Subsegmental language detection in Celtic language text

Akshay Minocha
IIIT Hyderabad
Hyderabad (India)
akshay.minocha@students.iiit.ac.in

Francis M. Tyers
Giellatekno / CLEAR
UiT Norgga árktalaš universitehta
9017 Romsa (Norway)
francis.tyers@uit.no

Abstract

This paper describes an experiment to perform language identification on a sub-sentence basis. The typical case of language identification is to detect the language of documents or sentences. However, it may be the case that a single sentence or segment contains more than one language. This is especially the case in texts where *code switching* occurs.

1 Introduction

Determining the language of a piece of text is one of the first steps that must be taken before proceeding with further computational processing. This task has received a substantial amount of attention in recent years (Cavnar et al., 1994; Lui and Baldwin, 2012). However, previous research has on the whole assumed that a given text will be in a single language. When dealing with text from formal domains, this may be the case — although there are exceptions — such as quotations embedded in the text in another language. But when dealing with informal text, particularly in languages where the speech community is predominantly bi- or multi-lingual, this assumption may not hold.

The work presented in this paper was motivated by the problems in normalising non-standard input for the Celtic languages as a precursor to machine translation. When applying a normalisation strategy to a piece of text, it is necessary to first know the language of the piece of text you are applying it to.

There are several ways in which an informal text may contain parts in different languages. Some of these are presented in Table 1.

The remainder of the paper is laid out as follows. In section 3 we describe the problem in more detail and look at relevant prior work before proposing a novel method of sub-sentential language detection. Section 4 describes the evaluation methodology. Then in section 5 we present the results of our method and compare it against several other possible methods. Finally, section 6 presents future work and conclusions.

Code switching:	You're a [Meiriceánach, cén fáth] are you [foghlaim Gaeilge?!]
Quotations:	The anthem starts with the words ['Mae hen wlad fy nhadau…']
Named entities:	[Dr Jekyll] ha [Mr Hyde] embannet gant [Éditions Aber]
Interjections:	Hey, that's great, [diolch yn fawr!]
Translations:	Bloavezh mat d'an holl ! [Bonne anné à tous !]

Table 1: Some examples of text segments containing more than one language

2 Related Work

Code-switching and segment detection problems have been the subject of previous research. A good deal of work has been done on detecting code-switched segments in speech data (Chan et al., 2004; Lyu et al., 2006). It is seen that language modelling techniques have shown promise earlier, such as in Yu et

Pair	Language	Statistics (%)	
		Tokens	Segments
Irish—English	Irish	332	40
	English	379	42
Welsh—English	Welsh	419	64
	English	378	66
Breton—French	Breton	388	54
	French	379	53

Table 2: Document statistics of the annotated data used.

```
[en You're a] [ga Meiriceánach, cén fáth] [en are you] [ga foghlaim Gaeilge?!]
@afaltomkins [cy gorfod cael bach o tan] [en though init]
[en omg] [cy mar cwn bach yn] [en black and tan] [cy a popeth,] [en even cuter!!]
```

Figure 1: Example of text from a microblogging site chunked manually.

al. (2013), the experiment on Mandarin-Taiwanese sentences show a high accuracy in terms of detecting code-switched sentences.

In Chan et al. (2004) the authors have made use of the bi-phone probabilities and calculated them to measure a confidence metric, to (Lyu et al., 2006) which has made use of a classification method, named syllable-based duration classification, which uses the tonal syllable properties along with the speech signals to help predict the code switch points. In Yeong and Tan (2010) the authors use syllable structure information to identify words in code-switched text in Malay-English, however they did not recognise segments in running text, only identifying individual words.

3 Methodology

As the number of possible languages for each segment is in theory the set of all the world's written languages, we take a decision to simplify the task by only looking at texts in the Celtic language and the corresponding majority language spoken where the Celtic language is spoken. That is, we looked at detecting between Irish and English, Welsh and English and Breton and French.

3.1 Corpus

We hand-annotated a small evaluation set from a selection of posts to a popular microblogging site.[1] The *tweets* (microblog posts) were filtered into three sets which had been identified as Irish, Welsh and Breton using the *langid* tool (Lui and Baldwin, 2012). From these, we manually selected between 40 and 50 tweets for each language pair. Statistics on the number of segments and tokens is presented in Table 2. Certain tokens were escaped from the data, such as the 'mentioned' character (`@` symbol), subject tags 'hashtags' which are preceded by a `#` symbol, hyperlinks and the sequence `rt` which stands for 're-tweet'. An example of the content of our corpus after hand annotation is given in Figure 1. All of the tweets had at least one instance of code-switching.

3.2 Alphabet n-gram approach

We use the character n-gram approach along with some heuristics which are relevant to our problem domain of identifying segments for subsequent processing. We would like to both predict the code switched points but looking at the surrounding structure also decide the inclusion of them into the current or the next segment.[2]

We first built character language models using IRSTLM (Federico et al., 2008) for the five languages in question. For English and French a model was trained using the EuroParl corpus (Koehn, 2005). For Breton, Welsh and Irish we used corpora of text crawled from the web. To ensure no bias and also since our dataset for Breton was around 1.5 million, we sampled the same size of data for all the five languages. In order to build a character language model we replaced spaces with the underscore symbol '_', and then

[1] `http://indigenoustweets.blogspot.in/2013/12/mapping-celtic-twittersphere.html`

[2] The software used in this experiment is available online at: `removedforreview`.

placed a space character between each character. Punctuation and non-letter characters are also part of this language model. For example, the word 'sláinte!' would be broken down into a sequence of {'_ s', 's l', 'l á', 'á i', 'i n', 'n t', 't e', 'e !', '! _'}.

3.3 Sequence chunking

This section describes the heuristics taken into account while performing the chunking of the input data according to the sequences of segments in different languages. In Figure 2, 'chunks' represent the list of evaluated tuples of segments and their labelled language, 'buffer' is the expanding segment. *lang_predict* corressponds to any function which is used to determine the language of the token. The flag variable helps in implementing the heuristic of minimum segment size while labelling chunks.

```
buffer = [ ]  #undecided expanding window of chunk
chunks = [ ]  #decided labelled segment
buffer_language ← lang_predict(sentence[0])  #Language of first word
flag ← 0
for word in sentence do
    if lang_predict(word)==buffer_language then
        if flag= = 1 then
            buffer+ = [word_d,word]
            flag ← 0
        else
            buffer+ = [word]
    if lang_prediction(word)!=buffer_langugae then
        if flag= = 0 then
            flag ← 1
            word_d ← word
            continue
        else
            chunk+= [(buffer,buffer_language)]
            buffer = [word_d,word]
            buffer_language =← lang_predict(word)
            flag ← 0
if length(buffer)!=0 then
    chunk+= [[(buffer,buffer_language)]
```

Figure 2: Chunking Algorithm

3.4 Word-based prediction

This is a simple heuristic which was designed on the basis of the the most common words in the wordlists of the languages which are in question. After checking each word against both of the word lists,[3] it is associated to one language or another. In case of a conflict, for example, when the word exists in both wordlists, or in the case that it is unknown to both, the option of continuing with the previous span was taken and the previous selected tag was labelled, thus increasing the chunk.

3.5 Word-based prediction with character backoff

A better way to predict the spans of the sub-segments of the text is to include the two methods of word and character-based techniques as described above. In case of the word being present in only one of the two monolingual word lists the classification is simple, but in case of a conflict, a character bigram backoff was introduced to help us disambiguate the language label. This method works well because the earlier heuristic approach of just labelling the word with the label of the span which is expanding, would mean less code-switch detection and more shift towards the majority class.

4 Evaluation

For the Evaluation procedure, we follow the footsteps of the CoNLL-2000 shared task on language-independent named entity recognition: dividing text into syntactically related non-overlapping groups of words. This chunking mechanism (Tjong Kim Sang and De Meulder, 2003) is very similar to ours, in

[3]For the wordlists we used the *aspell* wordlists widely available on Unix systems.

System		Irish—English		Welsh—English		Breton—French	
		Irish	English	Welsh	English	Breton	French
`baseline`	p	2.50	0.0	0.0	0.0	0.0	0.0
	r	2.56	0.0	0.0	0.0	0.0	0.0
`langid-3character`	p	5.00	14.29	0.0	21.21	1.85	20.75
	r	5.41	8.45	0.0	14.58	1.92	12.36
`wordlist`	p	32.50	28.57	26.69	**40.91**	57.41	33.96
	r	23.64	26.09	**26.03**	**33.75**	47.69	33.33
`character bigram`	p	32.50	35.71	23.44	19.70	57.41	52.83
	r	22.41	26.79	15.31	16.67	41.33	37.84
`wordlist+character bigram`	p	**52.50**	**50.00**	**32.81**	31.82	**70.37**	**67.92**
	r	**38.18**	**43.75**	24.14	25.61	**57.58**	**57.14**

Table 3: Precision, p and recall, r for the systems by language.

System	Accuracy (%)		
	Irish—English	Welsh—English	Breton—French
`baseline`	42.76	42.16	44.07
`langid-3character`	57.24	45.92	43.16
`wordlist`	79.75	**74.28**	83.96
`character bigram`	81.29	65.62	76.79
`wordlist+character bigram`	**85.79**	72.40	**88.79**

Table 4: Accuracy of the systems over the three language pairs. The accuracy measures how often a token was assigned to the right language, independent of span.

terms of words which only belong to one category (here, language), and also evaluation based on the segment structure present in the data. The chunks here are such that they belong to only one language.

For our current task, each of the texts have been limited to two languages i.e. the primary language (the Celtic language) and the secondary language (the 'majority' language). Hence the type of chunking tags are limited to these. The evaluation statistics shown in Tables 3 and 4 mention two values for each of the experiment conducted on the three bilingual language datasets. The first, is the percentage of correctly detected phrases, which is the overall precision and the second is the number of phrases in the data that were found by the chunker, which is the overall recall.

Apart from the techniques discussed in section 3, some baselines are also used to give a comparative view of how well all the mechanisms perform.

4.1 Baseline

This is the most naïve method of classification, we used the language identification tool `langid.py` (Lui and Baldwin, 2012) on the whole dataset and labelled all the individual lines according to this single majority classification. As no chunking is performed we can expect that the precision and recall will be very low. However it does provide a reasonable baseline for the per-word accuracy.

4.2 `langid` character trigram prediction

For this system we used the character trigram probabilities to predict the detected language for each token. Trigrams were chosen after experimenting with 1–5 grams. The heuristics in section 3 were followed for the text processing and chunking part of the method.

5 Results

As described in Section 3 the data collected from Twitter for the three language pairs, was processed using the techniques mentioned. The statistics of the same are given in Table 2.

In terms of per-word accuracy, the baseline performs as expected, given that half of the words in the test set are from a given language. The `langid.py` based methods allow for segmenting the text into chunks, but favour the majority language.[4] While the precision and recall are low for the remaining models, we see that we are able to improve the performance by combining the wordlist based model with a character bigram model. And what is more, we are able to begin to not only identify particular words in a language, but also segments.

6 Conclusions

This paper has presented a very preliminary investigation into subsegment language identification in Celtic language texts. We have proposed a model that chunks input text into segments and performs language identification on these segments at the same time. Precision and recall are low, leaving a lot of room for further work. Some of our plans are as follows. We would like to annotate more test data, at least 100 tweets in each language. Although In (King and Abney, 2013), the authors label on a per word level, yet we would like to include supervised methods and features talked about in this research to improve our efficiency while dealing with segments. We would also like to attempt our method using higher order character n-gram models for backoff, and n-gram word language models for detection.

Acknowledgements

We thank Kevin Scannell for help with providing data from Twitter and for useful comments and suggestions. This work was undertaken as part of the Apertium project in the 2014 Google Summer of Code.

References

William B Cavnar, John M Trenkle, et al. 1994. N-gram-based text categorization. *Ann Arbor MI*, 48113(2):161–175.

Joyce YC Chan, PC Ching, Tan Lee, and Helen M Meng. 2004. Detection of language boundary in code-switching utterances by bi-phone probabilities. In *Chinese Spoken Language Processing, 2004 International Symposium on*, pages 293–296. IEEE.

M. Federico, N. Bertoldi, and M. Cettolo. 2008. IRSTLM: an open source toolkit for handling large scale language models. In *Proceedings of Interspeech, Brisbane, Australia*, pages 1618–1621.

Ben King and Steven P Abney. 2013. Labeling the languages of words in mixed-language documents using weakly supervised methods. In *HLT-NAACL*, pages 1110–1119.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86.

Marco Lui and Timothy Baldwin. 2012. langid. py: An off-the-shelf language identification tool.

Dau-cheng Lyu, Ren-yuan Lyu, Yuang-chin Chiang, and Chun-nan Hsu. 2006. Language identification by using syllable-based duration classification on code-switching speech. In *Chinese Spoken Language Processing*, pages 475–484. Springer.

Erik F Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 142–147. Association for Computational Linguistics.

Yin-Lai Yeong and Tien-Ping Tan. 2010. Language identification of code switching malay-english words using syllable structure information. *Spoken Languages Technologies for Under-Resourced Languages (SLTU'10)*, pages 142–145.

Liang-Chih Yu, Wei-Cheng He, Wei-Nan Chien, and Yuen-Hsien Tseng. 2013. Identification of code-switched sentences and words using language modeling approaches. *Mathematical Problems in Engineering*, 2013.

[4] This could be because of the default models being used.