Universidad del País Vasco
Euskal Herriko Unibertsitatea

UNIVERSITÉ DE LORRAINE

Co-funded by the Erasmus+ Programme of the European Union

A multilingual approach towards improving the linguistic module of a TTS system: Case Navarro-Lapurdian dialect

Author: María Andrea Cruz Blandón
Supervisors: Prof. Inma Hernaez Rioja[1]
Prof. Eva Navas Cordón[1]
Prof. Denis Jouvet[2]
[1] University of the Basque Country

[2] University of Lorraine

European Masters Program in
Language and Communication Technologies (LCT)

Hizkuntzaren Azterketa eta Prozesamendua
Language Analysis and Processing

Master's Thesis

June 2019

Departments: Computer Systems and Languages, Computational Architectures and Technologies, Computational Science and Artificial Intelligence, Basque Language and Communication, Communications Engineer.

Abstract

The Navarro-Lapurdian dialect is a Basque dialect spoken in the French side of the Basque country. This dialect differs from the standard Basque in terms of its phonology, as well as at the grammatical and lexical levels. Additionally, passages in this dialect are code-switched texts with French. TTS systems for this dialect need to handle both Navarro-Lapurdian and French phonemes repertoire. Inaccurate processing of the French words can result in using the Basque phonology to transcribe them or even in a wrong verbalisation. Previous TTS system has shown that failing to identify and correctly preprocess the French words cause a drop in the quality of the system.

In this work, we propose a multilingual approach for the linguistic module of the system to improve the phonetic transcription of French words. We included a language identification (LID) task at the first stage of the process and a multilingual Grapheme-to-Phoneme (G2P) model at the last stage. A Max-Entropy classifier and a Conditional Random Field (CRF) classifier are used to identify the language at the word-level. Besides, the Transformer architecture, a deep neural network, is used to train the multilingual G2P model. CRF outperforms the Max-Entropy classifier achieving a 0.828 F1-measure for the French words in the LID task, showing an improvement of 0.126 over the Max-Entropy classifier. The best G2P model trained on monolingual and code-switched sentences and tested on the code-switched corpus achieves a PER of 6.96% and a WER of 14.13%.

Keywords: Code-switching, Multilingual G2P, Language Identification, TTS systems, Deep Neural Networks, CRF classifier

Acknowledgements

First of all, I would like to thank my supervisors, professors Inma Hernaez, Eva Navas and Denis Jouvet, for their support and patient guidance throughout this project. I would like to thank all the members of AhoLAB; our conversations were always helpful and fruitful. I very much appreciate my stay at the laboratory. I have learnt a lot during that time.

I would like to thank Elhuyar company for the sponsorship to this project. I would like to express my deep gratitude to the European Union Erasmus Mundus programme to allow me studying as a scholarship holder of the Language and Communication Technologies (LCT) programme. I wish to extend my thanks to my colleagues in the LCT master's programme, all the people I have met during these two years, professors and students.

I would also like to thank Talita Anthonio for her help and advice on the planning of this project. I would like to offer my special thanks to Alejandro Valdés for his support during this time, and his advice and assistance in keeping my progress on schedule.

Finally, I wish to thank my family for their support and encouragement throughout my study. Without their support, this project would not have been possible.

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

Acronym/ Abbreviation	Description
AhoTTS	TTS system developed in AhoLAB
API	Application Programming Interface
BAC	Basque Autonomous Community
BPTT	Back-Propagation Through Time
CNN	Convolutional Neural Networks
CRF	Conditional Random Fields
CS	Code-Switching
EM	Expectation Maximisation
G2P	Grapheme-to-Phoneme
HHM	Hidden Markov Model
HSMM	Hidden Semi-Markov Model
HTS	Hidden Markov Model/Deep Neural Network-based Speech Synthesis System
IPA	The International Phonetic Alphabet
IXA pipes	Natural language processing tools developed by the research group IXA
L-BFGS	Limited-memory Broyden-Fletcher-Goldfarb-Shanno (an optimisation algorithm)
LID	Language Identification
LM	Language Model
LMBR	Lattice Minimum Bayes-Risk
LSTM	Long Short-Term Memory
Max-Entropy	Maximum Entropy
ML	Machine Learning
MT	Machine Translation
NE	Name Entity
NER	Named-Entity Recognition
NL	Navarro-Lapurdian dialect
NLP	Natural Language Processing
NLTK	Natural Language ToolKit
NMT	Neural Machine Translation
NN	Neural Networks
OOV	Out-of-Vocabulary
PER	Phoneme Error Rate
POS	Part-of-Speech
RNNLM	Recurrent Neural Network Language Model
SAMPA	The Speech Assesment Methods Phonetic Alphabet

Acronym/ Abbreviation	**Description**
Seq2Seq	Sequence-to-Sequence
SGD	Stochastic Gradient Descent
SIWIS	Spoken Interaction with Interpretation in Switzerland
SMS	Short Message Service
SMT	Statistical Machine Translation
TTS	Text-To-Speech
UTF-8	Unicode Standard
WA	Word Accuracy
WER	Word Error Rate
WFST	Weigthed Finite-State Transducers

Table 1: Table of acronyms and abbreviations

1 Introduction

Text-To-Speech (TTS) is the field of study that converts text input into speech. Nowadays, TTS systems are widely used in different everyday applications, such as news readers or automatic call centres. Through the years, the researchers have developed different techniques to implement these systems going from concatenation synthesis to deep neural networks approaches. Nevertheless, independently from the chosen technique, these systems require a big amount of data to be trained to achieve high-quality in terms of naturalness, intelligibility and comprehensibility. Few data or phonetically unbalanced corpora can lead to a poor model of the different phonemes and therefore to poor quality.

Although it is desirable to have a large corpus to train a TTS system, this is not always possible. Notably, this is the case for the Navarro-Lapurdian Basque dialect, an under-resourced dialect spoken in the French side of the Basque country. The first TTS system for this dialect is described in (Navas et al., 2014). In that work, they used the architecture of the TTS system for standard Basque (namely, Batua) described in (Hernaez et al., 2001) and adapted it to the dialect. The adaptation included adjusting the phoneset and the recordings of 4,000 Navarro-Lapurdian utterances, which correspond to our knowledge, to the only available spoken corpus for this dialect. Despite the fact that native speakers of the dialect preferred the TTS for the dialect over the one for the standard Basque, the system fails to correctly pronounce the occurrence of French words which impacts the quality of the system.

Besides being different from the standard Basque language in the Navarro-Lapurdian dialect, the texts in this dialect usually code-switch with French. Code-switching presupposes an additional challenge for the TTS system: in addition to adapting the phonemes of the dialect, it must include the phonemes of the French language which were scarce in the built corpus. In (Pierard et al., 2016), the authors explored what they called 'surgery' to overcome the scarcity of the data. They approached the enhancement of the system by identifying the phonemes that had a lousy realisation in synthetic speech. Since the system is a Hidden Semi-Markov Model (HSMM)-based system (based on AhoTTS), they proposed to conduct surgery upon the problematic HSMM states, by transplanting phonemes from a model trained with French phonemes. The results, although better, are not statistically significant in some phonemes.

Usually, TTS systems are made of two modules, the linguistic module and the synthesiser; this is the structure of the previous systems. In the linguistic module, the text is preprocessed so the phonetic transcription can be obtained and passed to the synthesiser to create the synthetic speech. The overall quality of a TTS system relies not only on high-quality recordings but on the accuracy of the linguistic module.

This work aims to improve the processing of French words in the linguistic module. Currently, the phonetic transcription of French words are obtained by means of a dictionary; this means that when new French words are found, the system will preprocess them as Basque words. On the other hand, since Basque is an ergative-absolutive and agglutinative language, there are cases in which some French words or proper French names will have the declination in Basque, in such cases we need to identify both morphemes so they can be transcribed using the phonology of each language. We propose a multilingual approach to handle code-switched texts that automatically transcribes the French words.

1.1 The Navarro-Lapurdian dialect

According to some linguists, the Basque language is a pre-Indo-European language, which is considered an isolated language (Lakarra et al., 1995). Moreover, the Basque language corresponds to a minority language spoken in multilingual communities (Lasagabaster, 2007). This language has historically been spoken in the area of the Basque Country, a region that has its northern part in France and its southern part in Spain. Today we talk about three administrative regions, the Basque Autonomous Community (BAC) that comprises: Bizkaia, Gipuzkoa and Araba; the Chartered Community of Navarre, both BAC and Navarre located in Spain and the Atlantic Pyrenees Department (also known as Iparralde in Basque) located in France (Lasagabaster, 2007). As stated in (Basque Government, 2011), in 2011 there was about $714,000$[1] Basque's speakers and about $388,000$ passive bilinguals (people who may understand but who do not speak Basque), where most of the speakers are in the BAC region. The vast majority of the population speak either Spanish or French. About 58.4% of the population do not speak Basque and the percentage increases for the Navarre and Iparralde regions.

From a syntactic perspective, Basque is an ergative-absolutive language, that means that in Basque there is a distinction between the object and agent in intransitive and transitive verbs, in contrast with nominative-accusative languages like English that only has the distinction with the object of the transitive verbs. For example[2]:

Example 1.1
[English] He has gone home
[Basque] Hura etxera joan da

Example 1.2
[English] He has killed him
[Basque] Hark hura hil du

In 1.1 we have the case of an intransitive verb (*to go* in English, and *joan* in Basque), the absolutive case for the third person is *hura*(the agent of the verb). Whereas in 1.2 we have the case of a transitive verb (*to kill* in English, and *hil* in Basque), where the ergative case for the third person is *hark*(the agent of the verb) and the absolutive case for the third person is *hura* (the object of the verb). On the other hand, the Basque language is an agglutinative language, that is, the main morphological mechanism is the agglutination; words are built by several morphemes, where each morpheme usually is a unit of meaning. For instance, in 1.1 the word *etxera* corresponds to **etxe**-*ra*, the stem is *etxe* which means

[1] Population aged 16 and over

[2] Examples extracted from the Linguistics for Natural Language Processing course notes, University of the Basque Country, 2018

'*house*' and the bound morpheme is *ra* meaning '*to the*', which corresponds to the allative case to indicate the direction of the motion.

For various social, political and linguistic reasons, there exist different Basque dialects. The first person who compiled and wrote about the dialects was Louis Lucien Bonaparte. He distinguished eight different dialects following the regions where Basque was spoken around 1860 (Lakarra et al., 1995). The number is reduced now to five dialects, namely, Biscayan or Western, Gipuzkoan or Central, Upper Navarrese, Navarro-Lapurdian and Souletin (Zuberoan).

As a consequence of the administrative division of the Basque country, the evolution of the Basque language has been partially different in the different regions. During the 16th and 17th centuries, there were no norms that ruled the writing style of Basque texts; it was the Labourdin coastal dialect the reference for the writers "due to its highly refined style" (Zuazo, 1995). Nevertheless, after the French revolution, the Basque language lost its social status and French was declared the only official language, that affected the use of Basque in Iparralde. It was not until the end of the 18th century that the Navarro-Lapurdian emerged as a literary dialect (Zuazo, 1995). In contrast, in the Spanish part of the Basque country, the language did not have the literary status either an official status, and it was even forbidden under General Franco's dictatorship. It was after the civil war that the language started to gain official status. There were several efforts to unify the language in a dialect that allows the communication between the different speakers of the different dialects. In 1968, Euskaltzaindia, the Academy of the Basque language presented the Euskara Batua, the unified Basque (Zuazo, 1995).

Nowadays, the Euskara Batua is the dialect used on the media, newspapers and academia as part of the linguistic policy to motivate the use of Basque. Contrarily, the Navarro-Lapurdian is spoken by about 73,000 speakers on the French side of the Basque country(Basque Government, 2011). Although the Navarro-Lapurdian dialect was a literary reference, in 2011 only 30.5% of the population in the Northern Basque country spoke Basque. The standardisation effort has been focused on the Euskara Batua and has not had the same results for the other dialects. Thus, texts written in Navarro-Lapurdian follow different norms, and there is not a strict norm that governs the writing, for instance, aspects of the use of a hyphen to separate the morphemes, in dates either *1989-ko* or *1989ko* are accepted. In table 2, we list some of the differences between the Euskara Batua and the Navarro-Lapurdian dialect[3].

1.2 Text-To-Speech Systems: Linguistic module

TTS systems are used in a wide range of applications, from online newspapers reading to synthetic voices for laryngectomees (a patient who has undergone a laryngectomy, surgical removal of all or part of the larynx). A TTS system should process a text input and based on that, predict the voice signal. The text processing comprises several steps done

[3] Examples extracted from `https://www.hiru.eus/es/lengua-vasca/caracteristicas-del-navarro-labortano` (accessed: April, 2019)

Aspect	**Description**	**Navarro-Lapurdian**	**Standard Basque**
Phonetics	Graphemes `r` and `rr`	voiced uvular trill/fricative	voice alveolar trill
	Grapheme `h`	unvoiced glottal fricative	It is not pronounced. It lost the aspiration
Phonology	Diphthong. Case `ei`	hog**oi** *twenty*	hog**ei**
	Vowel alternation. Case `e`	z**o**nbait *some*	z**e**nbait
	Drop of vowels and devoicing	beran`t` *to delay*	beran`du`
	Sibilants at the beginning of words	`s, z, x` are more frequent than the fricative `tx`	
Morphology	Emphatic Pronouns (Rebuschi, 1995)	nerini/nihaur, 1. sg guhaur, 1. pl hihaur, 2. sg zuhaur, 2 sg polite zuhiauk, 2. pl (hura) bera, 3. sg (hek) berek, 3. pl	neu, 1. sg geu, 1. pl heu, 2. sg zeu, 2 sg polite zeuek, 2. pl (hura) bera, 3. sg (haiek) beraiek, 3. pl
	Suffix to answer interrogative forms zertako (*For what*) and zergatik (*why*)	-kotz Elgarrekin bizitzekotz *together-With live-For* *To live together*	-ko Elkarrekin bizitzeko
Syntax	Quantifier order	They can be placed either at the left or right side of the noun. asko euskaldun *many Basques*	They go after the noun euskaldun asko
	Word order	More flexible than in standard Basque	

Table 2: Some differences between the Navarro-Lapurdian dialect and the Standard Basque (Euskara Batua). Abbreviations: pl.: Plural, sg.: Singular

in what is called the linguistic module (or front-end) of a TTS system. That module aims to preprocess the text so it can be transformed into its linguistic representation, which will be used later for the synthesiser module (or back-end) to generate the synthetic speech. Figure 1 shows a general scheme of a TTS system.

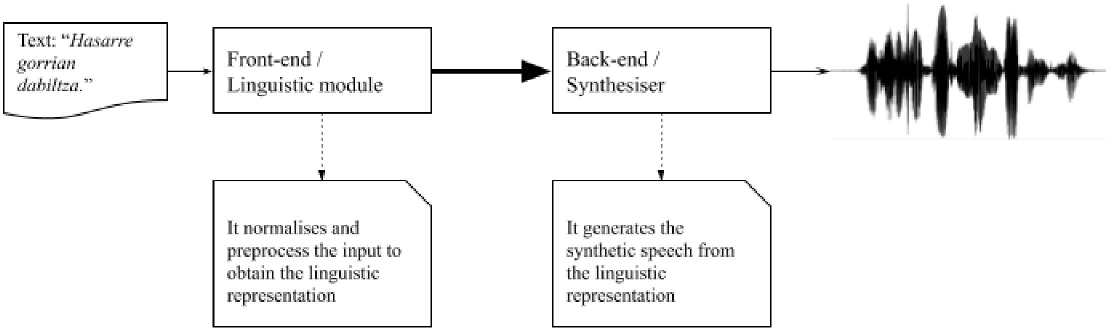

Text: "*Hasarre gorrian dabiltza.*"
Front-end / Linguistic module
Back-end / Synthesiser
It normalises and preprocess the input to obtain the linguistic representation
It generates the synthetic speech from the linguistic representation

Figure 1: General scheme of a TTS system

Briefly speaking the linguistic module needs to predict all the linguistic features to reconstruct the speech signal from them. The process is not a trivial conversion; to begin with, in TTS synthesis, we are going from a discrete space (characters, words) to a continuous one (speech signal). There are aspects of the speech that are not strictly encoded in the text, like the intonation pattern. Also, we can find cases where the text being analysed can have different possible options for pronunciation. That being said, there is an extensive work done in the linguistic module so the system can produce a clean (unambiguous) and the most comprehensive representation of the linguistic features. Although the back-end of a TTS involves several and complex processes to predict a continuous speech signal from the linguistic features, in this work, we focus our attention on the steps carried out in the linguistic module. For further information about the techniques used in synthesis we recommend to consult (Rabiner and Schafer, 2007; Taylor, 2009), and (Adiga and Prasanna, 2018) for a compilation of the last techniques to model the acoustic features for statistical parametric speech synthesis.

In (Taylor, 2009), the author makes a clear separation between the form and the writing; the form is clean, abstract and unambiguous, whereas the writing is considered a noisy signal. We draw our attention to this distinction as a helper to distinguish the steps to decode the writing. To illustrate the relation between the writing and the form consider the sentence: "In NYC there is a present every 15 days.". To process this sentence our system should identify that `NYC` is an abbreviation that means "New York City", that `15` is the numeric representation of "fifteen" and that the word `present` is the noun and not the verb which results in a different pronunciation.

The linguistic module is a series of tasks on the basis thereof. For some of those tasks, the order in which they are arranged has an impact on the result. Whereas, others can be performed in parallel. We summarise the main tasks explained in (Taylor, 2009).

1. **Pre-processing:** Even though UTF-8 is widely used as the computer encoding standard, other standards are also used. Hence, our system needs to preprocess

the text, so all the tokens are encoded in the same encoding. On the other hand, the system should identify the tokens of different languages so that the appropriate modules can process them in further tasks. After that, we need to tokenise by sentences and words. Our system should be able to distinguish between the number separators (`,.`) and the punctuation symbols.

2. **Semiotic system identification:** As we show with the previous example, a text can encode not only natural language but other semiotic systems. A crucial task is to accurately identify the semiotic system of each token in the text so further processing can be accurately done. For instance, we need to identify whether a number is a cardinal, ordinal number, or a date, probably context and patterns help to resolve the ambiguities. Additionally, the system should be able to differentiate between acronyms, abbreviations, and uppercase titles. A text normalisation task can handle the uppercase texts while acronyms and abbreviations will need further processing.

3. **Decoding and Parsing:** Once a semiotic class has been assigned to a token, the system needs to find the underlying form of it by using the appropriate parser for the semiotic class. It can be the case that for some classes, like abbreviations and acronyms, we need to look up in the lexicon. There is no single rule to decode acronyms. Some are pronounced as the literal word they form, for example "NATO" (North Atlantic Treaty Organization) pronounce as /neIt@U/[4]; and others pronounce letter by letter as in "IBM" (International Business Machines) pronounce as /VIbi:"Em/. Hence the system should have a lexicon as complete as possible, for acronyms not found in the lexicon a good strategy is to use the letter by letter rule.

4. **Verbalisation:** We must not confuse decoding and parsing with verbalisation. Whereas in the decoding and parsing phase, we identify only one underlying form for the token, in verbalisation, we can have different options to *translate* non-natural language text. The election can be based on the application of the system or the language preferences of the users (the case for dialects). Once a decision has been taken, the verbalisation task can be merely a mapping function depending on the semiotic class. For example, the verbalisation for the cardinal number `15` is "fifteen".

5. **Disambiguation:** One characteristic of natural language is ambiguity. There are ambiguities at all levels of interpretation of a sentence. We encounter syntactic ambiguity, the classic example of "I saw a man with the telescope" which has two possible interpretations where the prepositional phrase (PP) "with the telescope" can be linked to the noun phrase (NP) "a man" or the verbal phrase (VP) "saw". Semantic ambiguity, which is related to the meaning of the sentence. Lexical ambiguity in which case a word can have different part of speech labels. In the case of TTS systems, we are interested in homographs, words that are written the same

[4] Using SAMPA: Speech Assessment Methods Phonetic Alphabet phonetic transcription.

but which pronunciation varies according to its meaning, the example of the word "present" that is pronounced /prEz@nt/ if it is the noun or /prI"zEnt/ if it is the verb.

6. **Phonetic transcription:** One step previous to the speech signal prediction is the phonetic transcription. In this task, the system transforms the unambiguous words decoded from the text to their phonetic representation. There are two big projects of phonetic alphabets, namely The International Phonetic Alphabet (IPA)[5] and The Speech Assessment Methods Phonetic Alphabet (SAMPA)[6]. These alphabets encode all the possible sounds (phonemes) that humans can make into symbols. Nonetheless, the phonetic transcription is not merely a mapping from characters to phonetic symbols. Words are not pronounced the same when they are pronounced isolated or in sentences. Articulatory processes are going on when we pronounce, that changes the canonical pronunciation of a word. In texts, there is often a separation between words (i.e. a blank space), but this is not the case in the speech signal. Hence, the system needs to predict the articulatory processes correctly; otherwise, the output signal can sound unnatural. The phonetic transcription can be performed employing the lexicon in combination with an automatic approach.

7. **Prosody prediction:** The phonetic transcription by itself is not sufficient to predict the speech signal. A human will barely speak with a flat rhythm; instead, we usually use different intonation patterns, among other reasons, because the prosody encodes information about the speakers' attitude and emotion. It is worth to mention that ordinarily, the text that needs to be synthesised is written to be communicated by writing and not by speech. That is, any information concerning the verbal content and style is not fully encoded in the texts, or it is absent. However, some features can be predicted, like the phrasing, in many languages, questions will have a rising pitch pattern. For example, English or Spanish. In this task, the system predicts phrasing (prosodic phrase breaks), prominence and intonational tune.

1.3 Code-switching

Code-switching (CS) is the alternation between two or more languages or dialects of the same language within the same conversation. CS is a phenomenon that frequently occurs in multilingual societies, such as in India. Nowadays, we can claim that with the connectivity and the globalisation, this phenomenon is also happening in monolingual societies. There are several issues of linguistic interest in such environments, for example, language representation on bilinguals' mind and why the speaker switches. One may not confuse borrowing with code-switching, as in borrowing the phonetic realisation of the loanword is adapted following the phonology of the recipient language.

[5] `https://www.internationalphoneticassociation.org/` (accessed: April, 2019)

[6] `https://www.phon.ucl.ac.uk/home/sampa/index.html`(accessed: April, 2019)

There are two main distinctions concerning the switch: intra-sentential, code-switching within a sentence and inter-sentential, code-switching between sentences. However, code-switching at the morpheme-level has also been found in agglutinative languages, for example in the pair Turkish-English (Boztepe, 2003):

Example 1.3
Sen-inle bu konu-da CONFLICT-imiz var.
you-PREP this issue-PREP conflict-POSS PRONOUN (1 ST PLURAL) exist.
[English] We (You and I) have a conflict (disagreement) over this issue.

In code-switching, the main code in a code-switched utterance to which a majority of phonological and morphological features of the discourse can be attributed is called base or recipient code (linguistic variety). Myers-Scotton's model (Boztepe, 2003) proposes that there is always an asymmetrical relation between the languages. Thus they introduced the matrix language (ML) and the embedded language (EL) terms. The matrix language is the language dominating the sentence in terms of the syntactic and morphological relations. In the Navarro-Lapurdian dialect, the matrix language is Basque, and the embedded language is French, see example 1.4 and 1.5, in bold the French words. The first example shows code-switching at the morpheme-level. Also, the second example illustrates inter-sentential code-switching. Examples extracted from the journal Herria.

Example 1.4
- Baiona mailaz jautsiko da, joanden ostiralean berdinketa ardietsirik ere ***Orléans****-eko zelaian, 11.*
- Bayonne level-FROM relegate FUT, last Friday tie reach-even though Orleans-OF pitch-AT, 11
[English] - Bayonne will be relegated, even though they tied last Friday at the Orléans pitch, 11.

Example 1.5
La deuxième vérité*: Frantzia iparraldeko hiri ttipi batean hatzemaiten dute neska bat hila*
The second truth: France northern city small one-IN find PAST-PART girl one dead
[English] The second truth: A dead girl has been found in a small town in northern France.

1.4 Master's thesis statement

Although the TTS system developed in (Pierard et al., 2016) improved the quality for the dialect, the inappropriate pronunciation of the French words makes the system fails in terms of naturalness and intelligibility. We identify two tasks that can be adapted to a multilingual approach in the linguistic module to process the French words better. We propose to add a language identification task in the preprocessing phase that will serve as a filter to redirect the words to the correct pipeline process. Currently, the phonetic transcription is done by using a rule-based approach for Basque words and by dictionary entries in the case of French words. We propose to use a multilingual Grapheme-to-Phoneme (G2P) model that can transcribe both languages once the language of the word has been identified.

The main application for which the TTS system was built is as the assistant reader of online newspapers. Under this application context, a reasonable assumption is that the texts are well-written in Basque, following the grammar and declination rules of the dialect. Although the assumption is valid to some extent, particularly, there is no such thing as a standard for the code-switching with French. Usually, the texts code-switch with proper names, but it is not limited to that.

One interesting phenomenon to analyse is the code-switching at the morpheme level; the base language is Basque that means that the declination cases are applied over the French words when needed, and free morphemes can be attached to French words following the agglutinative morphology. How does the morphophonology of both languages interact at the morpheme boundaries? In this work, we do not intend to fully answer this question as it requires a large amount of data to have statistical support to draw any conclusion, but we want to draw our attention to these cases as they are of linguistic interest.

1.5 Master's thesis outline

This document is organised as follows: in section 2 we discuss the techniques that have been used to address the problems of language identification and Grapheme-to-Phoneme, particularly for the case of code-switched texts. In section 3, we explain our proposal, the technologies used, the corpora employed to train the different models and the metrics to measure the performance. Section 4 contains the results of the different experiments and the analysis of those. Finally, section 5 gives an overview of the work done and suggests further steps for continuing the study.

2 Literature Review

In this section, we explore some techniques that have been used to perform the language identification task and the grapheme-to-phoneme conversion. It should be noted that there are several methods for monolingual contexts, that is when the input text is written in only one language. However, over the last years, the multilingual information on the internet has increased, and access to data is becoming more convenient. Researchers have turned their attention to multilingual perspectives as well. We review here both approaches and how they have been used for the case of code-switching.

2.1 Language Identification Problem

When it comes to working with CS, one of the most common tasks in Natural Language Processing (NLP) is the Language Identification (LID) task (Rosner and Farrugia, 2007; Vyas et al., 2014; Sitaram and Black, 2016; Rallabandi and Black, 2017). Despite the fact of better processing the words once we know their language, one can also take advantage of the monolingual resources of the specific language. It is often the case that there are more available monolingual resources than code-switched ones. In contrast to LID for monolingual documents, LID for code-switched texts is intended to be at the word-level. That level of granularity makes this process a challenging process, and it is still an unsolved problem as evidenced on the last two shared tasks on LID in code-switched data (Solorio et al., 2014; Molina et al., 2016).

Several strategies have tackled this problem, but one recurrent approach is the use of character n-grams either as features for Machine Learning (ML) algorithms or to build language models. One reason to use them is the relation that exists between the language and the distribution of the character n-grams. Zipf's law can show this relation. Zipf's law establishes an inverse proportional relationship between the frequency of a word and its position in the decreasing rank of words. The corpus gives the relation factor, and it will be constant throughout the whole corpus; see equation 1, where f is frequency, r is the position in the rank, and k is the constant for the corpus. In (Ha et al., 2003), the authors found that Zipf's law is not only valid at word-level, but it is also valid for syllable- and character-level. Furthermore, characters n-grams can encode aspects of the morphology (Kulmizev et al., 2017), which can be a significant distinction between two distant morphological languages. We will see the importance of these two statements for our experiments in section 3.1

$$f = \frac{k}{r} \tag{1}$$

In the following sections, we are going to explain some of the techniques used to carry out LID for code-switched texts.

2.1.1 Hidden Markov Model (HMM) approach

In (Rosner and Farrugia, 2007), the authors developed a LID process based on HMM to identify words in Maltese-English code-switched SMS messages for a TTS system. In their approach, the authors proposed a 2-node HMM, modelling a bigram language transition change. The authors calculate the distributions of the HMM model as described in equations [2, 4], where $l_i \in L$, L is the set of languages: English and Maltese.

$$\pi_i = \frac{count(l_i)}{count(\text{tagged samples})} \tag{2}$$

$$a_{ij} = \frac{count(l_i l_j)}{count l_i} \tag{3}$$

$$b_i(w) \approx \frac{count(\text{w tagged as } l_i)}{count(l_i \text{ tokens})} \tag{4}$$

The authors included a bias number ($\alpha = 4$) of entries of the word in the corpus tagged as l_i if the word was found in the dictionary of the l_i language. The bias was used as a feature for preference. To avoid impacting the probabilities, they also included a small number ($\beta = 1$) of the word tagged as the other language. For the case of unknown words, also known as Out-of-Vocabulary (OOV) words in the literature, they used a trigram Markov Language model to find the probability of the word belonging to one of the languages. They trained the model with 200 real-world Short Message Service (SMS) messages, and tested it with 100, obtaining an accuracy of 96.5% in their best configuration.

2.1.2 Machine Learning approaches

(King and Abney, 2013) explored weakly supervised methods to perform LID in mixed-language documents. Among the methods explored, the authors trained a Logistic Regression with Generalised Expectation. They estimated the marginal label distribution utilising a regular supervised naïve Bayes classifier. This approach was later tested in English-Hindi code-switched social media data (Vyas et al., 2014), which also included a code-switching probability to model the context. Their LID task reached an accuracy of 84.6%.

Shared tasks on LID in code-switched data have been a source of state-of-the-art algorithms. In (Solorio et al., 2014), the teams worked on four pairs of languages (Mandarin-English, Modern Standard Arabic-Arabic dialects, Nepali-English and Spanish-English). In (Molina et al., 2016), the teams worked with two pairs (Modern Standard Arabic-Arabic dialects and Spanish-English). Most of the teams in both tasks proposed solutions based on Conditional Random Fields (CRF) showing the best $F1$-measure for the majority of the pairs of languages.

CRF was introduced in (Lafferty et al., 2001) for labelling sequences. In CRF, we have two random variables X and Y, where X ranges over the data to be tagged, in our case, that would be words from natural language. Y ranges over the labels; in our case, the languages we want to identify. These two random variables are jointly distributed and modelled as a graph $G = (V, E)$, where V is the set of vertices or nodes, and E are the edges or links. (X, Y) is a conditional random field where Y follows the Markov property with respect to the graph when it is conditioned on X as described in equation 5, where v and w are nodes in the graph and $w \sim v$ means that w and v are neighbours in G.

$$p(Y_v|X, Y_w, w \neq v) = p(Y_v|X, Y_w, w \sim v) \tag{5}$$

Under the assumption that G has a linear chain graph structure, the joint distribution over Y given X is denoted by equation 6; where y is the sequence of labels and x the data sequence, λ_k and μ_k are parameters of the model and f_k and g_k are given and fixed features.

$$p_\theta(y|x) \propto exp(\sum_{e \in E,k} \lambda_k f_k(e, y|_e, x) + \sum_{v \in V,k} \mu_k g_k(v, y|_v, x)) \tag{6}$$

To determine the parameters $\theta = (\lambda_1, \cdots ; \mu_1, \cdots)$ from the training data $D = \{(x^{(i)}, y^{(i)})\}_{i=1}^N$ with the empirical distribution $\tilde{p}(x, y)$ they use the objective function shown in equations 7 and 8.

$$\begin{aligned} O(\theta) &= \sum_{i=1}^{N} logp_\theta(y^{(i)}|x^{(i)}) && (7) \\ &\propto \sum_{x,y} \tilde{p}(x, y) logp_\theta(y|x) && (8) \end{aligned}$$

CRF was one of the methods explored in (King and Abney, 2013), which showed the best results for the LID task. In (Solorio et al., 2014), there was not an overall winner algorithm for all of the language pairs, but the CRF proposed in (Chittaranjan et al., 2014) was on top three for several test sets for each pair language. The authors developed a CRF with 27 features: 3 Capitalisation features, 3 Contextual features, 16 Special character features, 4 Lexicon features and one character n-gram feature. The character n-gram feature was obtained using binary Maximum Entropy classifiers trained with monolingual words. The probabilities obtained are later binned into ten equal buckets, more details about the features can be found in section 3.1. They achieved an accuracy of 95.3% for Nepali-English pair.

2.2 Grapheme-to-Phoneme Conversion

In G2P conversion, we want to obtain the phoneme sequence from the written word; we can see this as going from a character to a phoneme sequence. The idea of using G2P algorithms arises from the need to get the phonetic transcription of OOV words. To get the phonetic transcription of a word, the system will look up first in the lexicon, and if there is no entry for it, the system will classify it as an OOV word. Then the TTS system is expected to apply the morphophonology that governs the pronunciation as a human would do it.

One of the most traditional strategies is rule-based (Taylor, 2009); this approach followed the dictionary look-up method. In rule-based systems, the transcription is done by subsequent application of usually context-sensitive rewrite rules. These rules are of the form $A \rightarrow B || L \_ R$ where A is rewritten as B under the left context of L and the right context of R. The process of creating the rules requires a linguist expert in the language of study since it is a manual task. The rule-based approach showed good results for consistent languages such as Spanish but it poorly performed for more irregular languages such as English. Example of systems using rule-based G2P are (Wypych et al., 2003), (Braga et al., 2006) and (Nair et al., 2013).

Although with knowledgeable approaches, we can cover all the morphophonemic rules, those approaches are costly in terms of time as the rules are hand-written. The data-driven strategies, on the other hand, can learn the rules from the data itself, which make them a suitable choice to cover the irregularities of the languages. The hypothesis is that the more examples we have, the more accurately the model will be able to infer the rules. This assumption does not restrict the use of the lexicon as it is complimentary. Under these strategies, we have Statistical approaches, Weighted Finite-State Transducers (WFST) and Neural Network (NN) approaches.

2.2.1 Statistical Approaches

One of the most popular G2P conversion models is the Joint-Sequence model presented in (Bisani and Ney, 2008). In this model, the G2P is formulated with the Bayes' decision rule, see equation 9, where $g \in G^*$ and G is the set of all graphemes, and $\varphi \in \Phi^*$ and Φ is the set of all phonemes.

$$\varphi(g) = \underset{\varphi\prime \in \Phi^*}{\operatorname{argmax}} \, p(g, \varphi\prime) \tag{9}$$

The hypothesis is that there is a relation between input (graphemes) and output (phonemes) symbols so that they can be generated by the same sequence of joint units (graphones). That is, units that combine both input and output symbols. A graphone is a tuple of the form $q = (g, \varphi) \in Q \subseteq G^* \times \Phi^*$. A sequence of graphones is a particular join segmentation or alignment that partitions the grapheme and phoneme sequences into an equal number of segments; the authors called this alignment as a many-to-many alignment. Under this definition, different alignments are allowed, and therefore, ambiguities can arise. Accordingly, they calculated the joint probabilities by summing all the probabilities of the possible graphones. See equation 10; where $q \in Q^*$ and $S(g, \varphi)$ is the set of all co-segmentations (alignments) of g and φ. They introduced a new symbol to model phenomena happening at word boundaries.

$$p(g, \varphi) = \sum_{q \in S(g,\varphi)} p(q) \tag{10}$$

The phoneme sequence is determined using the Expectation Maximisation (EM) algorithm, in which the co-segmentation of the joint units is a hidden variable. See equation 11, where h correspond to the sequence of preceding graphones, and $e(q, h; \vartheta)$ is what they called evidence for q, that is, the expected number of occurrences of q in the training set under the current set of parameters ϑ. This method can infer both alignments and subsequence chunks.

$$p(q|h; \vartheta\prime) = \frac{e(q, h; \vartheta)}{\sum_{q\prime} e(q\prime, h; \vartheta)} \tag{11}$$

2.2.2 Weighted Finite-State Transducers (WFST)

In (Novak et al., 2012) a WFST-based toolkit for G2P conversion is presented, Phonetisaurus. The authors describe the problem of G2P as a set of three main tasks: Sequence alignment, Model training and Decoding. The sequence alignment is the task of aligning both grapheme and phoneme sequences in a training dictionary. The model training task is getting a model for new instances, and the decoding task is finding the most likely phoneme sequence.

In their proposal, the alignment is done through a lattice representation. They modified the EM-driven multiple-to-multiple alignment algorithm proposed in (Jiampojamarn et al., 2007), so it allows one-to-many and many-to-one arcs at the initialisation. It calculates the forward and backward probabilities of each entry in the dictionary and normalises the probabilities to avoid zero weights in the lattice. The model training uses a joint N-gram model where the aligned sequences are converted into label sequences which are used to train the N-gram model. The resulting N-gram model is transformed into a WFST.

Finally, they used three techniques to perform the decoding task. 1) Best short (lowest cost path through a composition of the word with the G2P model and a projection of the output labels. 2) Parallel Recurrent Neural Network Language Model for the N-best reranking. 3) By appealing to the similarities between the Statistical Machine Translation (SMT) problem and G2P conversion, they tried Lattice Minimum Bayes-Risk (LMBR) decoding. LMBR was proven to have good results in SMT lattices. In LMBR the lattice obtained with the G2P model is scaled by a factor and passed as the input to calculate the best path in an intermediate lattice. The intermediate lattice built by means of the N-grams. They use Word Accuracy (WA) to evaluate the performance of the toolkit with different configurations. The Phonetisaurus reached the best results with the Recurrent Neural Network Language Model (RNNLM) approach for the English dictionaries CMU-dict, NETalk15K and OALD.

2.2.3 Neural Networks Approaches

From the similarities with the Machine Translation (MT) problem, researchers found inspiration in the satisfactory performance of different NN architectures in that problem. That is the case of (Yao and Zweig, 2015), who proposed two approaches using Sequence-

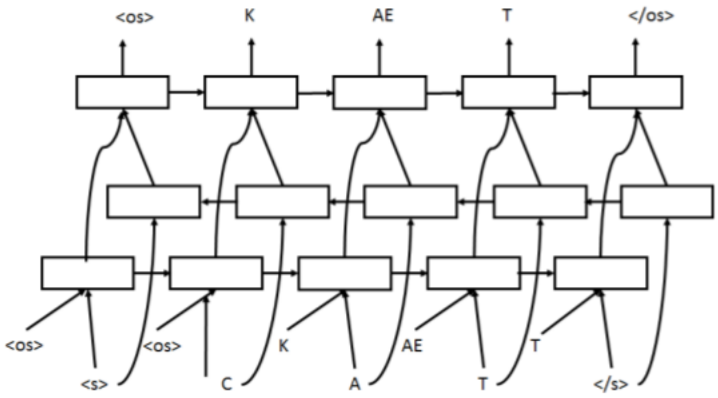


Figure 2: Bi-directional LSTM architecture

to-Sequence (Seq2Seq) models based on conditioned models on source language in Neural MT (NMT). In contrast with the MT problem, the vocabulary sizes of both source and target sets are small, which allow having reliable n-gram models. The authors proposed side-conditioned Language Models (LM) for generation and alignment-based models.

One of the advantages of the side-conditioned LM is that those models do not require explicit alignment information; the prediction of the phoneme is based on the past phoneme prediction and the input sequence. The architecture consisted of an encoder-decoder LSTM where the input was given in reversed order. To train the encoder and decoder networks, they used Back-Propagation Through Time (BPTT) and beam search to generate the phoneme sequence during the decoding phase. The result is selected according to the highest posterior probability.

In the case of alignment-based models, the authors proposed uni- and bi-directional LSTM architectures. For the uni-directional Long Short-Term Memory (LSTM), the posterior probability of a phoneme will depend on the previously predicted phoneme and the input grapheme as described in equation 12, where φ_1^T is the phoneme sequence, S is the alignment, and g_1^T is the grapheme sequence. While for bi-directional LSTM the probability will be conditioned with the whole grapheme sequence (see equation 13), because of the forward and backward networks as shown in figure 2 (Figure is taken from (Yao and Zweig, 2015)). The information of the alignment was calculated using the method described in (Jiampojamarn et al., 2007).

$$p(\varphi_1^T|S, g_1^T) = \prod_{t=1}^{T} p(\varphi_t|\varphi_t^{t-1}, g_1^t) \tag{12}$$

$$p(\varphi_1^T|S, g_1^T) = \prod_{t=1}^{T} p(\varphi_t|\varphi_t^{t-1}, g_1^T) \tag{13}$$

They showed that using alignment information improved the performance of the G2P in terms of Phoneme Error Rate (PER) and Word Error Rate (WER), which will be explained in section 3. They achieved the best result using the bi-directional LSTM with three layers and 300 hidden units. That model outperformed the results obtained with the Joint-Sequence model in (Bisani and Ney, 2008) for the three English dictionaries CMUDict, NetTalk15K and Pronlex.

In an attempt to dispense with the alignment, in (Toshniwal and Livescu, 2016), an attention mechanism is presented as an extension of the encoder-decoder LSTM model. The motivation is given by the fact that alignment is not per se the desired end, but an intermediate result used to predict the phoneme sequence. Thus the information about alignment can be learned through the attention mechanism. In their architecture, the encoder is a stacked bi-directional LSTM that receives as input the sequence of vectors x, resulting after multiplying the one-hot vector representation of the characters and a character embedding matrix. On the other hand, the decoder is a staked uni-directional LSTM. It uses a context-vector c (computed from the last encoder's state) and the projection of the previous prediction with phoneme embeddings for predicting the next phoneme. The authors presented and compared two attention strategies: global attention and local attention.

The global attention consists of having a context-vector for each decoder timestep instead of a unique context-vector. The attention mechanism can be interpreted as a soft alignment, they calculate the context vectors as in equations [14-16], where α_{it} represents the importance of the hidden state h_i to produce y_t, v^T, W_1, W_2 and b_a are parameters learnt during training, d_t is the output of the decoder at time t and Tg is the length of the sequence, the decoder is now conditioned on the context-vector c_t.

$$\begin{aligned} u_{it} &= v^T tanh(W_1 h_i + W_2 d_t + b_a) && (14) \\ \alpha_t &= softmax(u_t) && (15) \\ c_t &= \sum_{i=1}^{Tg} \alpha_{it} h_i && (16) \end{aligned}$$

For the local attention, the authors used two types of alignments: monotonic (local-m) and predictive (local-p). In local attention, the hypothesis is that the context window needed to predict can be smaller than the whole sequence (as in global attention). The first task is to find an aligned position p_t to then calculate the context-window $[p_t - D, p_t + D]$; by experimental search they found $D = 3$ was their optimum. In local-m, the alignment is assumed to be simplistic $p_t = t$ therefore, the attention weights are calculated as in global attention. While in local-p, the model learns to predict p_t using the length of the sequence, in this case, the model favours input positions that are close by p_t by rescaling the attention weights with a Gaussian prior centred at p_t as in the equation 17, where α_{it} is calculated as in the global attention, i correspond to the position being analysed and σ^2 is the variance of the Gaussian distribution.

$$\tilde{\alpha_{it}} = \alpha_{it} \dot{e} xp(-\frac{(i - p_t)^2}{2\sigma^2}) \qquad (17)$$

They trained a 3-layer stacked LSTM with 512 hidden units; they used Stochastic Gradient Descent (SGD) and schedule sampling with linear decay. To predict the phonemes, they used a greedy decoder. Both global attention and local-m attention obtained better results than the bi-LSTM before explained. Although local-m attention is a simplistic approach for MT, it is suitable for G2P conversion. Comparison with uni-directional reversed LSTM shows that bi-LSTM may not be needed for this task, given the short length of the input sequences about six on average. Interestingly, they found that the number of hidden units has a significant impact on the performance of the networks, as using four times more units (from 50 LSTM units to 256 LSTM units) improved the results by 8.7%.

2.2.4 Multilingual Approaches

The approaches described above are monolingual approaches; that means, it is necessary to train a model for each language. On the contrary, multilingual approaches propose to have a single model that can handle two or more languages. These approaches try to leverage the similarities between writing systems and phonetic inventories of the languages. Inspired by multilingual NMT, (Peters et al., 2017) presented a multilingual G2P that seeks to overcome the data-scarcity of low-resource languages. The authors reformulated the G2P problem as a multisource NMT problem where the input sequences can be in different writing systems (e.g. Latin or Arabic) and the output sequences are in IPA.

They used the encoder-decoder LSTM model with a global attention mechanism. Two input models were proposed, one in which a language token was added to the grapheme sequence (LangID). The language token identifies the language of the word being analysed, for example, `<eng> r e a l`. The second model did not include the language token (NoLangID). The architecture of the network consisted of a 2-layer bi-directional encoder and a 2-layer decoder that used a beam width of 100 to predict the phonetic transcriptions; both networks had 150 hidden layers. The model was trained using SGD and a maximum of 10,000 words for each language. To evaluate the performance they employed PER, WER and WER 100, which penalised if the target word is not in the first 100 predictions.

For comparison purposes, the dataset used was the same in (Deri and Knight, 2016), a WFST-based G2P. In the WFST model, the first step is to train the model with high-resource languages and then by means of language and phoneme distances adapt that model to related low-resource languages. They split in three the training data, one only with high-resources, another with languages that were adapted in (Deri and Knight, 2016) and the last one is a set including all the language, a total of 331. Their best results overpassed the results obtained with the WFST model and were reached using the LangID model and training in all languages.

In further experiments, they found that even though they had a lower performance compared with monolingual WFST, their model learned phonemes embeddings that were reasonably clustered. Their model was able to predict phonemes that although outside of the phonetic repertoire, they were similar to the targets. A possible way they suggested to improve the results is to use a reranking strategy based on the language inventory. Furthermore, they found evidence that the model was not only able to learn similarities but negative associations as well. They tested the model with unseen languages using the language token and obtained better results than when using the NoLangID model.

3 Methodology

This section compiles the different methods used for the LID and G2P tasks, the metrics employed to measure the performance of the models and the integration of the best models with the current TTS system.

3.1 LID Models

To accomplish the LID task, we compared the approaches proposed in the shared task (Solorio et al., 2014). This shared task looked for proposals for tackling the problem of identifying the language in code-switched tweets at the word-level. We chose the system proposed in (Chittaranjan et al., 2014) considering it was one of the top systems and was tested in more than two pairs of languages showing stable performance across different languages. In this section, we explain the two systems we developed for the LID task, the corpora used and the metrics used to evaluate the performance of the different systems.

3.1.1 Corpora

To build the corpora for training the CRF, we followed the methodology employed in the original paper. Nevertheless, since we are working with a low-resource dialect, there were not Navarro-Lapurdian corpora available in the datasets for Name Entity (NE), and frequent words. Rather, we used the available resources for the Standard Basque dialect. The resources for Basque will be assumed to be the Standard dialect unless otherwise specified.

To build the NE list, we used the corpora available on DBpedia[7] (version 2016-10), corpora based on the 2016 released version of Wikipedia. We used ten main NE from the ontology defined by DBpedia: `Agent`, `Award`, `Device`, `Holiday`, `Language`,`PersonFunction`, `Places`, `MeanOfTransportation`, `Name`, and `Work`.

The frequent words were obtained from the Leipzig's Corpora(Goldhahn et al., 2012), as well as DBpedia corpora, Leipzig's corpora are based on released versions of Wikipedia. For French, we used Wikipedia version 2010 (Leipzig's Corpora, 2010). For Basque, we used Wikipedia version 2016 (Leipzig's Corpora, 2016a). After obtaining the frequent words lists, we cleaned it by removing entries that contain special characters except for hyphens or apostrophe. Table 3 lists the first 15 words for French and Basque corpora.

To train the Maximum Entropy classifiers, we used the sentences in the Leipzig's Corpora for Spanish (Leipzig's Corpora, 2016c), English (Leipzig's Corpora, 2016b), and French. For the Basque corpus, we employed Leipzig's corpus and the Navarro-Lapurdian corpus, keeping a proportion of 75% words from the Navarro-Lapurdian corpus and 25% from the Standard Basque (Leipzig's corpus). The Navarro-Lapurdian corpus was developed in (Navas et al., 2014); it is a spoken corpus with two native speakers of the dialect who read sentences extracted from newspapers.

[7] `https://wiki.dbpedia.org/downloads-2016-10` (accessed: May, 2019)

French	**Basque**
de	eta
la	da
et	zen
le	bat
à	zuen
l’	ziren
des	izan
les	ere
en	animalia
est	generoko
d’	dira
du	du
un	ez
une	zuten
dans	bere

Table 3: Top 15 of the most frequent words in Leipzig’s corpora French and Basque

To train the CRF classifiers, we needed annotated data per token. All the datasets from Leipzig’s corpora are monolingual corpora, however, bare in mind that some sentences may contain tokens from other languages (e.i. NE, original spellings, ...). Only the Navarro-Lapurdian corpus contains code-switched French-Basque sentences, although it was not annotated. The code-switching in the corpus was mainly to introduce NE in French, such as names of organisations, peoples or places. That is why identifying NE could help us to identify French tokens in the corpus. To build the corpus for CRF, we used the *IXA pipes*(Agerri et al., 2014), NLP tools, to tokenise the sentences in the corpus and to identify the NE.

Once we have the list of tokens for all the 3998 sentences, we automatically created a target annotated file, in which we tagged as Basque (`bqe`[8]) all the tokens, except for the ones that where punctuation symbols or numbers which were tagged as Undefined (`und`). After that, we manually checked every utterance looking for French tokens, whenever found them we updated the target file with the French code for the token (`fra`). During this process, there were occasions where the lemmatisation was not accurate, in which case we updated the source (tokens) and target files. These annotations resulted in 317 code-switched sentences. Besides *IXA tokenisation*, we wanted to compare with another tokenisation. Thus, a general tokeniser from the python package Natural Language ToolKit

[8] Language code according to ISO-639-3 standard

(NLTK) was also used. The process of the annotations for the code-switched sentences was repeated for the NLTK tokenisation. The final version of the annotated corpus was then compiled into a `.json` file, a more convenient format.

Table 4 illustrates the statistics of the corpora and where each corpus was used.

Corpus	**Identifier**	**No. sentences**	**No. tokens**	**Usage**
DBpedia	eu_ne	-	438*	NE list. CRF training
DBpedia	fr_ne	-	61,147*	NE list. CRF training
Leipzig	eu_2016	300,000	419,604	CRF training Max-Entropy classifier training
Leipzig	fr_2010	1,000,000	571,999	CRF training Max-Entropy classifier training
Leipzig	en_2016	10,000	39,460	Max-Entropy classifier training
Leipzig	sp_2016	10,000	39,365	Max-Entropy classifier training
Leipzig	eu_fw	-	500*	Frequent words. CRF training
Leipzig	fr_fw	-	500*	Frequent words. CRF training
Navarro Lapurdian	nl	3,998	13,274	CRF training Max-Entropy classifier training

Table 4: Description of the corpora employed to train the CRF and Max-Entropy classifiers. The numbers correspond to the size of the corpora, but final sentences/tokens used varied as needed. The number of tokens includes special characters and it is case-sensitive. (*) The numbers correspond to the final tokens used for training

3.1.2 Experiments: CRF Classifier

Given the nature of the shared task (LID in code-switched tweets), some features were relevant for that purpose, such as if the token contains the '#' symbol (to identify hashtags). However, in our case, the TTS system is intended to be used for newspaper articles. Under that consideration, the texts are expected to be well-written, and we can reduce the number of features initially suggested by (Chittaranjan et al., 2014). The final set of features is shown in table 5.

To implement the CRF classifier, we used the python scikit-learn extension library sklearn-crfsuit[9]. Unlike the original work, we have experimented with different types of tokenisation, processes for Name Entity identification and representations for the probabilities obtained with the Maximum Entropy classifier. Also, we tried different configurations of the corpora size, which will be explained in the following sections.

[9] `https://sklearn-crfsuite.readthedocs.io/en/latest/index.html` (accessed: May, 2019)

Category	**Feature**	**Type**	**Example**
Capitalisation	CAP1: Is first letter capitalised?	Boolean	True/False
	CAP2: Is any character capitalised?	Boolean	True/False
	CAP3: Are all character capitalised?	Boolean	True/False
Contextual	CON1: Lowercase token	String	‘arts’
	CON2_i: ±3 tokens	String	‘des’
	CON3: Token length	Integer	4
Character	CHR3: Does it contain ′ symbol?	Boolean	True/False
	CHR11: Does it start with number?	Boolean	True/False
	CHR12: Does it start with punctuation?	Boolean	True/False
	CHR13: Is it a number?	Boolean	True/False
	CHR14: Is it a punctuation symbol?	Boolean	True/False
	CHR15: Does it contain a number?	Boolean	True/False
Lexical	LEX1: Is it in the dictionary of most frequent words of Basque?	Boolean	True/False
	LEX2: Is it in the dictionary of most frequent words of French?	Boolean	True/False
	LEX3: Is it a Name Entity?	Boolean	True/False
N-grams	CNG0_i: Binned probabilities gotten from Max-Entropy Classifier	Float	0.851

Table 5: List of the CRF features for LID task

To prepare our data for training and testing, we assumed we receive sentences as input and proceed to tokenise them. We used two strategies to tokenise, one that employed the generic tokeniser available in the NLTK python package[10], `TweetTokenizer`. The second one, using the tokeniser developed in (Agerri et al., 2014), `ixa-pipe-tok`, a tokeniser explicitly trained in Basque (Batua) sentences. We used the pre-trained tokeniser, Part-of-Speech (POS) tagger and Named-Entity Recognition (NER) tagger in their released `.jar` version (1.1.1).

For the NE feature, we compared two processes. The first one was a simple approach in which we check if any of the NEs in the given NE list was in the sentence. The second one used the NER tagger provided by *IXA-pipes*, `ixa-pipe-nerc`, again a tool explicitly trained in Basque. On account of Basque being an ergative-absolutive language, and its agglutinative morphology, we did a further process after identifying the NE. Only for NEs, we verified if the NE was in its declined form. To do so, we checked if the lemma given by `ixa-pipe-nerc` corresponded with the beginning of the NE. If it was the case, we created

[10] `https://www.nltk.org/index.html` (accessed: May, 2019)

Classifier	**Basque (Batua)**		**Basque (NL)**		**French**		**Code-Switched**		**binned**	**Max. length of**
	Train	**Test**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**	**probabilities?**	**French sentences**
1	1, 140	488	2, 577	1, 104	1, 140	488	222	95	Yes	10 tokens
2	1, 140	488	2, 577	1, 104	1, 140	488	222	95	No	10 tokens
3	1, 140	488	2, 577	1, 104	3, 500	1, 500	222	95	No	10 tokens
4	1, 140	488	2, 577	1, 104	3, 500	1, 500	222	95	No	Unlimited
5	1, 140	488	2, 577	1, 104	7, 000	3, 000	222	95	No	Unlimited
6	42	18	63	27	105	45	222	95	No	Unlimited

Table 6: Configuration of CRF classifiers

a new token with the suffix of the NE and updated the token of the NE with the lemma. This split is relevant since there are French NE that were declined and so, only the stem was French, while the suffix was Basque.

Since one of the features for the CRF implied to train a Maximum Entropy classifier to get the probabilities of a word belonging to Basque or French, we used it as the baseline for LID task. In the following section, we discuss the experiments carried out to train the Max-Entropy classifier. Once we got the best Max-Entropy classifiers, we tried using 10-bucket binned probabilities given by the classifier and a 2-size array containing the probabilities given by both classes (French and Basque).

Table 6 summarises the different configurations to train the CRF classifiers. We randomly selected the number of sentences of each language as specified by the configuration of the classifier. The French and Basque (Batua) sentences were tokenised using NLTK tokeniser, while the Basque (NL) monolingual and code-switched sentences were tokenised with IXA tokeniser.

3.1.3 Experiments: Maximum Entropy Classifiers

A Max-Entropy classifier is a probabilistic classifier that is based on the principle of maximum entropy (Jaynes, 1957). The classifier searches the probability distribution that best represents the data, that is, the distribution that has the largest entropy. During training, the algorithm estimates the weights (λ_i) of each feature $f_i(x, y)$ using the maximum likelihood estimation method. Where x are the observable features, and y is the class. Hence the probability given x to be classified as y is given by equation 18.

$$p(y|x) = \frac{exp(\sum_i \lambda_i f_i(x, y))}{\sum_y exp(\sum_i \lambda_i f_i(x, y))} \tag{18}$$

For training the Max-Entropy classifiers, we used the implementation available in the NLTK library, `MaxentClassifier`. The character n-grams ranging from 1 to n were used as features to train the classifiers. As we early mentioned, Zipf's law establishes a relation between the frequency of n-grams and their rank in a corpus. Table 7 illustrates the top 5 bigrams (word beginnings and endings) for French and Basque in Leipzig's corpora, as we can see, the lists are different. To test the impact of such a relation, we tried two different configurations for the n-grams, one with word-boundary tokens and the other without them. An example of features for the French word 'tout' (*all*) $n = 3$ are: $\{t, o, u, to, ou, ut, tou, out\}$ and $\{\langle w\rangle, t, o, u, \langle /w\rangle, \langle w\rangle t, to, ou, ut, t\langle /w\rangle, \langle w\rangle to, tou, out, ut\langle /w\rangle\}$ when using the word-boundary tokens.

Similar to the original work, we trained the Max-Entropy classifiers with different configurations for the languages. We tried including other languages aside from French and Basque, and compare them with those classifiers that only used French and Basque languages. We used Spanish tokens when training the French classifier and English tokens for the Basque classifier. When another language was set to true, we split the sets as 100% for the primary language, 75% for the second language and 25% for the other language. When it was set to false, we used 100% of the size of the training set for both languages,

Word beginnings		**Word endings**	
French	Basque	French	Basque
d'-	er-	-es	-en
l'-	ba-	-nt	-ko
co-	ko-	-er	-ak
Ma-	be-	-on	-an
pr-	es-	-re	-ik

Table 7: Top 5 of the most frequent bigrams for word beginnings and endings for both French and Basque (Batua)

French and Basque. We also varied the number of n-grams and the maximum number of iterations for training. Table 8 shows the different configurations tested. The combination of the various parameters resulted in 144 Max-Entropy classifiers 72 for French and 72 for Basque; the total number is reduced to 108 classifiers because French and Basque classifiers trained with only French and Basque languages are the same as the selection of the data set is deterministic. The training set size was 75% of the training size specified in the configuration, and the test set size was the 25% left. The dataset for the Basque language was composed of 75% words from the monolingual Navarro-Lapurdian corpus and 25% words from the Standard Basque corpus.

Parameter	**Values**
Size training set	$3,000$ and $6,000$
Another language	Yes, No
N-grams	3, 5 and 7
Maximum iterations	10, 20, and 30
start/end tokens	Yes, No

Table 8: Parameters for training the Max-Entropy classifiers

3.1.4 Metrics

In order to evaluate the performance of the different classifiers, we used the F1-measure. The Max-Entropy classifiers were ranked according to the F1-measure obtained for the French class. We defined one test set for comparing all classifiers, that consists of words different from the ones used during training. The test had a size of $9,000$ words half French and half Basque (keeping the relation between Navarro-Lapurdian and Standard Basque dialects used in training). All the words were extracted from the same corpora used during training.

The evaluation of the CRF classifiers was performed on the test set of each configuration. We obtained the confusion matrix for each classifier and the F1-measure weighted average. Nonetheless, considering we want our classifier to perform correctly for the code-switched sentences, we ranked the classifiers by their F1-measure of the French class in the code-switched test set.

3.2 G2P Models

Although an obvious strategy to do the phonetic transcription of the French words would be to use a monolingual G2P, we wanted to explore a multilingual G2P approach as such described in (Peters et al., 2017). In their approach, they trained a multilingual G2P Seq2Seq model in which each word was represented as a sequence of characters conditioned on the language token (see section 2). Using their strategy could bring the benefit of modelling two phonology systems with a unique model. That is a characteristic that can serve for overcoming mistakes that come from the LID task. Thus, we wanted to test the capability of the model to learn the relation between the orthography and the phonetic transcription, considering the differences in orthography for French and Basque.

Furthermore, in contrast with (Peters et al., 2017), we wanted to model the articulations, which are only present in phrases or sentences. By training the models with sentences instead of single words, we wanted the model to predict articulation phenomena such as *liaison* in French. In this section, we explain the approach employed for the G2P task, the corpora used and the metrics used to evaluate the performance of the different models.

3.2.1 Corpora

Likewise, as with the corpora for the LID task, we used two monolingual corpora (French and Navarro-Lapurdian Basque dialect) and the same code-switched corpus used for LID. In the case of the French corpus, we used the SIWIS[11] French Speech Synthesis Database (Honnet et al., 2017); and the Navarro-Lapurdian corpus for both the monolingual and code-switched corpora.

The SIWIS French Speech Synthesis Database is a corpus built specifically for speech synthesis. The corpus consists of six parts, table 9 describes the details of each part. The corpus is claimed to have phonetically balanced sentences. Although the corpus provides the phonetic transcriptions, they are encoded into the HTS[12] label format (Roekhaut et al., 2014). The HTS label format splits the sentence into phonemes and describes different features for each one such as the position of the phoneme in the syllable, the previous phoneme, among others. Notwithstanding, to create the corpus to train the G2P, we needed the phonetic transcriptions per each word, a more transparent and explicit format.

Moreover, to keep the consistency, we needed to follow the same phonological rules both in the monolingual French as for the French words in the code-switched corpus. Since

[11] Spoken Interaction with Interpretation in Switzerland

[12] Hidden Markov Model/Deep Neural Network-based Speech Synthesis System. http://hts.sp.nitech.ac.jp/ (accessed: May, 2019)

Part	No. Sentences	Description
parl	$4,500$	Parliament debates
book	$3,500$	French novels
siwis	75	SIWIS database
sus	100	Semantically unpredictable sentences
emph	$1,575$	Emphatic speech for sentences taken from the other parts
chap	-	A full book chapter

Table 9: The SIWIS French Speech Synthesis Database. Description of the corpus

we did not have available the HMM-based speech synthesis system, instead we used the phonetic transcriptions generated by the open source TTS eSpeak[13].

To build the French monolingual corpus, we took the book part of the SIWIS corpus and obtained their phonetic transcriptions with eSpeak. Although eSpeak uses SAMPA for the transcriptions, some phonemes did not use the standard. In which case we mapped the eSpeak phoneme with its SAMPA equivalent, for example, the unvoiced glottal fricative '_|' in eSpeak was replaced by 'h'. We also cleaned the transcription; that is, we removed symbols that are used for prosody; only the stress symbol was kept. Once we got the cleaned version of the sentences, we proceeded to verify the length of the sentences and their transcriptions. As we needed a one-to-one relation between the sentence and its phonetic transcription, we removed the sentences in which there was not such a relation, for example, sentences that had two punctuation symbols and the transcription was only one pause symbol. After that, we investigated that the sentences did not include phonemes outside of the French phoneme inventory [14]; we ignored the ones that included phonemes from other languages. Finally, to produce the source data to train the G2P model, we split each word into the characters and added the language token of each word, the words were separated using the bar symbol '|'; In the case of the punctuation symbols, we used the undefined language code. Table 10 shows an example of source and target sentences.

In the case of the Navarro-Lapurdian corpus, we obtained the phonetic transcriptions using modulo1y2 of the AhoTTS system for the Navarro-Lapurdian dialect. We used the `-TxtMode=Spell` feature that gives the transcription for each word; the system is rule-based for the Basque phonology and lexicon-based for the French words. The cleaning process was similar to the one described before. We separated the monolingual and code-switched sentences based on the corpus built for the LID task (see section 3.1).

As previously mentioned, we wanted to keep coherence concerning the rules employed to generate the French transcriptions. Hence, we go through all the sentences in the code-switched corpus (317 sentences) and check that the French transcription corresponded

[13] `http://espeak.sourceforge.net/`. Accessed on: May 2019. Version 1.48.03

[14] `https://www.phon.ucl.ac.uk/home/sampa/french.htm`. Accessed on: May 2019

with the one provided by eSpeak. Table 10 shows examples of transcriptions for both monolingual and code-switched sentences.

Language	**Type**	**Example**
French	Source	fra L a | fra r é c o l t e | fra f u t | fra f a c i l e | und .
	Target	l a | R e k O l t | f y | f a s i l | _
Navarro-Lapurdian monolingual	Source	bqe E t a | bqe d e u s i k | bqe e z t u e n a k | und ?
	Target	e t a | D e w s‘ i k | e s t w e n a k | _
Navarro-Lapurdian code-switched	Source	fra L a f a r g u e | bqe d a | bqe h e m e n | und .
	Target	l a f a R g | D a | h e m e n | _

Table 10: Example of source and target sentences for G2P model. In these examples the transcription did not include the stress marker

We developed two versions of the corpora, one in which the target files (the phonetic transcriptions) have the stress symbol, and the other without it. Table 11 summarises the resulted corpora used for training the G2P models.

Corpus	**No. Sentences**
French (SIWIS-book part)	$2,873$
Navarro-Lapurdian (monolingual)	$3,592$
Navarro-Lapurdian (code-switched)	317

Table 11: Summary of G2P corpora. Each corpus has two versions of phonetic transcriptions, one with the stress marker and the other without it

3.2.2 Experiments

In (Peters et al., 2017) the Seq2Seq model used was the encoder-decoder model with attention mechanism described in (Bahdanau et al., 2014) (please see section 2). They used OpenNMT (an NMT toolkit (Klein et al., 2018)) to implement their model. As explained in their work, G2P can be modelled as an NMT problem, and so we proposed to test their input model using other neural networks models applied to NMT. We tested the Transformer model (Vaswani et al., 2017), an architecture widely used for NMT and whose computational complexity is better than other architectures also robust like Convolutional Neural Networks (CNN). Provided the advantage that the Transformer model is also implemented in the OpenNMT toolkit.

In contrast with the encoder-decoder model architecture, in the Transformer model architecture, there are no LSTM units but stacked layers each one with the multi-head self-attention mechanism, which corresponds to one of the novelties of this model. Figure 3 illustrates the architecture of the Transformer model, as presented in (Vaswani et al., 2017).

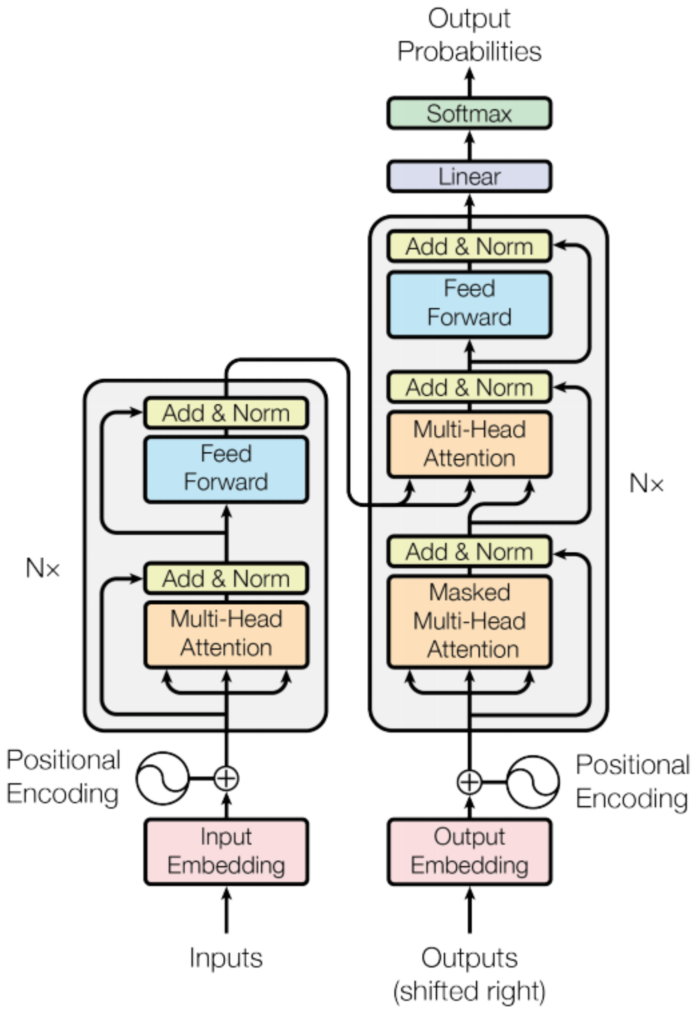


Figure 3: Transformer model architecture

The Transformer model architecture is based on attention mechanisms. In the architecture, there are N stacked encoders, and N stacked decoders. Each encoder has a layer of self-attention mechanism followed by a position-wise feed-forward network; the decoder is alike, but between the self-attention layer and the feed-forward network there is an encoder-decoder attention layer, similar to the attention mechanism used in Seq2Seq models.

The self-attention layer allows the model to take context into account when encoding or decoding a specific word, in the context of NMT, this is helpful for anaphora resolution.

The equation 19 describes how the attention is calculated; The matrices Q, K and V (Queries, Keys and Values respectively) are weight matrices learnt during training; d_k corresponds to the dimension of the queries and keys. This scaled dot-product attention models the importance of words while encoding another word (self-attention of the word being encoded).

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{19}$$

Furthermore, the attention mechanism in the Transformer model uses what they called Multi-Head Attention. This mechanism enables the model to have several representations of the relations of the words by using several heads. Each head learns different Q, K and V matrices modelling different features of the relations. Hence, for each self-attention layer, there are h parallel attention layers once the layers have been processed; they are concatenated before being sent to the feed-forward network. Equations 20 and 21 shows how the multi-head attention is calculated, where $W^O \in \mathbb{R}^{hd_v \times d_{model}}$, d_v is the dimension of the values and d_{model} is the dimension of the model.

$$\begin{aligned} MultiHead(Q, K, V) &= Concat(head_1, \cdots, head_h)W^O & (20) \\ head_i &= Attention(QW_i^Q, KW_i^K, VW_i^V) & (21) \end{aligned}$$

In our experiments, we used the `TransformerBig` model available in OpenNMT. The details of the model parameters are listed in table 12 used for all the experiments. The FFN Inner dimension corresponds to the dimension of weight matrix learn in the feed-forward network. The ReLU dropout corresponds to the dropout for the ReLU activation in the feed-forward network.

Parameter	**Value**
d_{model}	$1,024$
No. encoders/ decoders	6
No. heads	16
FFN Inner dimension	$4,096$
Dropout	0.3
Attention dropout	0.1
ReLU dropout	0.1
Batch size	$4,096$

Table 12: Parameters of Big Transformer

In our experiments, we tested the ability of the models to learn both phonologies and to adapt them whenever needed based on the surrounding words. We tried three different configurations of the dataset: all monolingual sentences; all code-switched sentences; and both monolingual and code-switched sentences. For each configuration, we test the capability of the model to learn the stress pattern of each language. To do so, we prepared two versions of the dataset, one including the stress marker ‘’’ and the other without it. For the three dataset configurations, we split the dataset into 80% for the training set, 10% for the validation set and 10% for the test set.

During earlier experimentation, we found that the number of the maximum sequence length either source or target was a parameter that affects the training model. Because of that, we decided to restrict our models to be trained using a maximum sequence length for both source and target of 100 items (characters, and phonemes). There is often the case that the speaker will make a pause when reading a punctuation symbol. Assuming coarticulation phenomena do not occur around punctuation symbols. Whenever possible, we cut the longest sentences around punctuation symbols; avoiding cutting a coarticulation phenomenon. However, that is not always possible, as there may be sentences in which case the punctuation symbol is outside of the 100 window, in such cases we cut the sentence around the last word in the window, avoiding cutting a sentence in the middle of a word. Table 13 summaries the different dataset configurations.

3.2.3 Metrics

G2P models are often evaluated using Word Error Rate (WER) and Phoneme Error Rate (PER) (Bisani and Ney (2008); Yao and Zweig (2015); Deri and Knight (2016); Kyaw Thu et al. (2016); Toshniwal and Livescu (2016); Peters et al. (2017)).

PER is the Levenshtein's distance between the predicted phoneme sequence and the target sequence, the actual sequence. In our case, besides the phonemes themselves, we included the word separator symbol ‘|’ in the calculation.

On the other hand, WER is the rate of the words which phoneme sequence is not exactly the actual phoneme sequence. We concatenated the phoneme sequence to form the word and used the Levenshtein's distance to find the wrong guesses. There may be cases in which a word separation symbol is added or removed it in the prediction, and so it may affect the metric if we do a one-to-one check.

3.3 TTS Integration

The current system is a modular system composed of three main modules: *Modulo1* which performs the preprocessing and normalisation tasks; *Modulo2* which obtains the linguistic features; and *Modulo3* which gets the acoustic features and performs the synthesis. Figure 4 shows a general schema of the modular architecture, *Modulo1* and *Modulo2* are often combined into a big module (*Modulo1y2*) for the final system.

In order to include the LID and G2P tasks, we had to modify the data flow in the *Modulo1y2* module. Figure 5 illustrates the data flow and process of the linguistic module. The module reads the input through a stream and at each time step performs the segmentation, the normalisation and the verbalisation of each word; this means that the

Configuration	Training set				Validation set				Testing set			
	Sent.	FRA	BQE	UND	Sent.	FRA	BQE	UND	Sent.	FRA	BQE	UND
Monolingual	7,839	34,762	26,814	10,790	980	4,102	3,569	1,386	980	4,397	3,232	1,350
Code-switched	453	464	2,925	658	57	54	408	89	57	69	361	89
Monolingual + Code-switched	8,292	35,226	29,739	11,448	1,037	4,156	3,977	1,475	1,037	4,466	3,593	1,439

Table 13: Statistics of the dataset for G2P models. The figures correspond for both the dataset with stress marker and without. Sent.: Number of sentences; FRA: Number of French words; BQE: Number of Basque words and UND: Number of punctuation symbols

text is not read all at once but per chunks. Once the normalisation and verbalisation are finished, the module sends the object representing the utterance (`UttWS`) to the package that extracts the linguistic features. The package extracts the POS tags, places the pauses where needed, obtains the phonetic transcription (using the lexicon or rules if the word is not in the dictionary), and predicts the prosody from the text. The dictionary is consulted at both phases. Finally, the module generates the labels required for the synthesiser.

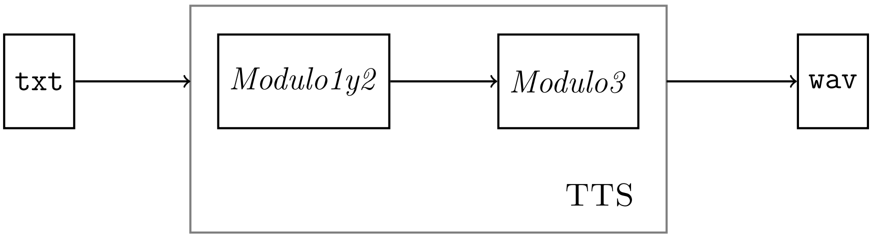


Figure 4: Modular structure of the current TTS

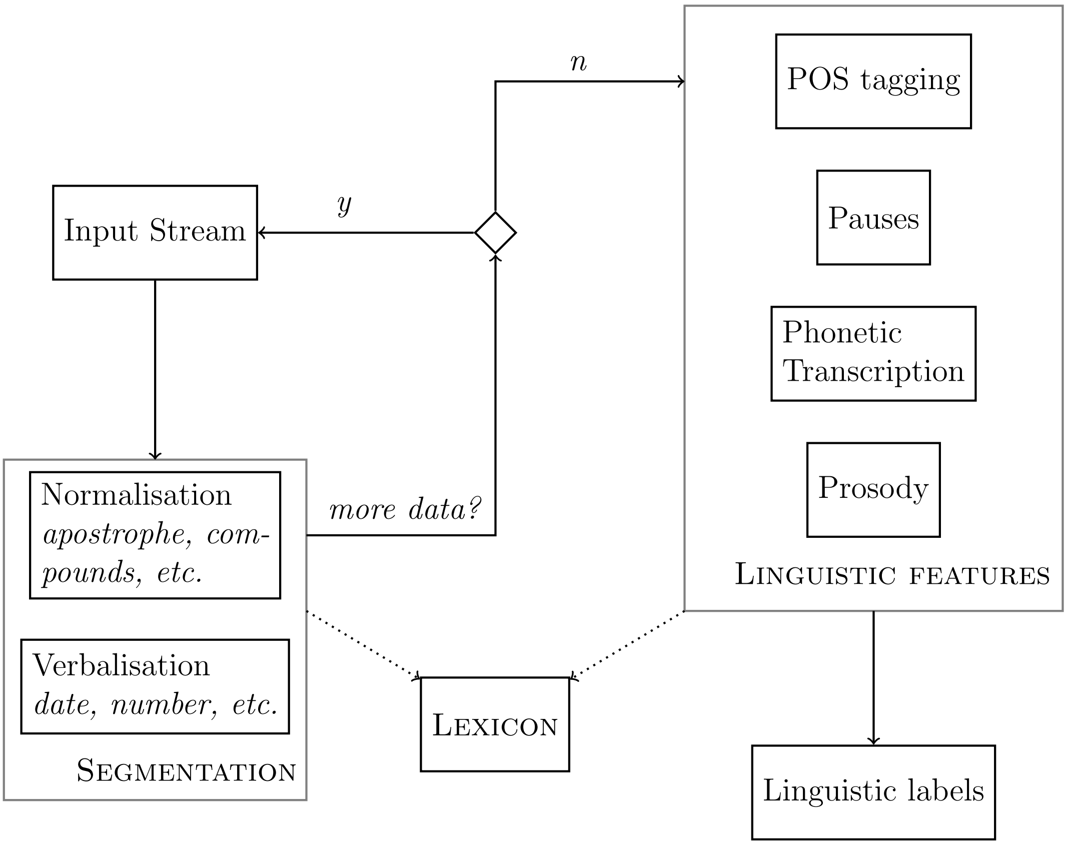


Figure 5: Data flow of the linguistic module

We introduced the LID task in the *Segmentation* block. The package first performs the LID task before the normalisation task; thus, it avoids changing the written word with the

Basque morphophonological rules when it is a French word. However, there are two steps in the normalisation task that are required even for French words. Specifically, we want to use the phonetic transcription of the dictionary if the word is in the lexicon. Also, we want to do compound normalisation. This step is useful to keep the language information about the morphemes that compose the word. For example *Andrée-rekin*, we want to keep the information that *Andrée* is French and that *rekin* is Basque, so later we avoid further normalisation steps for *Andrée* but not for *rekin*.

On the other hand, the G2P task is placed in the *Linguistic Features* block. We added a new tag (`POS_FR_TF_MRK`) to state if a word is a French OOV word and to indicate that later we need to obtain the phonetic transcription with the G2P. The tag is assigned during the POS tagging step. If the word has been marked, the package calls the G2P model to obtain the phonetic transcription and maps the output characters with the phonemes in the system inventory.

The identification of the language is performed at the word level. Once the language code of the word is obtained, it is stored as a feature of the word object. This feature is passed from the object mapping the input streams to the object that represents the word in the *Linguistic Features* block. Figure 6 depicts the new tasks.

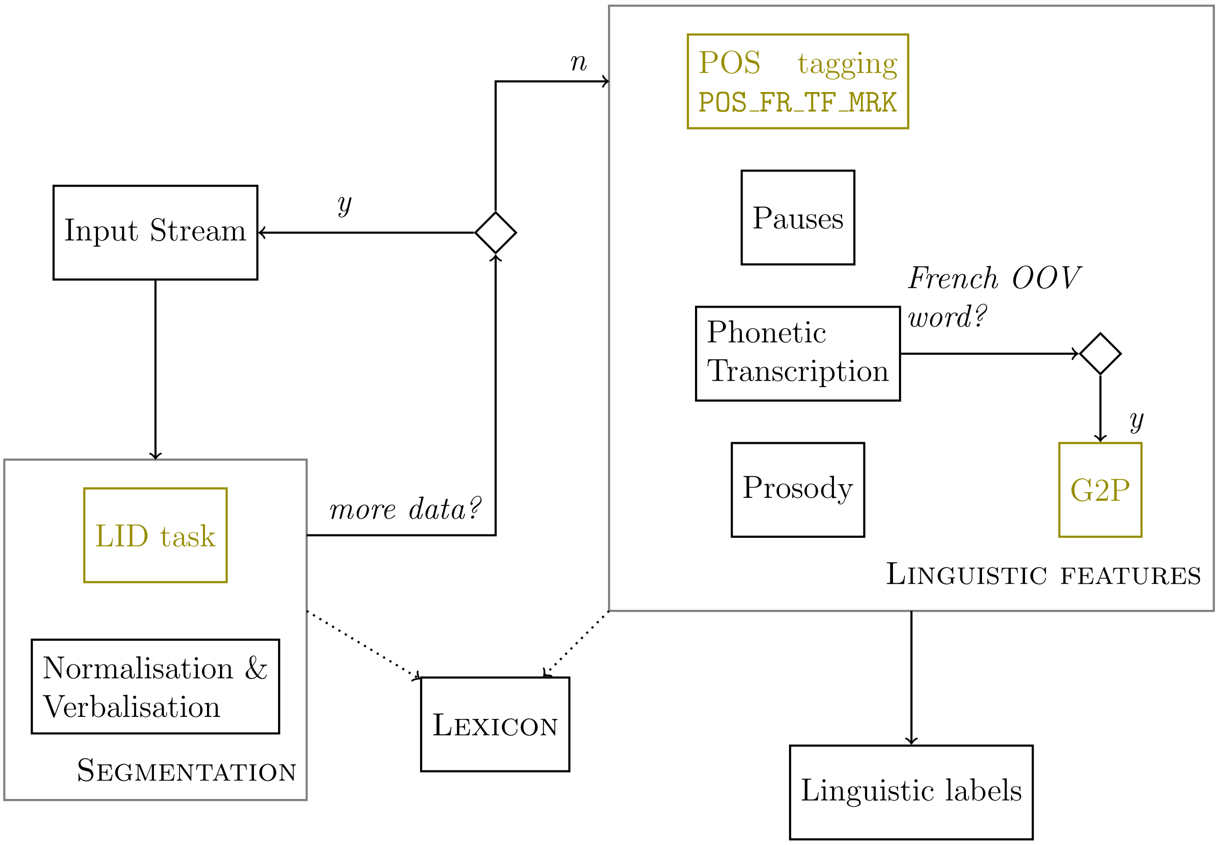


Figure 6: New tasks in the linguistic module

3.3.1 Technologies

As the technologies used for developing the TTS system and the LID and G2P models were all different, we had to use Application Programming Interfaces (APIs) to communicate them. The TTS system is developed using `C++` programming language, whereas the LID and G2P models are developed with `Python`. Additionally, the G2P model uses `TensoFlow` library.

To communicate the `Python` applications we used `Python/C API`[15]. This API allows embedding `Python` scripts into `C++` applications. We wrapped the trained model into a manager script that gets as input the words sent from the `C++` code, then calls the model and sends back the language code or transcription depending on the case.

The LID trained models: N-gram and CRF models can be accessed using `Python`, by loading the models every time the manager script is called. The communication protocol for the G2P is more complicated as we need to consume the trained model. We used the `onmt-main infer`[16] recipe available in the `OpenNMT-tf` library. Our G2P trained model is accessible with a shell call. We send the word and get the phonetic transcription. Figure 7 shows the interaction of the different protocols.

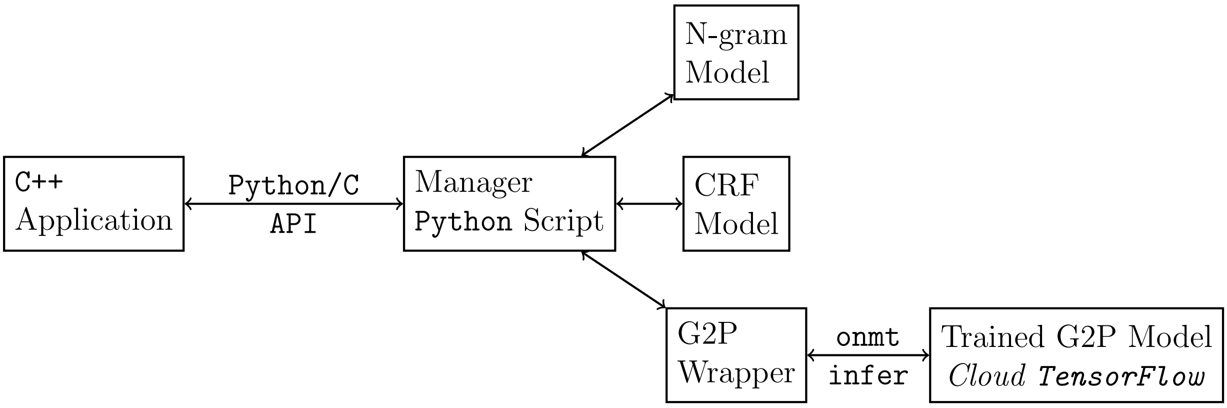


Figure 7: Communication protocols between the TTS system and the trained models

[15] `https://docs.python.org/2/c-api/index.html` (accessed: May, 2019)

[16] `http://opennmt.net/OpenNMT-tf/inference.html` (accessed: May, 2019)

4 Findings

In this section, we analyse the results obtained from the different experiments described in 3. In the first two subsections examine the performances for the LID and G2P tasks individually. In the last part of this section, we consider a small experiment combining the two tasks.

4.1 LID Task Results

We first trained the 108 Max-Entropy classifiers to select the best one as the baseline model and to use it in the CRF training. The methodology employed in (Chittaranjan et al., 2014) proposed to use the binned probabilities of the Max-Entropy classifiers of the language pair under study. However, it was not obvious that in the experiments there will be two best classifiers, one for French and another for Basque (both different one to the other). As a matter of fact, that is not the case as we will see in the results. In section 3.1, we explained that 36 classifiers were the same for French and Basque since they were trained with the same data set (there was no other language included). Thus, our best classifier is at the same time the best classifier for French and Basque language. This means that for the CRF training, we used the binned probabilities of one classifier instead of two classifiers as in the original paper.

Table 14 shows the best ten classifiers ranked by their result for the F1-score for the French class, the class of our interest. Most of the best classifiers did not use another language for training. The overall F1-score was calculated using the weighted average, that denotes keeping the proportions of the class distribution to calculate the measure.

Language	**Other language**	**N-gram**	**Max. Iterations**	**F1-score**	**F1-score bqe**	**F1-score fra**
Both	No	5	30	0.8953	0.8957	0.8949
Both	No	5	20	0.8946	0.8949	0.8942
Both	No	7	10	0.8939	0.8942	0.8936
Both	No	5	10	0.8933	0.8934	0.8933
Both	No	7	20	0.8928	0.8931	0.8925
Basque	Yes	5	20	0.8931	0.8939	0.8923
Both	No	7	30	0.8918	0.8922	0.8913
Basque	Yes	5	10	0.8916	0.8921	0.8910
Basque	Yes	5	30	0.8913	0.8922	0.8905
Both	No	3	10	0.8884	0.8880	0.8889

Table 14: Best 10th Max-Entropy classifiers ordered by their F1-score for the French class. All of the classifiers have a training set size of $6,000$ and employed word-boundary tokens

Although the ranking allows selecting the best classifier, it does not show the evolution of the metrics depending on the different parameters of configuration. Figures 8 and 9 show the performance in terms of F1-score (weighted average) of the different configuration. The green colour represents the classifiers trained with a training set size of 3, 000 words; the purple one represents the training set size of 6, 000. The dashed lines represent the classifiers trained with other languages (either English for Basque classifiers, or Spanish for French classifiers); the dotted ones represent the classifiers trained only with French and Basque words. The big dot represents the classifiers that used word-boundary tokens, and the big star represents the classifiers that do not use word-boundary tokens. In the x-axis, we compare the impact of the `n` value for the character N-grams, as well as the number of maximum iterations, `m`. The values go from the lowest to the highest order first by the `n` value and then the `m` value.

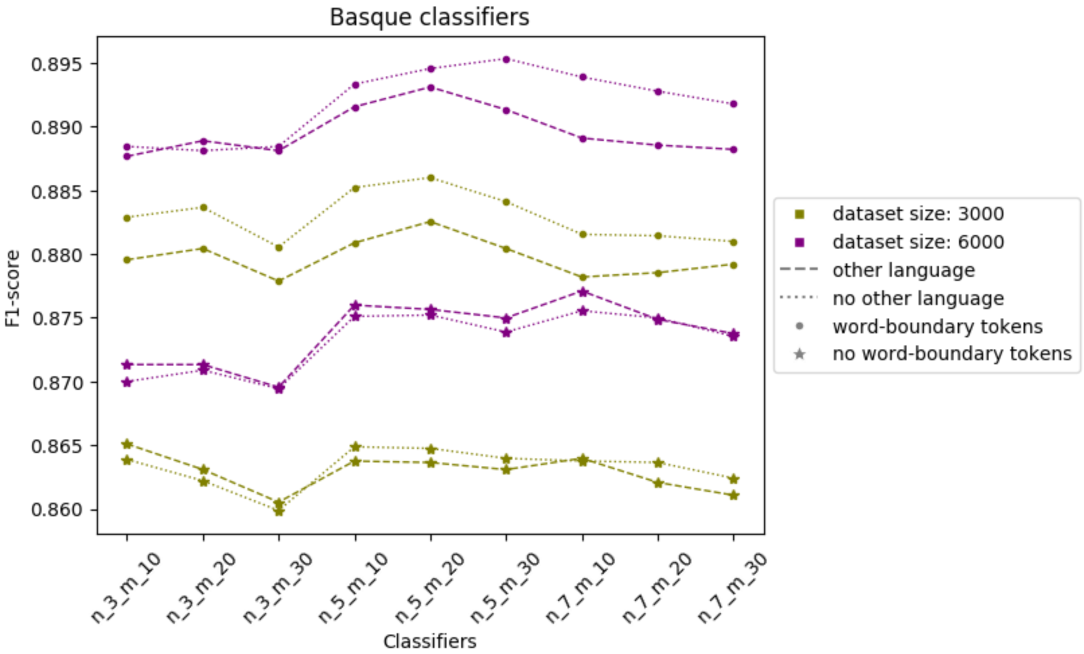


Figure 8: Comparison of Basque Max-Entropy classifiers. Where $\bar{x} \in [0.8629, 0.8919]$ and $\sigma \in [0.0014, 0.0027]$ per series

In the figures, we can see that for the two languages, for the same configurations, the classifiers showed a better performance when they were trained with more data. The positive impact of increasing the dataset size is common in the field of machine learning for several domains and models. Nevertheless, if we look at the differences between the datasets, the improvement is about 0.01 with the double of data. This result can be supported by Zipf's law discussed in section 3.1.3. Under the hypothesis that the character N-grams follow the Zipf's law, with few data, we can cover the vast majority of the most common N-grams.

Also, in general, using word-boundary tokens improved the results in comparison with-

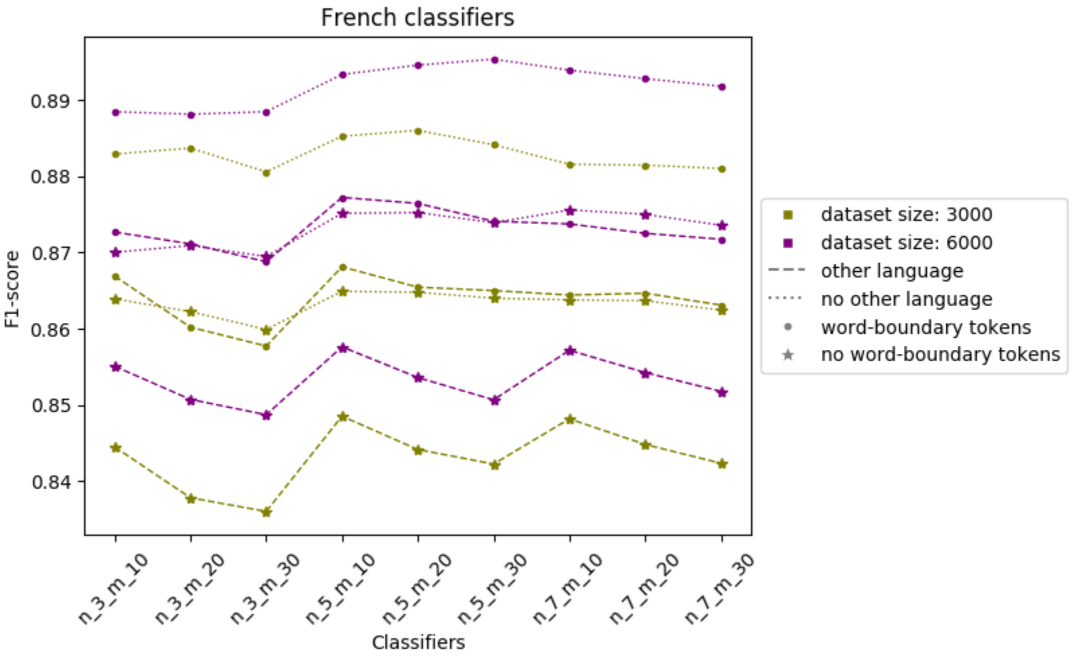


Figure 9: Comparison of French Max-Entropy classifiers. Where $\bar{x} \in [0.8432, 0.8919]$ and $\sigma \in [0.0015, 0.0039]$ per series

out using them. This result is also related to Zipf's law. As we saw in table 7, the word beginnings and word endings most common in French were different from the ones in Basque. By using the word-boundary tokens, we included extra information about the distribution of the character N-grams. In the table, we illustrated this with the word beginning 'es-' common in Basque and the word ending '-es' common in French. Without the word-boundary token, the character N-gram 'es' could be considered frequent for both languages.

For the Basque case, for the same configuration, the difference between using another language or using only Basque and French words is small. We see in figure 8 that these two parameters group the classifiers. On the other hand, this relation is not exhibited for the French classifiers. The French classifiers show a tendency where using another language affects performance. The difference in the patterns can be explained by the distance between the languages used for training the classifiers. In (Deri and Knight, 2016) they calculated the distance of several languages pairs based on the phonetic inventory, the grapheme system, the geological location, among other features[17]. The interpretation of the distance measure is the lower the value the closest the languages are. The distance between French and Spanish is 0.372, while for Basque and English the distance is 0.580. It is probable that the French classifier trained with Spanish as belonging to the other language misclassifies those words as French words, favouring the pattern found in the

[17] The complete list of the calculation is available on `https://drive.google.com/drive/u/0/folders/0B7R_gATfZJ2aWkpSWHpXUklWUmM` (Accessed: April 2019)

figure.

Contrary to the hypothesis, the results show no clear correlation between increasing the number of characters in the N-grams and the performance. We see a modest improvement from 3-grams to 5-grams for both languages, though more evident for the Basque classifiers. On the other hand, the tendency is not kept from 5-grams to 7-grams. A similar pattern is shown concerning the number of maximum iterations; where the increment of the number of iterations does not show a definite improvement. Notably, the standard deviation for the Basque classifiers goes from ±0.001 to ±0.003, excluding the classifiers trained with 3-grams, comparing classifiers with the same configuration (series in the figure 8). Moreover, for the French classifiers it goes from ±0.001 to ±0.004. The highest variations were found for the classifiers trained with Spanish and without word-boundary, see the lower two series in figure 9.

Besides the grid search to find the best Max-Entropy classifier, we also validated the best model to evaluate if there was a dependency on data. We used the K-fold cross-validation with randomisation of the dataset since the size of the whole corpora were more significant than the size of the best configuration ($6,000$ words). We first selected $6,000$ random words from the corpora and then split the dataset into train and test set, ten folds were used. The following table shows the measures obtained for each fold. The performance of the Max-Entropy classifier shows a modest variation as the dataset is changed, an F1-score (weighted average) of 0.8847 ± 0.004 with a confidence interval of 95%.

Fold	**F1-score**	**F1-score bqe**	**F1-score fra**
1	0.8713	0.8710	0.8717
2	0.8797	0.8788	0.8805
3	0.8927	0.8930	0.8923
4	0.8900	0.8895	0.8905
5	0.8850	0.8830	0.8869
6	0.8840	0.8843	0.8837
7	0.8843	0.8852	0.8834
8	0.8863	0.8862	0.8864
9	0.8867	0.8870	0.8863
10	0.8867	0.8867	0.8866
$\bar{x}$	0.8847	0.8845	0.8848
$z\frac{\sigma}{\sqrt{n}}$	0.004	0.004	0.004

Table 15: Cross-validation for the best Max-Entropy classifier. Configuration of the classifier: 5-gram, 30 maximum iterations, word-boundary tokens, and trained only with French and Basque words. Confidence interval of 95%

For the CRF classifiers, we first trained the six different configurations (see table 6) with the default parameters of the sklearn-crfsuite library: gradient descent algorithm using the L-BFGS[18] method; L1 regularisation coefficient equals 0, and L2 regularisation coefficient equals 1. We also set the values for the cut-off threshold for the minimum frequency to 1, that is, the CRF classifiers ignore features that do not occur more than the minimum frequency in the training set; and the maximum number of iterations to 500. Table 16 shows the performance of the first classifiers including the baseline (Max-Entropy classifier). We evaluated the classifiers with the code-switched test set, a set of 95 code-switched sentences.

For the undefined class (und) we used a list that includes the punctuation symbols found in the corpora for the case of the Max-Entropy classifiers, also whenever the probability of being French was exactly 0.5, the classifier outputs und. We do not report the results for the undefined class because in all of the classifiers, the accuracy for this class was 100%. Therefore we focus our analysis in the French and Basque classes.

Classifier	**Tok.**	**Acc.**	**F1**	**F1 bqe**	**F1 fra**	**Re. bqe**	**Re. fra**	**Pre. bqe**	**Pre. fra**
CRF 1	ixa	0.895	0.879	0.930	0.427	0.973	0.314	0.891	0.667
CRF 1	nltk	0.886	0.852	0.925	0.260	**0.991**	0.157	0.867	**0.750**
CRF 2	ixa	0.929	0.922	0.952	0.651	0.980	0.538	0.925	0.825
CRF 2	nltk	0.900	0.875	0.934	0.393	**0.994**	0.252	0.880	**0.883**
CRF 3	ixa	0.927	0.922	0.951	0.657	0.974	0.562	0.928	0.792
CRF 3	nltk	0.904	0.886	0.936	0.460	**0.988**	0.319	0.889	**0.827**
CRF 4	ixa	0.922	0.916	0.947	0.627	0.973	0.529	0.923	0.771
CRF 4	nltk	0.906	0.889	0.937	0.476	**0.988**	0.333	0.891	**0.833**
CRF 5	ixa	0.917	0.910	0.943	0.605	0.968	0.514	0.920	0.735
CRF 5	nltk	0.910	0.898	0.939	0.539	**0.980**	0.410	0.902	**0.789**
CRF 6	ixa	0.948	0.948	0.964	0.787	0.966	0.776	0.961	0.799
CRF 6	nltk	0.943	0.939	0.961	0.742	**0.984**	0.643	0.938	**0.877**
Max-Entropy	ixa	0.920	0.922	0.943	0.702	0.929	0.762	0.958	0.650
Max-Entropy	nltk	0.915	0.918	0.939	0.692	0.927	0.743	0.952	0.647

Table 16: Comparison of all LID classifiers using the code-switched test set (95 sentences). Tok.: Tokenisation, Acc.: Accuracy, F1: F1-measure average weighted, Re.: Recall and Pre.: Precision. In blue the best results. See the configuration of the different CRF classifiers in table 6

The second experiments involved hyperparameter tuning for the CRF classifiers. As early mentioned, the first experiments kept the default parameter values for training the

[18] L-BFGS: Limited-memory Broyden-Fletcher-Goldfarb-Shanno (an optimisation algorithm)

classifiers. However, this can lead to erroneous conclusions if the performance is correlated to these parameters and not to the configuration (dataset, tokenisation, and Max-Entropy probabilities). We used the fit and score method for the tuning using 5-fold cross-validation over the training dataset specified for each configuration. We worked with `RandomizedSearchCV` method available in the scikit-learn library. To evaluate the classifiers, we used the F1-measure for the French class. Table 17 shows the results obtained after the hyperparameter tuning.

Classifier	**Tok.**	**Acc.**	**F1**	**F1 bqe**	**F1 fra**	**Re. bqe**	**Re. fra**	**Pre. bqe**	**Pre. fra**
CRF 1	ixa	0.911	0.901	0.940	0.545	0.975	0.429	0.908	0.750
CRF 1	nltk	0.907	0.889	0.938	0.474	**0.990**	0.329	0.891	**0.852**
CRF 2	ixa	0.935	0.930	0.956	0.688	0.983	0.576	0.931	0.852
CRF 2	nltk	0.921	0.907	0.947	0.570	**0.995**	0.410	0.903	**0.935**
CRF 3	ixa	0.936	0.931	0.956	0.695	0.981	0.590	0.933	0.844
CRF 3	nltk	0.923	0.913	0.948	0.604	**0.990**	0.457	0.910	**0.889**
CRF 4	ixa	0.929	0.923	0.952	0.657	0.978	0.552	0.927	0.811
CRF 4	nltk	0.926	0.919	0.950	0.641	**0.984**	0.514	0.918	**0.850**
CRF 5	ixa	0.927	0.922	0.951	0.657	0.974	0.562	0.928	0.792
CRF 5	nltk	0.923	0.916	0.947	0.636	**0.976**	0.529	0.920	**0.799**
CRF 6	ixa	0.959	0.958	0.971	0.828	0.976	0.805	0.967	0.854
CRF 6	nltk	0.956	0.954	0.969	0.808	**0.985**	0.733	0.953	**0.901**
Max-Entropy	ixa	0.920	0.922	0.943	0.702	0.929	0.762	0.958	0.650
Max-Entropy	nltk	0.915	0.918	0.939	0.692	0.927	0.743	0.952	0.647

Table 17: Comparison of all LID classifiers after hyperparameter tuning. Tok.: Tokenisation, Acc.: Accuracy, F1: F1-measure average weighted, Re.: Recall and Pre.: Precision. In blue the best results

In general terms, the patterns found with the default parameters were still valid in the results of the tuned classifiers. In 97.92% of the metrics, the tuned version of the classifiers performed better. The improvement of the F1-measure for the case of NLTK tokenisation was on average 0.03 points for all the classes and 0.01 for IXA tokenisation.

We can see from the tables 16 and 17, that for the case of the NLTK tokenisation for CRF classifiers, the recall of Basque words and the Precision of French words were slightly better than the IXA tokenisation. The two tokenisation contrast mainly in the identification of suffixes. NLTK tokenisation will separate a stem from the suffix only if a hyphen already separated these; otherwise, it will be considered as a single word. In the code-switched corpus, many of the French tokens were in the declined form without the hyphen, and since they were annotated as French, this could have confused the classifiers.

We can interpret the results as many of the declined French words were classified as Basque by reason of the suffix, and words classified as French were mostly not declined words. Thus, this produces a better recall of Basque words and better precision of French words taking into account the number of recalled French words.

On the other hand, the IXA tokenisation models better the nature of the corpus, in which the majority of the French tokens were NE. By using a tokeniser that accounts for the linguistic systems of the language, it provides an appropriate data-approach. Figures 10 and 11 show the performance of the classifiers with respect to the F1-measure for the French class. We can see the importance of the tokenisation when the parameters of the classifiers are not set to the best parameters. The separation between IXA and NLTK tokenisation were more drastic when the parameters were set to default values than when the parameters were tuned.

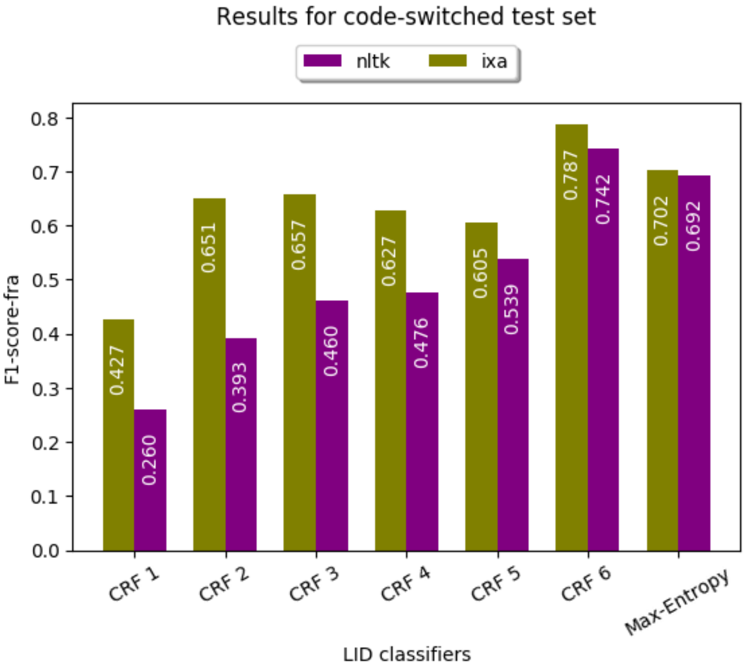


Figure 10: Comparison all LID classifiers with respect to the F1-measure for the French class with the default parameters

Configurations one and two are the same except for the $CNG0_i$ features, the probabilities of the Max-Entropy classifier. In configuration one, we used binned probabilities, while in configuration two, we used the probabilities directly. The effect of these changes is not notable when we compare their results for all the classes, around 0.02 of improvement using the probabilities directly. Nevertheless, when we compare them only for the French class, we have an improvement of 0.081 for the NLTK tokenisation and 0.147 for the IXA tokenisation (see the F1-measure for French class in table 17). There is no clear insight into why the performance is improved by using direct probabilities. Nonetheless, remember that the binned probabilities were proposed in the original work (see section 3.1) as

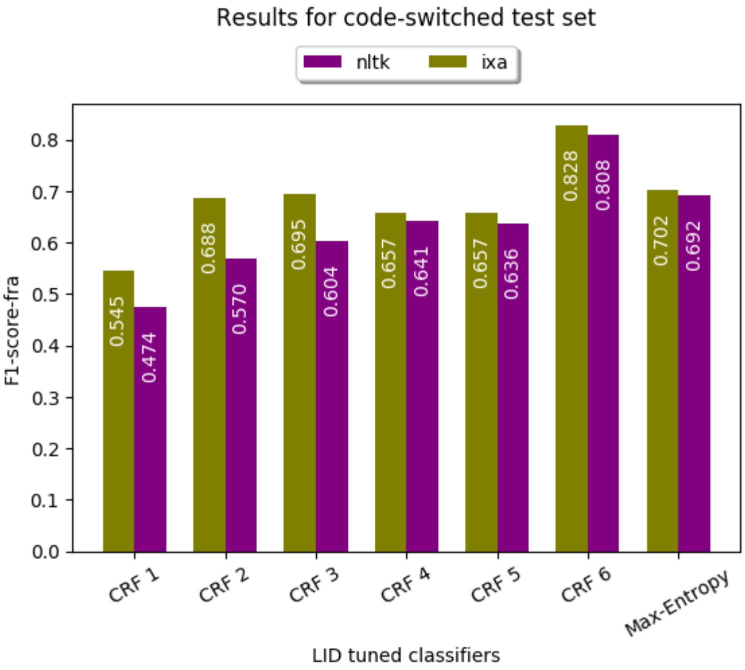


Figure 11: Comparison all LID classifiers after hyperparameter tuning with respect to the F1-measure for the French class

a combination of the probabilities given by two Max-Entropy classifiers, which is not the case in our experiments, in which we only have one Max-Entropy classifier. In that sense, the direct probabilities provide more concise information than the binned probabilities.

For the hyperparameter tuning, we iterated 50 times, that means that 50 different parameters where tested. For each pair of parameters, we run 5-fold cross-validation resulting in 250 different CRF classifiers for each configuration. Each fold split the training set specified in the configuration (see table 6) into a new training and test set, the evaluation of the classifier was based on the F1-measure for the French class. The standard deviations of the performance for each configuration were: 0.0036 (CRF 1), 0.0030 (CRF 2), 0.0011 (CRF 3), 0.0010 (CRF 4), 0.0005 (CRF 5) and 0.0018 (CRF 6). That shows that there is no significant effect on the chosen data when the classifiers are tested in the same kind of data. However, as figure 11 shows, when the data changes, the classifiers respond differently.

Classifier CRF 6 outperformed the rest classifiers when tested on the code-switched test set because the majority of the samples used for its training came from the code-switched corpus. The effect of the training data can be seen if we compare classifiers CRF 3 and CRF 4. The two classifiers only differ from the maximum length that the French sentences have. CRF 3 was trained using a maximum length of 10 words while there was no limit for CRF 4. We can see that even though they were training using the same amount of data, classifier CRF 3 has a better performance for the French class. In the code-switched corpus, the longest French sentence in one switch is 5-word length. We see that increasing the number of monolingual French sentences (CRF 5) did not improve the results for the code-switching cases. The results for the code-switching sentences show a significant dependence between the training data and the data on which the classifier will be evaluated.

Taking into account the type of code-switching that the Navarro-Lapurdian written corpus shows, the best classifiers correspond to the baseline, the Max-Entropy classifier and the CRF 6 classifier. Considering the previous TTS system was assuming all the words were Basque, the baseline is an improvement of 0.702 points in the F1-measure for the French class. An improvement relying only on the character n-grams of the word. On the other hand, the CRF 6 classifier shows that including information from the context helped to have a better result, an improvement of 0.126 points (see figure 11, IXA tokenisation) with respect to the Max-Entropy classifier. Table 18 shows the confusion matrix of each classifier.

		Predicted	
		French	**Basque**
Actual	**French**	160	50
	Basque	86	1,128

(a) Confusion matrix of Max-Entropy classifier

		Predicted	
		French	**Basque**
Actual	**French**	169	41
	Basque	29	1,185

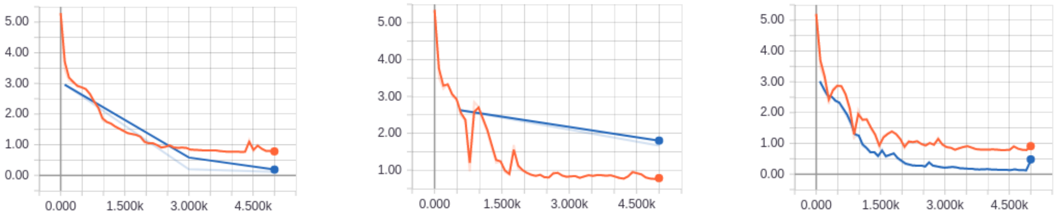
(b) Confusion matrix of CRF 6 classifier

Table 18: Confusion matrix of best LID classifiers using the code-switched test set

However, it is not surprising that CRF 6 classifier is the best classifier for the code-switched corpus, as we said earlier, this classifier was mainly trained in this kind of sentences. Although the dataset of this classifier corresponds to the smallest one in all of the configurations (see table 6), we examined its performance in more extensive monolingual sentences. We evaluated it using the test set of the configuration 5 (488 sentences from Standard Basque, 1, 104 sentences from the Navarro-Lapurdian dialect, 3, 000 sentences from French and 95 sentences from the code-switched corpus). It obtained an F1-measure of 0.989 for the French class (support of 58, 281 words) and 0.965 for the Basque class (support of 13, 461 words). CRF 6 is not the best classifier for that test set though; there was a drop of 0.007 point with respect to the F1-measure for the French class in the best performance. Still, the points dropped in the monolingual sentences could be acceptable considering the improvement gained on the data of our interest, the code-switched corpus.

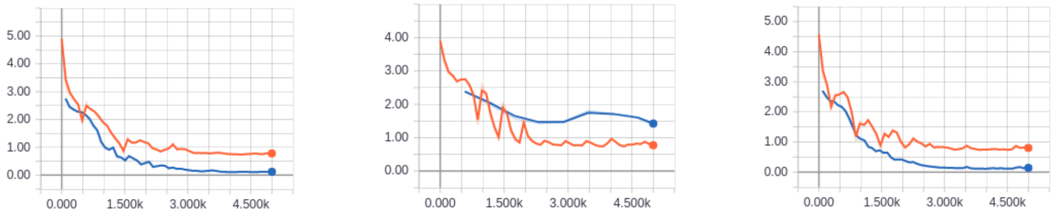
4.2 G2P Task Results

We run the training for all the configurations described in 3.2. Six configurations in total, three considering the stress marker and three without it. For the settings with a large number of sentences, the training time was about 10 hours per experiment, using a GPU of 12 Gb; however, all of the experiments were run for 5, 000 training steps, also known as epochs. Figures 12 show the loss curves for the configurations with the stress marker. Figures 13 show the loss curves for the configurations without the stress marker. The blue curve corresponds to the validation loss, and the orange curve corresponds to the training loss.



(a) Monolingual dataset (b) Code-switched dataset (c) Monolingual + Code-switched dataset

Figure 12: Training and validation loss curve. Configurations with the stress marker



(a) Monolingual dataset (b) Code-switched dataset (c) Monolingual + Code-switched dataset

Figure 13: Training and validation loss curve. Configurations without the stress marker

From the loss curves, we can already detect some problems in the configurations trained only with the code-switched corpus. We see that the validation loss curve does not decrease at the same pace as the training loss. This behaviour may be caused by the small size of the code-switched corpus; there are only 453 phrases (sequences) for the training set and 57 for the validation set. On the other hand, the loss curves for the monolingual and monolingual + code switched corpus configurations show the models learnt during training time.

To evaluate how well all of these configurations learnt, we used PER and WER measures to compare their results for training and test sets. Table 19 shows the results obtained by each configuration. For monolingual dataset and code-switched dataset with the stress marker, we had an error in the sequence size; some sequences were longer than the size limit (100 tokens). We got an error during the prediction of the validation and training set because of the sequence size. Given time limitations, it was not possible to re-train the models with the correct size of sequences for those sets. Instead, we eliminated the sequences that did not satisfy the size limit. The validation set of the monolingual dataset was reduced to 809 sentences, and the training set of the code-switched dataset was reduced to 365 sentences. We reported the performance for the best model under the last five best models during training.

Configuration	Training set		Validation set		Test set	
	PER	WER	PER	WER	PER	WER
Stress and Monolingual[1]	1.28	0.95	3.50	5.90	3.30	5.47
Stress and Code-switched[2]	2.30	3.00	75.83	80.80	72.66	83.64
Stress and Monolingual + Code-switched	1.46	1.22	4.36	7.27	4.34	7.30
No stress and Monolingual	2.31	2.18	3.41	4.60	3.29	4.44
No stress and Code-switched	2.42	2.38	51.88	69.7	53.45	73.39
No stress and Monolingual + Code-switched	3.03	3.50	4.46	6.23	3.88	5.72

Table 19: PER and WER in percentage for each configuration. [1] The validation set was reduced to 809. [2] The training set was reduced to 365

The results confirm the models with the configuration for the code-switched corpus do not generalise well; we can see it is the case of overfitting. Besides, in general for validation and test set, the models without the stress marker had better measures than the configurations without the stress marker (only configurations with the same dataset can be directly comparable). It is clear that a small corpus is not suitable for this kind of approach. In that sense, the models trained with only the code-switched corpus would require more data to avoid overfitting. In the following analysis, we focus on the other two configurations, monolingual and monolingual plus code-switched datasets.

In the interest of comparing all the models directly, we evaluate their performance only in the code-switched test set, which is the kind of data we want our model to perform well. Table 20 shows the PER and WER measures obtained for the monolingual and monolingual plus code-switched datasets.

From the tables 19 and 20, we can see that, although the monolingual configurations had good performance in the monolingual test set, their performance on the code-switched test set was worse. The PER increased by 17% and 20% for the configuration with the stress marker and the configuration without it, respectively. Moreover, the WER increase by 26% and 35% for the configuration with the stress marker and the configuration without

Configuration	Test set	
	PER	**WER**
Stress and Monolingual	20.16	31.62
Stress and Monolingual + Code-switched	**8.17**	**15.63**
No stress and Monolingual	23.74	39.08
No stress and Monolingual + Code-switched	**6.96**	**14.13**

Table 20: PER and WER in percentage for the test set of the code-switched corpus

it, respectively. On the contrary, we see that by introducing code-switched sentences to the monolingual dataset, the models were able to perform much better. If we check at the proportion of code-switched sentences introduced concerning the monolingual sentences, we see that, by introducing about 6% of sentences from the code-switched corpus, the models had an improvement around 12% for the configuration with the stress marker and 17% for the configuration without the stress marker.

Given the fact we did not run cross-validation either hyperparameter tunning, we cannot draw definite conclusions about the results obtained. However, we see these results as promising taking into account that the Transformer architecture has shown the ability to learnt two different phonologies and apply them even in unseen data such is the case for the monolingual configuration tested on the code-switched corpus. As preliminary experiments, the models trained show that the Transformer architecture is capable of learning the phonologies, by conditioning the phonetic transcription with the language code token, and the language-specific stress pattern. We ask the reader to bare this into mind for the following examination of the results.

Comparing the measures strictly, we see the best trained-model is the configuration which does not take into account the stress pattern. However, the differences between that configuration and the configuration that also learns the stress pattern are small, around 1% in both measures. Which can be considered a bearable drop in the performance as long as we will not need to train or use another system to get the stress pattern for each language. In TTS systems, the stress pattern is needed to have an accurate intonation of the words, having a wrong stress pattern can cause the listeners to do not understand what it is being said.

Due to the sequence size limit, we had to trim some sentences of the code-switched corpus, as we explained this caused some of the phrases in the code-switched corpus ended up being monolingual. We wanted to verify the performance of the two best models, only with the phrases containing words from the two languages. Table 21 shows the results obtained for 35 code-switched phrases that were in the test set of the code-switched corpus. The variation from the scores achieved for the whole test set and the ones achieved for the code-switched phrases is about 1.2% for the PER measure and 3.2% for the WER measure. The overall scores for the test set are improved because of the monolingual sentences. The metrics are suggesting the French words in the

Configuration	**PER**	**WER**
Stress and Monolingual + Code-switched	9.31	18.78
No stress and Monolingual + Code-switched	8.36	17.40

Table 21: PER and WER in percentage for the 35 code-switched phrases

We investigated the kind of mistakes that the model made in the subset of 35 code-switched phrases. We found that most of the mistakes are related to French words. There were 65 mistakes in the configuration with the stress marker, 34 from them were French words. Also, there were 61 mistakes in the configuration without the stress marker, where 39 of them were French words. In the phrases, there are 69 French words, which means the models failed predicting for about 49% and 57% of words, for the model with the stress marker and the model without it, respectively. We consider those percentages as a high rate error. On the other hand, not all the error have the same impact in the transcription, for example, transcribing the French word ‘Aubisque’ as /ob’is‘ke/ instead of /ob’isk/ may not be understood by a native speaker since there is the use of a basque phoneme /s‘/ and the pronunciation of /e/, on the contrary, transcribing the French word ‘Louis’ as /lu’i/ instead of /lw’i/ it will sound different to a native speaker, but it will be understood.

Table 22 shows some of the errors that may have a less severe effect. We also found words that were transcribed as e-Speak would do it. However, the transcription in the gold standard was not the same as in e-Speak, which is a mistake in the gold standard. Bear in mind that a non-native speaker of the dialect manually checked the gold standard.

Another intriguing behaviour is the use of the Basque phonology for French words, for example, ‘Maxime’ being transcribed as /maSim/ but in another context being well transcribed with the French phonology (/maks’im/). It seems that the Transformer model is learning how the context can influence the pronunciation of the French words. However, it will require more data and training time to learn those correlations properly. An interesting example of this is the transcription of the French name ‘Philippe’ in the training set. The phonetic transcription of ‘Philippe’ following the French phonology is /fil’ip/, however, it can change if the name is declined with a Basque suffix. For example ‘Philippek’ will be

Type	Prediction	Target	Example
Stress Pattern	'a	a	Oçafrain: /os'afR'e /
	u	'u	Petersbourg-eko: /p@tERsbuRg-eko/
Vowels	'e	E	Jean-Michel: /Z'a miS'el/
	o	O	Aubert: /oB'eRt/
	'i	'j	Hiriart: /hir'iaR/
	'o	O	Sarkozy: /saRk'ozi/
	u	w	Louis: /lu'i/
	'i	j	Marie-Agnés: /maR'iaJ'e/
Allophones	B	b	Aubert: /oB'eRt/
Overpronunciation	@	∅	Banque: /ba k@/
	t	∅	Aubert: /oB'eRt/

Table 22: Example of 'tolerable' errors for the model that includes the stress marker

transcribed as /fil'ippek/. We see that in the nine occurrences of the name 'Philippe', the model was able to transcribe correctly 8 of them. We believe this may be a feature that can be exploited in further experiments.

4.3 Proof of concept: LID plus G2P

In an endeavour to see the behaviour of the two tasks combined, we run an experiment of the workflow: from sentences to labelled tokens including the language code, and from the labelled tokens to the phonetic transcription. Considering the test set of the LID models was different from the test set used in the G2P, we decided to give more importance to the unseen phrases by the G2P model. We wanted to see if a word was wrongly labelled, the G2P model was able to catch the mistake and assign the correct transcription. We run the experiment with the same 35 code-switched phrases used in the G2P task (see 4.2). From those phrases, 23 were in the training set of the CRF classifier and the Max-Entropy classifier.

We tried the two best LID classifiers, that is, the tuned CRF configuration No. 6 (see table 17) and the best Max-Entropy classifier after cross-validation (see section 15). The output of the LID classifiers was used as input for the G2P models. We tried with the two G2P models that were trained with both monolingual and code-switched sentences (see table 21).

Tables 23 and 23 show the metrics for the different language codes for the CRF classifier and the Max-Entropy classifier, respectively. Table 25 shows the results with the different of the two task combined.

The results obtained were congruent with the performance of the classifiers and models independently. The best performance was the combination of the best LID classifier (the

Language code	**Precision**	**Recall**	**F1-measure**	**Support**
bqe	0.957	0.991	0.974	225
fra	0.967	0.885	0.908	69

Table 23: Metrics for the CRF classifier

Language code	**Precision**	**Recall**	**F1-measure**	**Support**
bqe	0.995	0.951	0.953	225
fra	0.843	0.855	0.849	69

Table 24: Metrics for the Max-Entropy classifier

G2P model	**LID classifiers**	
	Max-Entropy	*CRF*
Stress and Monolingual + Code-switched	11.49 (PER)	11.06 (PER)
	23.30 (WER)	22.72 (WER)
No stress and Monolingual + Code-switched	12.00 (PER)	**10.37 (PER)**
	23.20 (WER)	**20.44 (WER)**

Table 25: Results of combining LID classifiers and G2P models measured with PER and WER, both in percentage

CRF classifier) and the best G2P model (No stress pattern and monolingual + code-switched dataset). As expected, the combination of the two components increases the error rate, PER increased 2%, and WER increased 3%. The magnitude increased it is similar to the increment when we when from the whole code-switched test set (including monolingual phrases) to the only code-switched phrases.

In our analysis of the ten wrong labelled French words (they were tagged as Basque), apart from the name ‘Nicolas’ we did not observe capability from the G2P models to overcome LID classification mistakes. Interestingly, the name ‘Piarres’ was labelled as Basque in the LID classification, and correctly transcribed with the G2P model; this name was wrongly tagged in the ground truth. Considering a good lexicon in the TTS system, the use of the LID classifier plus G2P model will account for the cases of OOV words; we considered the result of this proof of concept as favourable towards the improvement of the phonetic transcription of French words in the TTS.

5 Conclusions and Future work

In this document, we presented the results of the work accomplished for this master's thesis. In this section, we revise our work and how this could be continued in the future.

5.1 Conclusions

This master's thesis presented a multilingual approach to improve the phonetic transcription for code-switched texts done by the TTS system for the Navarro-Lapurdian Basque dialect. To do so, we combined two tasks, a Language Identification (LID) task, and a Grapheme-to-Phoneme (G2P) task.

This work explored two classifiers for the LID task, a Max-Entropy classifier trained on character 5-grams and a Conditional Random Fields (CRF) classifiers trained on monolingual (French and Basque) and code-switched sentences. In the experimentation, the best configuration for the Max-Entropy classifier achieved an F1-measure of 0.702 for the French class, and the best setting for the CRF obtained an F1-measure of 0.828 for the French class. The CRF overcame the Max-Entropy classifier by including features of the word context and the information of the Name Entities (NE).

The Max-Entropy experiments showed that including word boundary for the N-gram calculation improved the performance of the classifier. This result accounts for the Zipf's law, that describes the ratio between the occurrence frequency of n-grams and their position in the rank. This result is more evident when the two languages are very different from the morphological point of view, as it is the case for Basque and French. Moreover, including another language in the training phase will help the performance if the included language is distant from the two main languages (in our case French and Basque). Experiments, including Spanish as noise, did not help the performance of French classifiers. While including English as noise helped the performance of Basque classifier.

The code-switching in the Navarro-Lapurdian texts found was mainly to introduce French NE. The corpus-driven approach helped to boost the performance of the CRF classifier; we tokenised the sentences using a Basque tokeniser and split those words that were NE and were declined. Across all the experiments, this approached resulted in a better performance than a general tokenisation.

This work contributed to the first version of annotations of language code and NE information for the Navarro-Lapurdina corpus. We annotated the corpus both for the LID module and for the G2P module; `.json` and `.txt` versions of the corpus are available for future usage. Although using available monolingual corpora did help to train the different modules and configurations, it is clear from the results that the more code-switched sentences we have, the better the phenomenon will be modelled.

We run preliminary experiments to train a multilingual G2P model using the transformer architecture. The results showed that the trained models were able to learn the two phonologies (French and Basque) and the two stress pattern. We believe the several attention spaces that the transformer architecture uses allows the models to learn all the correlations between the language code and the proper phonetic transcription. On the other hand, in cases where the code-switching was at the morpheme level, the G2P models showed an acceptable performance where the French words were transcribed according to the suffix context. That means when a French word was declined, as long as the word and the suffix where both annotated with the language code (French for the word and Basque for the suffix), the model was able to model the articulation phenomenon and adapt the standard rules.

The best G2P model had a PER of 6.96% and a WER of 14.13%; this model did not include the stress pattern in the transcription. Additional, the performance of the second best model did not differ a lot from the first one, it got a PER of 8.17% and a WER of 15.63% and it included the stress pattern in the transcriptions. Both models were trained using monolingual French and Basque sentences and code-switched sentences. Although the results obtained, we cannot raise strong conclusions given that the experiments were preliminary and further experiments and evaluations are required to confirm the nature of the training data does not condition the results. Bear in mind that the phonology of the Basque language establishes certain one-to-one relation between the orthography characters and the phonemes. For languages with more complex phonology, these results may not apply.

Finally, the best Max-Entropy classifier and the best G2P model were integrated into the TTS system. The new TTS system does include the language information of the word being analysed, provides the phonetic transcription of OOV French words. The TTS system still processes the sentence word by word. However, this time if the word is French, it searches for its phonetic transcription in the lexicon, and if it is not found, it will use the G2P model. In order to evaluate the impact of the new linguistic module, it will be necessary to complete the synthesis process and to run a listeners test.

5.2 Future work

According to the results obtained for the different experiments, we identified two primary types of steps that can be done to continue this work. The first type is related to the TTS integration, also considered mid-term type. The second type corresponds to the activities towards building more data-adapted models, also considered long-term type.

Currently, the TTS system has been integrated with the best Max-Entropy classifier. However, this only allows for a word-level analysis of the phrase. Although it is sufficient to obtain an F1-measure of about 0.7, it ignores the code-switching phenomenon. Adapting the TTS systems so a phrase analysis can be done would be the first step for a TTS system for code-switched texts. Having the option of phrase analysis also allows us to include the CRF classifier into the TTS system. Including the CRF classifier will increase the LID accuracy and therefore improve the phonetic transcription.

Additionally, the integration of the two tasks (LID and G2P) into the TTS system was done through shell calls. A further step could be testing the system response to different environments and evaluating its scalability. Especially, in the case of the G2P model, the shell strategy can be replaced by serving the model on a server and consuming it as required. The tensorflow library offers the possibility of serving several versions of the models with better management of the computational resources.

Concerning the LID methods, we mentioned the Navarro-Lapurdian code-switching in texts was mainly to introduce NE; an interesting next step could be to explore models trained mainly on French NE. Another further step is to work around with unsupervised learning techniques. Given the low-resource status of the Navarro-Lapurdian dialect, to study techniques that do not require annotated data could help to increase the available training data. The dialect is used in local journals, but the difficulty to automatically labelled the passages makes it hard to leverage the use of those resources in machine learning approaches. Unsupervised techniques could overcome those problems and provided the benefit of using more data for training.

Finally, the G2P models are at an early stage of research. It will be necessary to perform hyperparameter tuning to adjust the parameters that perform the best for the multilingual phonetic transcription, as well as, to do cross-validation to discard dataset dependencies. Furthermore, it would be interesting to verify if the results encountered with the Navarro-Lapurdian code-switching apply for other language pairs of code-switching. Especially, in pair of languages where the phonology of one or both languages is being affected by the code-switching phenomenon, we presuppose this is the case for code-switching at the morpheme level. By running experiments on different language pairs, we not only would be able to check if the transformer architecture is suitable for the multilingual G2P problem but to explore the code-switching phenomenon in the phonological boundaries. However, that is not a trivial task; usually, there are not high-quality phonetic transcriptions, especially for code-switching utterances. Also, since the G2P models are being trained using the transcriptions, this may suppose a limitation for the model to learn the actual phenomenon.

References

Nagaraj Adiga and S. R.M. Prasanna. Acoustic Features Modelling for Statistical Parametric Speech Synthesis: A Review. *IETE Technical Review (Institution of Electronics and Telecommunication Engineers, India)*, 4602:1–20, 2018. ISSN 09745971. doi: 10.1080/02564602.2018.1432422. URL `https://doi.org/10.1080/02564602.2018.1432422`.

Rodrigo Agerri, Josu Bermudez, and German Rigau. IXA pipeline: Efficient and Ready to Use Multilingual NLP tools. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, 2014. European Language Resources Association (ELRA). ISBN 978-2-9517408-8-4.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. *3rd International Conference on Learning Representations, {ICLR} 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, pages 1–15, 2014. URL `http://arxiv.org/abs/1409.0473`.

Basque Government. Fifth Sociolinguistic Survey. Technical report, Basque Autonomous Community Department of Education, Language Policy and Culture, Navarre and Iparralde, 2011. URL `http://www.euskara.euskadi.net/r59-738/en/%0A%0Acontenidos/informacion/sociolinguistic_research2011/en_2011/2011.html`.

Maximilian Bisani and Hermann Ney. Joint-sequence models for grapheme-to-phoneme conversion. *Speech Communication*, 50(5):434–451, 5 2008. ISSN 0167-6393. doi: 10.1016/J.SPECOM.2008.01.002. URL `https://www.sciencedirect.com/science/article/abs/pii/S0167639308000046`.

Erman Boztepe. Issues in Code - Switching : Competing Theories and Models. *Teachers College, Columbia University Working Papers in Applied Linguistics & TESOL*, 3(2):1–27, 2003. doi: 10.7916/D8ZP4JMT. URL `https://tesolal.columbia.edu/article/issues-in-code-switching/`.

Daniela Braga, Luís Coelho, and Fernando Gil Vianna Resende. A Rule-Based Grapheme-to-Phone Converter for TTS Systems in European Portuguese. *2006 International Telecommunications Symposium*, pages 328–333, 2006. doi: 10.1109/ITS.2006.4433293.

Gokul Chittaranjan, Yogarshi Vyas, Kalika Bali, and Monojit Choudhury. Word-level Language Identification using {CRF}: Code-switching Shared Task Report of {MSR} {I}ndia System. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 73–79, Doha, Qatar, 2014. Association for Computational Linguistics. doi: 10.3115/v1/W14-3908. URL `https://www.aclweb.org/anthology/W14-3908`.

Aliya Deri and Kevin Knight. Grapheme-to-Phoneme Models for (Almost) Any Language. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 399–408, Berlin, Germany, 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1038. URL `https://www.aclweb.org/anthology/P16-1038`.

Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. Building Large Monolingual Dictionaries at the Leipzig Corpora Collection: From 100 to 200 Languages. *In Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 759–765, 2012. URL `https://www.cancer.org/cancer/breast-cancer/about/how-does-breast-cancer-form.html`.

Le Quan Ha, E I Sicilia-Garcia, Ji Ming, and F J Smith. Extension of Zipf's Law to Word and Character N-grams for English and Chinese. In *Computational Linguistics and Chinese Language Processing*, volume 8, pages 77–102, 2003. URL `https://www.aclweb.org/anthology/O03-4004http://www.aclweb.org/anthology/O03-4004`.

Inma Hernaez, Eva Navas, Juan Luis Murugarren, and Borja Etxebarria. Description of the AhoTTS conversion system for the Basque language. *4th ISCA Tutorial and Research Workshop (ITRW) on Speech Synthesis*, 2001. URL `http://aholab.ehu.es/users/inma/publicaciones/ss2001.pdf`.

Pierre-Edouard Honnet, Alexandros Lazaridis, Philip N Garner, and Junichi Yamagishi. The SIWIS French Speech Synthesis Database – Design and recording of a high quality French database for speech synthesis. Technical Report Idiap-RR-03-2017, Idiap, 2017. URL `http://publications.idiap.ch/downloads/reports/2017/Honnet_Idiap-RR-03-2017.pdf`.

E T Jaynes. Information Theory and Statistical Mechanics. *Phys. Rev.*, 106(4):620–630, 5 1957. doi: 10.1103/PhysRev.106.620. URL `https://link.aps.org/doi/10.1103/PhysRev.106.620`.

Sittichai Jiampojamarn, Grzegorz Kondrak, and Tarek Sherif. Applying Many-to-Many Alignments and Hidden Markov Models to Letter-to-Phoneme Conversion. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, number April, pages 372–379, Rochester, New York, 2007. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/N07-1047`.

Ben King and Steven Abney. Labeling the Languages of Words in Mixed-Language Documents using Weakly Supervised Methods. *Proceedings of the 2013 Conference of the North {A}merican Chapter of the Association for Computational Linguistics: Human Language Technologies*, (June):1110–1119, 2013. URL `https://www.aclweb.org/anthology/N13-1131`.

Guillaume Klein, Yoon Kim, Yuntian Deng, Vincent Nguyen, Jean Senellart, and Alexander Rush. {O}pen{NMT}: Neural Machine Translation Toolkit. In *Proceedings of the 13th Conference of the Association for Machine Translation in the {A}mericas (Volume 1: Research Papers)*, pages 177–184, Boston, MA, 2018. Association for Machine Translation in the Americas. URL `https://www.aclweb.org/anthology/W18-1817`.

Artur Kulmizev, Bo Blankers, Johannes Bjerva, Malvina Nissim, Gerardus Gertjan van Noord, Barbara Plank, and Martijn Wieling. The Power of Character N-grams in Native Language Identification. *The 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 382–389, 2017. doi: 10.18653/v1/w17-5043.

Ye Kyaw Thu, Win Pa Pa, Yoshinori Sagisaka, and Naoto Iwahashi. Comparison of Grapheme-to-Phoneme Conversion Methods on a {M}yanmar Pronunciation Dictionary. In *Proceedings of the 6th Workshop on South and Southeast {A}sian Natural Language Processing ({WSSANLP}2016)*, pages 11–22, Osaka, Japan, 2016. The COLING 2016 Organizing Committee. URL `https://www.aclweb.org/anthology/W16-3702`.

John D Lafferty, Andrew McCallum, and Fernando C N Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, volume 2001 of *ICML '01*, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1-55860-778-1. URL `http://dl.acm.org/citation.cfm?id=645530.655813`.

Joseba Lakarra, R L Trask, and José Ignacio Hualde. *Towards a history of the Basque language*. J. Benjamins Pub. Co., 1995.

David Lasagabaster. Language Use and Language Attitudes in the Basque Country. In *Multilingualism in European Bilingual Contexts: Language Use and Attitudes*, volume 2, chapter 3, pages 65–89. Multilingual Matters 135, 2007. ISBN 9781853599309.

Leipzig's Corpora. eus_wikipedia_2016_300K.tar.gz. `http://wortschatz.uni-leipzig.de/en/download/`, 2016a. [accessed May 2019].

Leipzig's Corpora. eng_wikipedia_2016_10K.tar.gz. `http://wortschatz.uni-leipzig.de/en/download/`, 2016b. [accessed May 2019].

Leipzig's Corpora. fra_wikipedia_2010_1M.tar.gz. `http://wortschatz.uni-leipzig.de/en/download/`, 2010. [accessed May 2019].

Leipzig's Corpora. spa_wikipedia_2016_10K.tar.gz. `http://wortschatz.uni-leipzig.de/en/download/`, 2016c. [accessed May 2019].

Giovanni Molina, Fahad AlGhamdi, Mahmoud Ghoneim, Abdelati Hawwari, Nicolas Rey-Villamizar, Mona Diab, and Thamar Solorio. Overview for the Second Shared Task on Language Identification in Code-Switched Data. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 40–49, Austin, Texas, 2016.

Association for Computational Linguistics. doi: 10.18653/v1/W16-5805. URL `https://www.aclweb.org/anthology/W16-5805`.

Sumi S Nair, Rechitha C R, and Santhosh Kumar C. Rule-Based Grapheme to Phoneme Converter for Malayalam. *International Journal of Computational Linguistics and Natural Language Processing*, 2(7):417–420, 2013.

Eva Navas, Inma Hernaez, Daniel Erro, Jasone Salaberria, Beñat Oyharçabal, and Manuel Padilla. Developing a Basque TTS for the Navarro-Lapurdian Dialect. In *Advances in Speech and Language Technologies for Iberian Languages*, pages 11–20, Cham, 2014. Springer International Publishing. ISBN 978-3-319-13623-3. doi: 10.1007/978-3-319-13623-3{\_}2.

Josef R. Novak, Nobuaki Minematsu, and Keikichi Hirose. WFST-based Grapheme-to-Phoneme Conversion: Open Source Tools for Alignment, Model-Building and Decoding. *Proceedings of the 10th International Workshop on Finite State Methods and Natural Language Processing*, (1):45–49, 2012. URL `https://www.aclweb.org/anthology/W12-6208`.

Ben Peters, Jon Dehdari, and Josef van Genabith. Massively Multilingual Neural Grapheme-to-Phoneme Conversion. *CoRR*, abs/1708.0:19–26, 2017. URL `http://arxiv.org/abs/1708.01464`.

Arnaud Pierard, Daniel Erro, Inma Hernaez, Eva Navas, and Thierry Dutoit. Surgery of Speech Synthesis Models to Overcome the Scarcity of Training Data. In *Advances in Speech and Language Technologies for Iberian Languages*, pages 73–83, Cham, 2016. Springer International Publishing. ISBN 978-3-319-49169-1.

Lawrence R Rabiner and Ronald W Schafer. *Introduction to Digital Speech Processing*, volume 1. Now Publishers Inc., Hanover, MA, USA, 2007. doi: 10.1561/2000000001. URL `http://dx.doi.org/10.1561/2000000001`.

Sai Krishna Rallabandi and Alan W. Black. On building mixed lingual speech synthesis systems. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pages 52–56, 2017. ISSN 19909772. doi: 10.21437/Interspeech.2017-1244. URL `http://dx.doi.org/10.21437/Interspeech.2017-1244`.

Georges Rebuschi. Weak and Strong Genitive Pronouns in Northern Basque: A diachronic perspective. In *Towards a history of the Basque language*, chapter 12, pages 313–356. 1995.

Sophie Roekhaut, Sandrine Brognaux, Richard Beaufort, and Thierry Dutoit. eLite-HTS: a NLP tool for French HMM-based speech synthesis. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, (September): 2136–2137, 2014. ISSN 19909772.

Mike Rosner and Paulseph-John Farrugia. A Tagging Algorithm for Mixed Language Identification in a Noisy Domain. In *INTERSPEECH*, pages 190–193. ISCA, 2007. URL http://dblp.uni-trier.de/db/conf/interspeech/interspeech2007.html#RosnerF07.

Sunayana Sitaram and Alan W Black. Speech Synthesis of Code-Mixed Text. *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 3422–3428, 5 2016.

Thamar Solorio, Elizabeth Blair, Suraj Maharjan, Steven Bethard, Mona Diab, Mahmoud Ghoneim, Abdelati Hawwari, Fahad AlGhamdi, Julia Hirschberg, Alison Chang, and Pascale Fung. Overview for the First Shared Task on Language Identification in Code-Switched Data. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 62–72, Doha, Qatar, 2014. Association for Computational Linguistics. doi: 10.3115/v1/w14-3907. URL https://www.aclweb.org/anthology/W14-3907.

Paul Taylor. *Text-to-Speech Synthesis*. Cambridge University Press, Cambridge, 2009. ISBN 9780511816338. doi: 10.1017/CBO9780511816338. URL http://ebooks.cambridge.org/ref/id/CBO9780511816338.

Shubham Toshniwal and Karen Livescu. Jointly Learning to Align and Convert Graphemes to Phonemes with Neural Attention Models. *CoRR*, abs/1610.0, 10 2016. URL http://arxiv.org/abs/1610.06540.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. (Nips), 2017. ISSN 1469-8714. doi: 10.1017/S0952523813000308. URL http://arxiv.org/abs/1706.03762.

Yogarshi Vyas, Spandana Gella, Jatin Sharma, Kalika Bali, and Monojit Choudhury. POS Tagging of English-Hindi Code-Mixed Social Media Content. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 974–979. Association for Computational Linguistics, 2014. doi: 10.3115/v1/d14-1105. URL https://www.microsoft.com/en-us/research/publication/pos-tagging-of-english-hindi-code-mixed-social-media-content/.

Mikołaj Wypych, Emilia Baranowska, and Grażyna Demenko. A Grapheme-to-Phoneme Transcription Algorithm Based on the SAMPA Alphabet Extension for the Polish Language. *International Congress of Phonetic Sciences-15*, pages 2601–2604, 2003. URL https://www.internationalphoneticassociation.org/icphs-proceedings/ICPhS2003/papers/p15_2601.pdf.

Kaisheng Yao and Geoffrey Zweig. Sequence-to-Sequence Neural Net Models for Grapheme-to-Phoneme Conversion. *CoRR*, abs/1506.0, 5 2015. URL https://arxiv.org/abs/1506.00196http://arxiv.org/abs/1506.00196.

Koldo Zuazo. The Basque Country and the Basque Language: An overview of the external history of the Basque langauge. In *Towards a history of the Basque language*, chapter 1, pages 5–30. 1995.