (`https://github.com/apache/lucene`), except for the Corsican one. The number of keywords per language varies between 78 and 393. Documents not detected as being mostly in Corsican were discarded (128 documents for *A Piazzetta* and 141 documents for *Wikipedia*).

[25] For performance reasons, it is recommended not to provide *ldig* with too long learning documents.

[26] `https://unitexgramlab.org/language-resources`, under LGPLLR license.

[27] `https://github.com/OpenTaal/opentaal-wordlist`, under revised BSD or CC BY 3.0 licenses.

[28] The "Romanian–English parallel wordlists" available at `https://elrc-share.eu`, under CC-BY 4.0 license.

[29] We used the "udhr2" corpus available in NLTK:

eral versions were produced, depending on the frequency and granularity of the composition between languages. The *UDHR-parag* corpus alternates at paragraph breaks, whereas for *UDHR-sent*, this occurs at the end of each sentence. Finally, the *UDHR-word* corpus provides switching within the sentence itself. The construction of the latter is carried out by replacing, after every three to seven tokens, within a sentence in a "main" language chosen randomly, segments of one to four tokens in another language, chosen randomly again. Even if the mixing has been performed as accurately as possible, a word-for-word replacement respecting the syntactic structure of the sentences is not possible[30]. For these three corpora, all the data are used, so the same text is found nine times in different languages. Finally, it should be noted that the original order of the sentences has not been kept.

The second evaluation corpus (*BDLC-ethno*) is made of authentic ethnotexts containing transcripts of oral interviews conducted in Corsican, in which passages in French may occur.

Beyond their artificial or authentic character, the particularities of these different corpora allow us to take into account different criteria: the frequency of alternation, the length of the segments, as well as the number of languages involved.

The evaluation data, as well as the data used to generate the language identification model, is available[31].

5.2. Experiments

The metric we are trying to maximise is the accuracy of language identification for each token (*overall accuracy*, noted Acc_o). The accuracy obtained specifically on the alternation zones (*targeted accuracy*, noted Acc_t) is also a secondary point of interest.

The experiments were grouped into three tests with different characteristics regarding the frequency of switching, the length of the segments and the number of languages involved.

TEST-1 concerns the *UDHR-parag* and *UDHR-sent* corpora. They consist of relatively long monolingual sequences, and switching is therefore infrequent[32], but involve all nine languages. This test was performed on synthetic data. The language alternation was created artificially from parallel documents, without altering the sentences.

TEST-2 uses the *UDHR-word* corpus, which is marked by short monolingual sequences, frequent switching[33], and again all nine languages. It is a synthetic data set, whose creation process may have led to an alteration of the sentence structure.

Finally, **TEST-3** focuses on the *BDLC-ethno* corpus, composed of authentic data. It is characterised by long sequences in Corsican with insertions of variable size, but rather short, in French[34]. The frequency of switches is slightly higher than in TEST-1, but much lower than in TEST-2.

In order to identify the best configuration for each test, we experimented with several values for the different parameters. First, the **training of the language identification model** was performed either on all data (*Base*) or on a subset (see Table 1). For *Filter*, we applied the keyword filtering method already used for Corsican (see section 5.1 and footnote 24) to all corpora whose documents were transformed into fragments of 200 characters. A second selection (*Filter2*), carried out in accordance with the same approach, but using the language detector CLD3[35], was produced from *Filter*. The **size of the sliding window** used for the analysis was set to 1, 3, 5 or 7 tokens. The **indecision gap** was set to values of 0, 0.05, 0.1 and 0.2. Finally, there are three **verification methods**[36] in case of a questionable language identification: using the dictionary (*dico*), language identification on the single token concerned only (*lgID*), and the combination of these two methods[37] (*full*). Finally, for the *BDLC-ethno* corpus, a **limitation to the two languages** actually present in the corpus can be requested. All the combinations of these parameters – 120 in total[38] – were tested and measured in order to identify the best performing ones.

Finally, our results were compared to those obtained by the three systems that we were able to test: *SegLang*, *LangId* and *Codeswitchador* (see section 3). *SegLang* has built-in data from *Wikipedia* or the *Universal Declaration of Human Rights*. Since TEST-1 and -2 also involved UDHR data, we only used *SegLang* with *Wikipedia* data (SLW) or by using a subset of our own data[39] (SLC). *LangId* (LID) and *Codeswitchador* (CS) also benefited from the same data and were used for TEST-3 in bilingual mode.

`https://www.nltk.org/nltk_data/` (public domain).

[30] It could be a drawback if the code switching detection method plans to use this information, which is not our case.

[31] `https://github.com/lkevers/coswid`

[32] *UDHR-parag* contains 16,095 tokens spread over 531 paragraphs, a switch of language occurring at each paragraph break. *UDHR-sent* contains 16,097 tokens divided into 621 sentences, the language being changed for each new sentence.

[33] *UDHR-word* contains 18,417 tokens and 2,598 language switching sequences (tokens in language X inserted in a sentence globally in language Y).

[34] Out of 79,421 tokens, 74,569 (93.89%) are in Corsican, 4,042 (5.09%) in French – spread over 959 segments – and 810 (1.02%) without an attributed language (abbreviations, proper nouns or speech turn markers when identified).

[35] `https://github.com/google/cld3`, (Apache-2.0).

[36] When the indecision gap is set to 0, the language with the highest probability is systematically selected and the verification method parameter is not used.

[37] When the two methods have different results, they are compared to the language detected globally for the whole document. If no options emerge, the initial result is retained.

[38] (3 filtering modes of learning data * 4 window sizes * 3 indecision gaps (different of 0) * 3 verification methods)=108 + (3 filtering modes of learning data * 4 window sizes * 1 indecision gap (equals to 0 : no verification requiered))=12.

[39] For memory usage reasons, the training corpus was limited to 500KB per language.

5.3. Results

For **TEST-1**, the ten best configurations are detailed for the paragraph (Table 3) or sentence (Table 4) switching level. With a few exceptions, the results tend to converge. The settings that emerge involve exclusively the datasets filtered twice or, to a lesser extent, once. The size of the window is rather large: in general seven tokens, five in a few cases. The values to adopt for these two parameters seem therefore quite clear, which is less the case for the indecision gap. However, the trend points towards values of 10% to 20%. The verification methods based on dictionaries – *dico* and, in a less important way, *full* – emerge as the most effective. The best performing configuration – which is identical in both cases – offers 98.15% of overall accuracy for the paragraph level switching corpus and 98.00% for the one containing sentence level switchings. The accuracy targeted at the language changing points was measured at 88.75% and 89.37% respectively.

SegLang obtained a lower performance than this optimal configuration with the embedded *Wikipedia* data, but outperforms it by 1.39% (paragraph) to 1.61% (sentence) with our filtered training data (*Filter2*).

#	*Configuration*				Acc_o	Acc_t
1	**Filter2**	**7**	**0.2**	**dico**	**0.9815**	**0.8875**
2	Filter	7	0.2	dico	0.9796	0.8865
3	Filter2	7	0.1	dico	0.9785	0.8625
4	Filter2	5	0.2	dico	0.9766	0.8941
5	Filter2	7	0.05	full	0.9762	0.8503
6	Filter2	7	0.05	dico	0.9762	0.8451
7	Filter	7	0.1	dico	0.9751	0.8536
8	Filter2	7	0.1	full	0.9747	0.8658
9	Filter2	5	0.1	dico	0.9740	0.8734
10	Filter2	7	0.05	lgID	0.9739	0.8362
SLW	Built-in Wikipedia data				0.9211	0.7768
SLC	**Custom data 500KB/language**				**0.9954**	**0.9774**

Table 3: Top 10 for TEST-1 (paragraph)

#	*Configuration*				Acc_o	Acc_t
1	**Filter2**	**7**	**0.2**	**dico**	**0.9800**	**0.8937**
2	Filter2	7	0.1	dico	0.9758	0.8663
3	Filter	7	0.2	dico	0.9749	0.8728
4	Filter2	5	0.2	dico	0.9737	0.8933
5	Filter2	7	0.05	dico	0.9715	0.8378
6	Filter2	7	0.1	full	0.9714	0.8619
7	Filter2	7	0.05	full	0.9712	0.8414
8	Filter	7	0.1	dico	0.9708	0.8462
9	Filter2	5	0.1	dico	0.9704	0.8704
10	Filter	5	0.2	dico	0.9695	0.8732
SLW	Built-in Wikipedia data				0.9053	0.7576
SLC	**Custom data 500KB/language**				**0.9961**	**0.9815**

Table 4: Top 10 for TEST-1 (sentence)

The performance[40] observed when varying the parameters one by one with respect to the optimal configuration is shown in Table 5. The use of filtered language identification training data has a positive impact on the overall accuracy (+ 1.35% between *Base* and *Filter2*) as well as on the the targeted accuracy (+1.85%). The size of the window is a decisive criterion, as the results become weaker as the window narrows. The results obtained for windows of seven and five tokens remain fairly close to each other. The setting using only one token has a much lower performance. The use of an indecision gap improves the accuracy, especially for the switching zones. The *dico* verification method is clearly more efficient in terms of overall accuracy (+1.55% with regard to *full* and +2.35% compared with *lgID*) and targeted accuracy (respectively +1.93% and +4.59%).

#	*Configuration*				Acc_o	Acc_t
1	**Filter2**	**7**	**0.2**	**dico**	**0.9800**	**0.8937**
3	**Filter**	7	0.2	dico	0.9749	0.8728
16	**Base**	7	0.2	dico	0.9665	0.8752
4	Filter2	**5**	0.2	dico	0.9737	0.8933
47	Filter2	**3**	0.2	dico	0.9425	0.8776
91	Filter2	**1**	0.2	dico	0.5937	0.6582
2	Filter2	7	**0.1**	dico	0.9758	0.8663
5	Filter2	7	**0.05**	dico	0.9715	0.8378
19	Filter2	7	**0**	n-a	0.9655	0.8019
20	Filter2	7	0.2	**full**	0.9645	0.8744
36	Filter2	7	0.2	**lgID**	0.9565	0.8478

Table 5: Variation of parameters with respect to the optimal solution (TEST-1 sentence)

For **TEST-2** (Table 6), the best performing configurations use again the language identification model trained on the filtered data (*Filter2* and *Filter*), but this time with smaller sliding window sizes (three to five tokens). The trend for the other parameters remains unchanged from TEST-1: a 10% to 20% indecision gap between candidate languages and the dictionary-based verification method. TEST-2, which features more language changes and shorter segments, can be considered as the most complex case we have to analyse. It is therefore not surprising that we find a decrease in both overall (87.29%) and targeted accuracy (82.54%).

SegLang is positioned in the same way as for TEST-1: lower with the *Wikipedia* data, and 0.78% higher with our filtered training data.

The results obtained after varying the parameters with respect to the optimal solution are shown in Table 7. The improvements brought by the filtering operations on the training data are significant for the global and targeted accuracies. They are mainly observable between *Base* and *Filter* (about +4%), while the move to *Filter2* brings an improvement that remains below 1%. The global accuracy obtained with a window of five tokens is very close to the optimal solution (three tokens), but offers a lower accuracy in the switching zones. A window reduced to a single token gives very poor performance. The results observed when the indecision gap varies confirm the orientation towards values be-

[40] These results are related to the sentence level, the paragraph level being comparable.

#	*Configuration*				Acc_o	Acc_t
1	**Filter2**	**3**	**0.2**	**dico**	**0.8729**	**0.8254**
2	Filter2	5	0.2	dico	0.8701	0.8106
3	Filter	3	0.2	dico	0.8672	0.8224
4	Filter	5	0.2	dico	0.8633	0.8017
5	Filter2	3	0.1	dico	0.8611	0.8071
6	Filter2	5	0.1	dico	0.8568	0.7894
7	Filter	3	0.1	dico	0.8563	0.8048
8	Filter2	3	0.05	dico	0.8527	0.7953
9	Filter	5	0.1	dico	0.8501	0.7810
10	Filter2	5	0.1	full	0.8464	0.7817
SLW	Built-in Wikipedia data				0.7061	0.6306
SLC	**Custom data 500KB/language**				**0.8807**	**0.8167**

Table 6: Top 10 for TEST-2

#	*Configuration*				Acc_o	Acc_t
1	**Base**	**7**	**0.2**	**dico**	**0.9631**	**0.6727**
2	Base	7	0.1	dico	0.9620	0.6554
3	Base	7	0.05	dico	0.9612	0.6469
4	Base	7	0.05	full	0.9605	0.6368
5	Base	7	0	n-a	0.9600	0.6370
6	Filter	7	0.2	dico	0.9597	0.7383
7	Base	7	0.1	full	0.9597	0.6340
8	Base	7	0.05	lgID	0.9596	0.6344
9	Filter2	7	0.2	dico	0.9590	0.7602
10	Filter	7	0.1	dico	0.9583	0.7237
SLW	Built-in Wikipedia data				0.9460	0.5648
SLC	**Custom data 500KB/language**				**0.9754**	**0.7120**

Table 8: Top 10 for TEST-3 (9 languages)

tween 10% and 20%. Finally, it is again the use of dictionaries that emerges as the most appropriate solution to verify the attribution of a language when the probabilities are not strong enough (about +6% compared to *full* and +9% with regard to *lgID*).

#	*Configuration*				Acc_o	Acc_t
1	**Filter2**	**3**	**0.2**	**dico**	**0.8729**	**0.8254**
3	**Filter**	3	0.2	dico	0.8672	0.8224
35	**Base**	3	0.2	dico	0.8282	0.7820
11	Filter2	**7**	0.2	dico	0.8462	0.7701
2	Filter2	**5**	0.2	dico	0.8701	0.8106
91	Filter2	**1**	0.2	dico	0.5892	0.5869
5	Filter2	3	**0.1**	dico	0.8611	0.8071
8	Filter2	3	**0.05**	dico	0.8527	0.7953
19	Filter2	3	**0**	n-a	0.8393	0.7774
56	Filter2	3	0.2	**full**	0.8096	0.7615
75	Filter2	3	0.2	**lgID**	0.7820	0.7300

Table 7: Variation of parameters with respect to the optimal solution (TEST-2)

For **TEST-3** (Table 8), using Corsican ethnotexts, the best configuration involves this time the language identification model trained on *Base*, wich is represented seven times in the Top 10. Filtered models also appear marginally with sligthly lower overall accuracy values. This could be explained by the fact that *Base* had benefited from a first filtering on Corsican, which is the majority language in the *BDLC-ethno* corpus. Interestingly, filtered models consistently achieve higher targeted accuracy results. The optimal sliding window size is unanimously found to be seven tokens, as in TEST-1, while the indecision gap values are scattered between 20% – as for the best setting – and 0%. The verification method is again rather dictionary-based, even if different solutions are represented. In short, only the size of the window seems to have a really clear and decisive influence. The optimal parameters allow to reach the overall accuracy of 96.31%, and a targeted accuracy of 67.27%.

SegLang is again a step behind with the embedded *Wikipedia* data, but slightly better (+1.23%) using our filtered data.

For this third test, investigation of changes in the detected optimal parameters (Table 9) highlights again the slightly lower overall accuracy, but higher targeted accuracy, for the language identification models based on filtered data. Similarly to the previous tests, the size of the window has an important impact on the results, which become weaker as the number of tokens decreases. However, the difference between a window of five or seven tokens is relatively small. In spite of the decrease in overall accuracy, a slightly higher targeted accuracy (just over 1%) can be achieved for configurations using smaller windows of five or three tokens. The solution using one token at a time is again lower. As far as the indecision gap is concerned, the results grow relatively slowly but steadily as its size increases. As already pointed out, the modification of the verification method shows that the approach using dictionaries is the most appropriate.

#	*Configuration*				Acc_o	Acc_t
1	**Base**	**7**	**0.2**	**dico**	**0.9631**	**0.6727**
9	**Filter2**	7	0.2	dico	0.9590	0.7602
6	**Filter**	7	0.2	dico	0.9597	0.7383
17	Base	**5**	0.2	dico	0.9554	0.7039
57	Base	**3**	0.2	dico	0.9244	0.7137
91	Base	**1**	0.2	dico	0.6833	0.4212
2	Base	7	**0.1**	dico	0.9620	0.6554
3	Base	7	**0.05**	dico	0.9612	0.6469
5	Base	7	**0**	n-a	0.9600	0.6370
20	Base	7	0.2	**full**	0.9548	0.6253
29	Base	7	0.2	**lgID**	0.9511	0.6179

Table 9: Variation of parameters with respect to the optimal solution (TEST-3, 9 languages)

As the *BDLC-ethno* corpus contains only Corsican and French, we observed the effect of choosing only between these two languages (**TEST-3bis**, Table 10). The global accuracy values obtained for the different configurations are very close to each other, the first ten varying from less than 0.2%, with an optimal configuration at 97.97%. The targeted accuracy results are slightly more variable, with the optimal configuration offering 78.39%. Note that this time there are only so-

lutions based on filtered data (*Filter2* or *Filter*). The window size remains high, between five and seven tokens, while the indecision gap holds around 20%, and the verification methods based on the use of dictionaries (*dico* or *full*) still seem to be the best.
For this last test, no third-party system scored better than ours. *Seglang*, used with our filtered data is the closest with a fairly small difference of 0.28%.

#	*Configuration*				Acc_o	Acc_t
1	**Filter**	**5**	**0.2**	**dico**	**0.9797**	**0.7839**
2	Filter2	7	0.2	dico	0.9788	0.7727
3	Filter	5	0.2	full	0.9786	0.7704
4	Filter	5	0.1	dico	0.9785	0.7706
5	Filter	7	0.2	dico	0.9783	0.7465
6	Filter	5	0.1	full	0.9781	0.7653
7	Filter	5	0.05	dico	0.9779	0.7637
8	Filter	7	0.2	full	0.9778	0.7393
9	Filter2	7	0.2	full	0.9778	0.7668
10	Filter	5	0.05	full	0.9777	0.7618
SLC	Custom data 500KB/language				0.9770	0.7151
LID	Custom data 500KB/language				0.9738	0.6664
CS	Custom data 500KB/language				0.9670	0.7576

Table 10: Top 10 for TEST-3bis, limited to two languages (cos-fra)

For the study of parameter variation (Table 11), we observe mainly small differences. However, we can highlight the usual effect of window size, lengths of seven and five being very close to each other.

#	*Configuration*				Acc_o	Acc_t
1	**Filter**	**5**	**0.2**	**dico**	**0.9797**	**0.7839**
11	**Filter2**	5	0.2	dico	0.9776	0.8066
21	**Base**	5	0.2	dico	0.9768	0.7092
5	Filter	**7**	0.2	dico	0.9783	0.7465
41	Filter	**3**	0.2	dico	0.9742	0.8262
94	Filter	**1**	0.2	dico	0.9021	0.7980
4	Filter	5	**0.1**	dico	0.9785	0.7706
7	Filter	5	**0.05**	dico	0.9779	0.7637
17	Filter	5	**0**	n-a	0.9770	0.7551
3	Filter	5	0.2	**full**	0.9786	0.7704
18	Filter	5	0.2	**lgID**	0.9770	0.7635

Table 11: Variation of parameters with respect to the optimal solution (TEST-3bis, 2 languages: cos-fra)

6. Discussion and conclusion

In the light of our results, no single setting can be identified. However, some trends can be observed and several parameters, such as the size of the sliding window and the magnitude of the indecision gap, can vary depending on the context of use.
The use of fragments of five to seven tokens has generally brought good results. In the case of very frequent and rather short alternations, a shortened window – between three and five tokens – has nevertheless proved to be more effective. In general, it is reasonable to assume that a shorter window is suitable for the analysis of texts in which language alternations occur quite dynamically and/or for short segments. A longer window will maximise accuracy for texts containing mainly long monolingual segments.
Beyond a constant value for optimal configurations (20%), it is not obvious to propose recommendations concerning the choice of the indecision gap. Since it is strongly linked to the verification method, the analysis is tricky and improvements – for example on the content of the dictionaries – could modify the configuration hierarchy. If a benefit is clearly obtained by this verification mechanism when several languages are involved, it is however much more marginal in the case of an alternation of only two languages.
About other parameters, it is worth noting that filtering the training data used for the language identification module is generally beneficial. More data does not always mean better results. Its suitability also matters. Finally, the most effective method for choosing between languages with close probabilities was the dictionary-based method.
We observe that the targeted accuracy sometimes improves by adopting slightly lower values for the window size, but this comes at the cost of a lower overall accuracy. It seems possible to consider a processing in multiple steps using different settings: a first approximation of the segments, followed by a closer review around the detected alternation points, and even a last check of the language attribution at the scale of a potentially homogeneous fragment, once it is defined.
Further work could include the improvement of the learning sets, as well as a closer investigation of language identification results or a wider and finer estimate of the parameters. The extension of the language number could also allow the testing of the reliability of the method, while providing a more universal module. Finally, the automatically performed annotations could be reviewed in some way, so that they can be used as training data for supervised approaches.
Let us conclude by highlighting that the achieved results – an overall accuracy between 87.29% and 97.97% – are in line with the state of the art. CoSwID is between 0.78% and 1.61% less accurate than *SegLang* when using our data, but it is slightly better for the third test featuring the two-language restriction. This leads us to confirm that a highly accurate and powerful language identification module can be used to detect language alternations such as code switching. We therfore consider that the use of CoSwID for language identification in the context of, among other things, morphosyntactic pre-annotation of Corsican corpora is possible. Generally speaking, the proposed approach allows to include without difficulty under-resourced languages with a minimum of resources. The release of the code and data will also contribute to address the small number of open and reusable systems for language identification in multilingual documents, and for the detection of code switching in particular.

7. Acknowledgements

This work was carried out thanks to CPER funding: *Un outil linguistique au service de la Corse et des Corses: la Banque de Données Langue Corse (BDLC)*.

8. Bibliographical References

Chang, J. C. and Lin, C.-C. (2014). Recurrent-Neural-Network for Language Detection on Twitter Code-Switching Corpus. *arXiv:1412.4314 [cs]*, December. arXiv: 1412.4314.

Dongen, N. (2017). Analysis and Prediction of Dutch-English Code-switching in Dutch Social Media Messages. Master's thesis, Universiteit van Amsterdam, Amsterdam, Netherlands.

Giwa, O. and Davel, M. (2014). Language identification of individual words with Joint Sequence Models. September.

Hammarström, H. (2007). A Fine-Grained Model for Language Identification. In *Proceedings of Improving Non English Web Searching (iNEWS-07) Workshop at SIGIR 2007*, pages 14–20, Amsterdam, Netherlands.

Hughes, B., Baldwin, T., Bird, S., Nicholson, J., and Mackinlay, A. (2006). Reconsidering language identification for written language resources. In *Proceedings of the fifth international conference on Language Resources and Evaluation (LREC2006)*, pages 485–488. ELRA.

Jauhiainen, T., Lui, M., Zampieri, M., Baldwin, T., and Lindén, K. (2018). Automatic Language Identification in Texts: A Survey. *arXiv:1804.08186 [cs]*, April. arXiv: 1804.08186.

Jhamtani, H., Bhogi, S. K., and Raychoudhury, V. (2014). Word-level Language Identification in Bi-lingual Code-switched Texts. In *Proceedings of the 28th Pacific Asia Conference on Language, Information and Computing*, pages 348–357, Phuket,Thailand, December. Department of Linguistics, Chulalongkorn University.

Joshi, P., Santy, S., Budhiraja, A., Bali, K., and Choudhury, M. (2020). The State and Fate of Linguistic Diversity and Inclusion in the NLP World. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online, July. ACL.

Kevers, L. and Retali-Medori, S. (2020). Towards a Corsican Basic Language Resource Kit. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2726–2735, Marseille, France, May. European Language Resources Association.

Kevers, L. (2021). L'identification de langue, un outil au service du corse et de l'évaluation des ressources linguistiques. *Traitement Automatique des Langues*, 62(3).

King, B. and Abney, S. (2013). Labeling the Languages of Words in Mixed-Language Documents using Weakly Supervised Methods. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1110–1119, Atlanta, Georgia, June. Association for Computational Linguistics.

King, L., Kübler, S., and Hooper, W. (2015). Word-level language identification in The Chymistry of Isaac Newton. *Digital Scholarship in the Humanities*, 30(4):532–540, December.

Kocmi, T. and Bojar, O. (2017). LanideNN: Multilingual Language Identification on Character Window. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 927–936, Valencia, Spain, April. Association for Computational Linguistics.

Lavergne, T., Adda, G., Adda-Decker, M., and Lamel, L. (2014). Automatic Language Identity Tagging on Word and Sentence-Level in Multilingual Text Sources: a Case-Study on Luxembourgish. In Khalid Choukri, et al., editors, *Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3300–3304, Reykjavik, Iceland, May. European Language Resources Association (ELRA).

Lignos, C. and Mitch, M. (2013). Toward web-scale analysis of codeswitching. In *87th Annual Meeting of the Linguistic Society of America*.

Lui, M., Lau, J. H., and Baldwin, T. (2014). Automatic Detection and Language Identification of Multilingual Documents. *Transactions of the Association for Computational Linguistics*, 2:27–40.

Mandl, T., Shramko, M., Tartakovski, O., and Womser-Hacker, C. (2006). Language Identification in Multi-lingual Web-Documents. pages 153–163, May.

Minocha, A. and Tyers, F. (2014). Subsegmental language detection in Celtic language text. In *Proceedings of the First Celtic Language Technology Workshop*, pages 76–80, Dublin, Ireland, August. Association for Computational Linguistics and Dublin City University.

Nakatani, S. (2012). Short Text Language Detection with Infinity-Gram, May. https://www.slideshare.net/shuyo/short-text-language-detection-with-infinitygram-1294944.

Nguyen, D. and Doğruöz, A. S. (2013). Word Level Language Identification in Online Multilingual Communication. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 857–862, Seattle, Washington, USA, October. Association for Computational Linguistics.

Okanohara, D. and Tsujii, J. (2009). Text Categorization with All Substring Features. In *Proceedings of SDM 2009*, pages 838–846, April.

Pethő, G. and Mózes, E. (2014). An n-gram-based language identification algorithm for variable-length

and variable-language texts. *Argumentum*, 10:56–82.

Prager, J. (2000). Linguini: Language Identification for Multilingual Documents. *J. of Management Information Systems*, 16:71–102, January.

Rehurek, R. and Kolkus, M. (2009). Language Identification on the Web: Extending the Dictionary Method. In *CICLing '09: Proceedings of the 10th International Conference on Computational Linguistics and Intelligent Text Processing*, pages 357–368, March.

Samih, Y., Maharjan, S., Attia, M., Kallmeyer, L., and Solorio, T. (2016). Multilingual Code-switching Identification via LSTM Recurrent Neural Networks. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 50–59, Austin, Texas, November. Association for Computational Linguistics.

Singh, A. K. and Gorla, J. (2007). Identification of Languages and Encodings in a Multilingual Document. In Fairon, Cédrick, et al., editors, *Building and Exploring Web Corpora (WAC3 - 2007). Proceedings of the 3rd web as corpus workshop, incorporating cleaneval.*

Stensby, A., Oommen, B., and Granmo, O.-C. (2010). Language Detection and Tracking in Multilingual Documents Using Weak Estimators. volume 6218, pages 600–609, August.

Ullman, E. (2014). Shibboleth - A Multilingual Language Identifier. Master's thesis, Uppsala University,, Uppsala.

Yamaguchi, H. and Tanaka-Ishii, K. (2012). Text Segmentation by Language Using Minimum Description Length. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 969–978, Jeju Island, Korea, July. Association for Computational Linguistics.