Identifying and Modeling Code-Switched Language

Víctor Soto Martínez

Submitted in partial fulfillment of the
requirements for the degree
of Doctor of Philosophy
in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2020

ABSTRACT

Identifying and Modeling Code-Switched Language

Víctor Soto Martínez

Code-switching is the phenomenon by which bilingual speakers switch between multiple languages during written or spoken communication. The importance of developing language technologies that are able to process code-switched language is immense, given the large populations that routinely code-switch. Current NLP and Speech models break down when used on code-switched data, interrupting the language processing pipeline in back-end systems and forcing users to communicate in ways which for them are unnatural.

There are four main challenges that arise in building code-switched models: lack of code-switched data on which to train generative language models; lack of multilingual language annotations on code-switched examples which are needed to train supervised models; little understanding of how to leverage monolingual and parallel resources to build better code-switched models; and finally, how to use these models to learn why and when code-switching happens across language pairs. In this thesis, I look into different aspects of these four challenges.

The first part of this thesis focuses on how to obtain reliable corpora of code-switched language. We collected a large corpus of code-switched language from social media using a combination of sets of anchor words that exist in one language and sentence-level language taggers. The newly obtained corpus is superior to other corpora collected via different strategies when it comes to the amount and type of bilingualism in it. It also helps train better language tagging models. We also have proposed a new annotation scheme to obtain part-of-speech tags for code-switched English-Spanish language. The annotation scheme is composed of three different subtasks including automatic labeling, word-specific questions labeling and question-tree word labeling. The part-of-speech labels obtained for the Miami Bangor corpus of English-Spanish conversational speech show very high agreement and

accuracy.

The second section of this thesis focuses on the tasks of part-of-speech tagging and language modeling. For the first task, we proposed a state-of-the-art approach to part-of-speech tagging of code-switched English-Spanish data based on recurrent neural networks. Our models were tested on the Miami Bangor corpus on the task of POS tagging alone, for which we achieved 96.34% accuracy, and joint part-of-speech and language ID tagging, which achieved similar POS tagging accuracy (96.39%) and very high language ID accuracy (98.78%).

For the task of language modeling, we first conducted an exhaustive analysis of the relationship between cognate words and code-switching. We then proposed a set of cognate-based features that helped improve language modeling performance by 12% relative points. Furthermore, we showed that these features can also be used across language pairs and still obtain performance improvements.

Finally, we tackled the question of how to use monolingual resources for code-switching models by pre-training state-of-the-art cross-lingual language models on large monolingual corpora and fine-tuning them on the tasks of language modeling and word-level language tagging on code-switched data. We obtained state-of-the-art results on both tasks.

Table of Contents

List of Figures

List of Tables

Acknowledgments

First and foremost, I would like to thank my advisor Professor Julia Hirschberg. When I first met Julia, I was a masters student trying to convince her to take me on for a research project on emotion detection. I am so thankful she took that chance on me. Since that day, working with her has been a privilege. Julia has been an exemplary advisor and mentor, and a constant source of scientific knowledge, encouragement and support. I can only aspire to be as good a researcher, advisor and colleague as her in the future.

Special thanks goes to my dissertation committee members Mona Diab, Kathleen McKeown, Smaranda Muresan and Andrew Rosenberg for accepting to be part of the committee and graciously lending their time to review this thesis. Their observations and feedback have made my research and this thesis better.

I am very grateful to the whole Computer Science department at Columbia University. Thanks to Jessica Rosa, Elaine Roth, Maria Joanta, Rob Lane, Daisy Nguyen and Lester Mau for so much administrative support and help throughout the years.

I want to thank every member of the Speech Lab for these years of research talks, incisive feedback and conversations: Daniel Bauer, Nishi Cestero, Erica Cooper, Bob Coyne, Sarah Ita Levitan, Rivka Levitan, Gideon Mendels, Rose Sloan, Morgan Ulinski, Ismael Villegas, Laura Willson, Shirley Xia, Brenda Yang, Michelle Levine, Svetlana Stoyanchev and so many more, thank you!

I was lucky to spend three summers at Google doing research in the Speech group. I would like to thank Pedro Moreno and Françoise Beaufays for giving me the opportunities to join their teams in New York City and Mountain View, respectively. I also want to thank Olivier Siohan, Fadi Biadsy and Daan Van Esch for their technical support and research guidance during my summers at Google.

I am incredibly thankful to “la Caixa” foundation for awarding me the fellowship that

made my American academic journey possible and inevitably changing my life. I would like to thank IARPA for providing funding during parts of my PhD through the IARPA Babel and IARPA Material programs. Similarly, I would like to thank Google for the support provided for research on code-switching through a Google research award.

I am very grateful to all my friends for their continuing support throughout all these years. Back in Spain, I want to thank Bea, Dani, Alberto, Jesus, Jorge O., Nacho, Paya, Diego, Fer, Irene, Jorge G., Juanma, Silvana and Oscar. In New York, I want to thank Uri, Anna, Avner, Bea, China, Diana, Elvira, Evita, Ferran, Karl, Lexi, Matt, Marta, Merche, Michael, Nick and Pilar. And of course my volleyball Riots: Byron, Cam, Erik, Felipe, Italo, Jeff, Juan, Marty and Ruben. Every moment with you guys has helped me keep my sanity.

Finally, this thesis is dedicated to my family: to my parents Rosa and Juan, and my siblings Lorena and Alejandro. Moving to New York from Madrid and you all to start an academic career has been the hardest thing I have ever done. I have missed you more than words can say, but I am so thankful for the infinite source of love and support that I have received from across the Atlantic. Without you, this thesis would not have been possible. Gracias.

Chapter 1

Introduction

Linguistic code-switching is the alternation between multiple languages during written or spoken communication. Code-switching is widely regarded as a functioning form of communication in multilingual communities and should not to be confused with other forms of bilingualism like the use of loanwords or calques, interlanguages or pidgin languages. It is characterized by the following features: speakers are fully fluent in both languages; speakers have a sense of judgment over what is grammatically admissible and what is not; and code-switched utterances are produced without hesitation.

Code-switching can occur at different linguistic levels. At the phrase level, code-switching can be inter-sentential, when it occurs across the limits of a sentence or utterance (e.g. "Where is she? No te ha llamado aún?"); or intra-sentential when the switch occurs within the boundaries of the utterance (e.g. "Aparcar el coche took me forever").

At the morphology level, code-switching can happen when the language switch happens from one morpheme onto another. For example, the word "oetverkocht" (sold out) is formed by the Dutch lexeme "verkocht" (sold) and the Limburgish morpheme "oet" (out). Similarly, switching can happen at the syntactical level when a syntactic structure from one language is used in another. For example, "Llámame de vuelta" (Call me back) in American Spanish, which follows a syntactic structure that is not native to Spanish. Code-switching also has effects on several aspects of speech production; for example some studies on voice onset time have shown a wide range of effects between monolingual and bilingual adult speakers. In this thesis, we will focus on lexical code-switching, when the switch occurs

at the word level ("His favorite course is matemáticas"). More specifically we focus on intra-sentential lexical code-switching.

Code-switching is a pervasive phenomenon in multilingual communities. It is used for pragmatic purposes, such as style-shifting between topics [Milroy and Gordon, 2003], or used in specific situations [Beebe, 1981]. It can also be used to signal social identity or to mark social belonging to two different groups [Woolard, 1998]. Other theories suggest that code-switching is not only used to reflect social situations, but to create them too [Auer, 1984]. Given how prevalent code-switching is as a communicative resource, the importance of developing NLP technologies for code-switched data is immense. In the US alone there is an estimated population of 56.6 million Hispanic people [US Census Bureau, 2014], of which 40 million are native speakers [US Census Bureau, 2015]. In India, it is estimated that 26% of the population (more than 314.9 million people) is bilingual, according to the 2011 census [Ministry of Home Affairs, Government of India, 2011]. Most of these speakers routinely code-switch.

One of Artificial Intelligence's ultimate goals is to enable seamless natural language interaction between artificial agents and human users. In order to achieve that goal, it is imperative that users be able to communicate with artificial agents as they do with other humans. In addition to such real time interactions, code-switched language is also pervasive in social media [David, 2001; Danet and Herring, 2007; Cárdenas-Claros and Isharyanti, 2009]. So any system which attempts to communicate with these users or to mine their social media content needs to deal with code-switched language.

However, despite the continually growing importance of code-switching, very little research has been done to develop NLP approaches to code-switched language. Code-switching presents serious challenges to all language technologies, including part-of-speech tagging, parsing, language modeling, machine translation, and automatic speech recognition, since techniques developed on one language quickly break down when that language is mixed with another.

In our opinion there are main four challenges to developing code-switched models:

1. **Code-Switched Corpora:** the lack of sufficiently large code-switched corpora is a big challenge towards building NLP models. Large language corpora are critical

to building generative language models that can later be fine-tuned for downstream tasks.

2. **Linguistic Annotations for Language Tasks:** supervised learning algorithms need linguistic annotations to train machine learning models on. These annotations are very expensive to obtain and often required highly-skilled or trained annotators to produce them. For code-switched language, the annotators are further required to be bilingual, making the collection even more expensive and difficult.

3. **Leveraging Monolingual and Parallel Corpora:** there is a wealth of monolingual, and to a lesser extent, parallel corpora waiting to be exploited for code-switching tasks. These resources can be incorporated in the form of pre-trained word embeddings or language models, and how to fine-tune such models to best exploit their potential on code-switched data is a key challenge.

4. **Incorporating Existing Knowledge about Code-Switching:** although code-switching has been mostly ignored by the computational linguistics community, there has been a large amount of work from linguists, including work on syntactical constraints to code-switching and how switches are triggered. Incorporating such knowledge in machine learning models could help boost performance.

Most of the research efforts on code-switching from the linguistics community have been focused on two goals: a) finding constraints in the way monolingual grammars interact with each other to produce well-formed code-switched speech and b) designing methods to build code-switching grammars from monolingual grammars. The following are the three most important theories that aim to describe the structure of code-switching.

1. Asymmetry: [Joshi, 1982] introduced the notion of asymmetry in code-switching, referring to one language dominating another and supplying the morpho-syntactic framework for the bilingual clause. The dominant language is identified as the Matrix Language (ML) and the other is the Embedded Language (EL). Joshi [1982] proposed a framework where there are just two separate monolingual grammars, the ML grammar G_{ML} and the EL grammar G_{EL}, and a control structure that allows shifting

control from the ML to the EL ($X_{ML} \rightarrow X_{EL}$, for a non-terminal X), but not vice versa.

2. Matrix Language Frame model: Myers-Scotton [1997] further developed Joshi's ideas into the Matrix Language Frame (MLF) model, which defines three different constituents in a code-switched clause: ML islands that are made of ML morphemes and under the control of ML grammar; EL islands that are well-formed according to the EL grammar but are inserted into a ML frame; and Mixed Constituents ML+EL which include morphemes from ML and EL. In the MLF model, the ML and EL are identified empirically following these two principles: 'The Morpheme Order principle', which states that the morphemes contained in a mixed constituent must follow the order of the ML; and 'The System morpheme principle', which states that all system morphemes contained in a mixed constituent must come from the ML.

3. Linear Order Constraints: Sankoff and Poplack [1981] rejects the idea of asymmetry and instead postulates two linear order constraints. The "Equivalence Constraint" postulates that a switch must occur where both languages share the same word order, before and after the switch. The "Free Morpheme Constraint" states that in order for a switch to occur between a free and a bound morpheme, the bound morpheme must be phonologically integrated into the language. In [Sankoff and Poplack, 1981], the authors propose an approach to building a new code-switched grammar G_{cs} that subsumes two monolingual grammars G_1 and G_2. This approach adds rule R from grammar G_i to G_{CS} if its equivalent rule R' in G_j fulfills the equivalence constraint: that every pair of output symbols in G_i maintain order in G_j. If that is not the case, a modified version of the rule is added to G_{CS} such that the out-of-order output symbols be expanded into lexical items from G_i.

4. The Generative Grammar model proposed by [Woolford, 1983] follows a similar scheme to Poplack's grammar building process [Sankoff and Poplack, 1981], with the difference that no rules are altered in any way. Phrase structure rules are drawn freely from both grammars during the construction of constituent structure trees, but the lexicon of each grammar is limited to filling only those terminal nodes created by rules drawn

> from the same language. In the case that there are rules common to both languages, such rules belong simultaneously to both languages and lexical items can be freely drawn.

Meanwhile, the computational linguistics community has mainly focused on the problems of word-level language identification, part-of-speech tagging, and applying existing machine learning methodology to leverage existing monolingual resources. Those contributions will be reviewed in the pertinent Chapters throughout this thesis.

In this thesis, we address different aspects of the four main challenges described above. The first part of the thesis focuses on how to obtain and annotate code-switched data. In Chapter 2 we propose a method to detect code-switching in sentences that we term "anchoring". We use this method to collect a corpus of more than 8,000 tweets which we annotate with word-level language identifiers, and we show that the corpus has a high degree of bilingualism, shows diverse types of switching, and helps yield state-of-the-art word-level identification performance. Chapter 3 proposes a crowdsourcing scheme to obtain part-of-speech tags for English-Spanish code-switched text. The scheme is divided into tasks that include automatic labeling, word-specific questions, and a disambiguation question-tree task. We adapt a previous approach designed for English and adapt it to the code-switched setting and expand it for the Spanish language.

The second part of the thesis focuses on the tasks of part-of-speech tagging, word-level language identification and language modeling. In Chapter 4, we propose a bidirectional LSTM model to perform part-of-speech and simultaneous part-of-speech tagging and language identification. We test the models on the Miami Bangor corpus of conversational speech and show that our models are superior to previous state-of-the-art models on code-switched language, and are also competitive on monolingual corpora. Chapter 5 presents an exhaustive statistical analysis on the relationship between code-switching and cognate words, part-of-speech roles and entrainment, on the largest corpus used to date for such analysis. We prove and disprove certain aspects of the Clyne's triggering hypothesis. Following from that research, in Chapter 6 we propose a set of cognate-based features that capture orthographic, phonetic and semantic similarities between cognate pairs and use them on the task of language modeling. The cognate-based features show perplexity im-

provements similar to those obtained by manually labeled gold features like part-of-speech tags and language identifiers. Furthermore, we show that these features can be used across (similar) language pairs. In Chapter 7 we present ongoing work on how to pre-train cross-lingual language models on large collections of monolingual corpora and fine-tune them for the tasks of word-level language identification and language modeling for code-switched data. Finally, we present our conclusions in Chapter 8.

Part I

Automatic Collection and Annotation of Code-Switched Data

Chapter 2

Collecting Code-Switched Data from Social Media

2.1 Introduction

In this Chapter, we address the problem of mining code-switched data from the web. The task of finding code-switched data is of key importance. Real examples of code-switching are needed to train statistical machine learning models for both unsupervised and supervised learning. However, very little code-switching corpora exist from which researchers can train them and the question of how to acquire code-switched data from web and social media resources automatically and accurately remains largely unaddressed.

Finding real examples of code-switching in large streams of data, like social media platforms or the world wide web is extremely challenging for a variety of reasons. Classifying a sentence as code-switched requires accurate word-level language identification, or a code-switching point detection algorithm that would need to be trained on specific language pairs. Existing language identification algorithms work well at the document, paragraph and even sentence level, but underperform at the word level. At the same time, performing word-level language identification on large collections of corpora to find examples of code-switching is extremely expensive.

In this Chapter, we introduce three simple and computationally cheap methods to finding code-switching in large collections of data. Our methods make use of "anchor words",

which are defined as words that can only exist in one language from a large pool of languages, and also sentence-level language identifiers.

We apply our method to Twitter data. Twitter data has been mined extensively for many Natural Language Processing and speech tasks [Mendels *et al.*, 2015; Kouloumpis *et al.*, 2011] as one of the only major platforms that provides an API for data collection. From Twitter, we collected a set of more than 43,000 tweets. We obtained language identifiers for a subset of 8,000 tweets using crowdsourcing with high inter-annotator agreement and accuracy. We validated our Twitter corpus by comparing it to the Spanish-English corpus of code-switched tweets collected for the EMNLP 2016 Shared Task for Language Identification, in terms of code-switching rates, language composition and amount of code-switch types found in both datasets. We then trained language taggers on both corpora and showed that a tagger trained on the EMNLP corpus exhibits a considerable drop in accuracy when tested on the new corpus and a tagger trained on our new corpus achieves very high accuracy when tested on both corpora.

The remainder of the Chapter is organized as follows. In Section 2.2 we give an overview of previous work on the topic of finding and collecting code-switched data. In Section 2.3 we present our anchoring method for retrieving code-switched tweets. Section 2.4 provides the details of our Twitter collection pipeline. Section 2.5 describes the language identification (LID) task we used to crowdsource the word language tags for the data collected. In Section 2.6.1, we compare the corpus we acquired using this method with a corpus of tweets that was collected for the EMNLP 2016 Shared Task for Language Identification in code-switched (CS) Data. We compare them in terms of the amount of bilingualism they contain and their code-switching rate – i.e., how frequently writers switch their language in the corpus. In Section 2.6.2 we train and test language ID taggers on our corpus and the Workshop corpus and compare their performance. Finally, we present our conclusions in Section 2.7

2.2 Related Work

In the past few years there have been increasing efforts on a variety of tasks using code-switched data, including part-of-speech tagging [Solorio and Liu, 2008b; Vyas *et al.*, 2014;

Jamatia *et al.*, 2015; AlGhamdi *et al.*, 2016], parsing [Goyal *et al.*, 2003], language modeling [Franco and Solorio, 2007; Li and Fung, 2012; Adel *et al.*, 2013b,a; Li and Fung, 2014], code-switching prediction [Solorio and Liu, 2008a; Elfardy *et al.*, 2014], sentiment analysis [Vilares *et al.*, 2015; Lee and Wang, 2015] and even speech recognition [Ahmed and Tan, 2012; Lyudovyk and Pylypenko, 2014].

The task that has received most of the attention has been Language Identification on code-switched data, thanks in part to the First and Second Shared Tasks on EMNLP 2014 and 2016 [Solorio *et al.*, 2014; Molina *et al.*, 2016]. Many of the current state-of-the-art models for Language Identification perform sequence labeling using Conditional Random Fields [Al-Badrashiny and Diab, 2016] or Recurrent Neural Networks [Jaech *et al.*, 2016b]. In the 2016 Shared Task the best performing system on the MSA-DA dataset used a combination of both [Samih *et al.*, 2016] on top of word and character-level embeddings, and the best performing system on the ES-EN dataset used logistic regression [Piergallini *et al.*, 2016] and character n-gram features.

On the task of finding and collecting code-switched data from the web, which is the focus of this Chapter, Çetinoglu [2016] obtained a corpus of German-Turkish tweets by automatically computing dictionaries of pure German and Turkish from a million Turkish, German and English tweets. They subsequently used those dictionaries to automatically tag ten million Turkish tweets from which they obtained 8,000 potentially code-switched tweets which they manually filtered down to 680.

Samih [2016] obtained a corpus of forum posts written in MSA and the Darija Dialect following this iterative process: they first started with a list of 439 words exclusive to Darija which they used to retrieve forum posts that contained one of the exclusive words; they then added all the words from the retrieved posts to the list of Darija words. They repeated the process until the corpus reached a certain size. The authors do not target MSA language explicitly during this iterative process under the assumption that MSA is ubiquitous in written Arabic. They obtained a corpus of 223K tokens with 73.9% of code-switched forum posts.

Barman *et al.* [2014] used a group of university students as data source to find code-switched media. They found a Facebook group and 11 Facebook users from which they

collected 2,335 posts and 9,813 comments. Vyas *et al.* [2014] collected almost seven thousand comments from 40 manually selected code-switched Facebook posts from three celebrity pages and the BBC Hindi news page. Finally, Jamatia *et al.* [2015] collected tweets and Facebook posts from a University billboard page, although it is unclear if they specifically targeted code-switched content or not.

The organizers of the EMNLP Shared Tasks on Language Identification in code-switched Data followed a semi-automatic approach. For the first Shared task, code-switched data was collected for the pairs Spanish-English (ES-EN), Mandarin-English (MAN-EN), Nepali-English (NEP-EN) and Modern Standard Arabic-Dialectal Arabic (MSA-DA). The social media sources they targeted were Twitter for all language pairs and Facebook for NEP-EN and blog comments for MSA-DA. For Twitter, their approach consisted in first locating code-switchers and then collecting their posts and posts from their followers and/or followees. For ES-EN, they located a subset of code-switchers by querying the Twitter API with frequent English words, and restricted results to tweets identified as Spanish by Twitter from users based in Texas and California. For NEP-EN, they started from a group of acquaintances that were known to code-switch and then identified their followers and followers of their followers that they found were code-switchers too. For Mandarin-English, they started by looking at the most followed Twitter users in Taiwan. They then added those users that they manually checked were code-switchers to their pool, and repeated a similar process on their followees. For MSA-DA, they seeded the search with text from Egyptian public figures. For the Second Shared task the language pairs were ES-EN and MSA-DA. For ES-EN they restricted the search of code-switchers to those based in New York and Miami and seeded the search from local radio station accounts. Again, they continued looking for followers and followees of the radio stations that tweeted code-switched messages. For MSA-DA, the same collection method from the 2014 Shared Task was reused.

All of these approaches to code-switched data collection, except [Samih, 2016], rely on manual inspection to some degree in order to either add a user to the code-switcher pool or select a post for collection. In the next section we introduce a fully automatic approach to finding and collecting code-switched data that is not dependent on manually curating lists of users.

2.3 Anchoring Methods

We define an anchor as a word which belongs to only one language from a large pool of languages. The motivation behind using anchor words stems from a simple rule that we impose to detecting code-switched sentences: "A sentence is code-switched in $L_1 + L_2$ if and only if it contains at least one anchor from language L_1 and at least one anchor from language L_2, and contains no anchors from any other language from the pool of languages $\mathbb{L}$."

The set of anchor words for a language L_i is computed as the set difference between its word lexicon $V(L_i)$ and the union of all other lexicons in the language pool:

$$AnchorSet(L_i) = V(L_i) \setminus \cup_{j!=i} V(L_j) \tag{2.1}$$

Note that the identification of the anchor sets for a given language pair depends upon the monolingual corpora used.

We can relax the definition of anchors in two different ways. First, in the context of detecting $L_1 + L_2$ language, we say a word is a "weak anchor" if it is seen in monolingual L_1 corpora, and never seen in monolingual L_2 corpora. Second, querying the Twitter API with every possible pair of one Spanish and one English anchor is unproductive because there are billions of possible queries and most of them would have no results. To avoid this problem we relaxed the definition of code-switching to: "a sentence is code-switched if and only if it is predicted to be L_1 by a monolingual automatic Language Identification program and contains at least one weak anchor from the L_2 anchor set." With this new rule we require only one anchor from one of our language pair plus language id results favoring the other member of the pair. We note that the definition of weak anchors closely resembles the definition of blacklisted words used by Tiedemann and Ljubešić [2012], although their application was to discriminate between a set of very similar languages (Serbian, Croatian and Bosnian).

Using these definitions, we performed a preliminary study on the task of classifying an utterance as monolingual or code-switched on the EMNLP 2016 Shared Task Corpus of Spanish+English tweets. Details of the collection and contents of that corpus were given in Section 2.2. We computed the anchors for Spanish and English from the Leipzig corpora

Collection (LCC), released 2007 to 2014 [Goldhahn *et al.*, 2012]. The LCC is a collection of corpora for a large set of languages from comparable sources (e.g. Wikipedia, news articles, websites). We computed the word lexicon of every language in the corpus from the news dataset for that language, and then we computed the anchor list first following equation 2.1. Words that contained numbers or tokens from a list of 31 punctuation tokens were discarded. In total the language pool contained 134 languages. The Spanish anchor set contained 50.68% of the words from the Spanish word lexicon and the English anchor set contained 54.37% of the words from the English lexicon. In both cases, this is one of the smaller percentages from the pool of 134 languages. In comparison, German, French and Italian kept 79.01, 59.67 and 62.94% of their lexicons, while other languages like Chinese and Japanese kept 93.40 and 72.18%.

Table 2.1 shows Precision, Recall and F1-Score results on the task of classifying a tweet as code-switched (CS) or monolingual (mono) for the strong definition of anchors, weak anchors and the weak anchor + LID approach. We report results on the test partition of the EMNLP 2016 Shared Task Corpus. The language ID used is langid.py [Lui and Baldwin, 2012].

Method	Class	Precision	Recall	F1-score
Anchors	Mono	0.58	1.00	0.73
	CS	0.94	0.03	0.07
Weak Anchors	Mono	0.68	0.98	0.80
	CS	0.93	0.38	0.54
Weak +LID	Mono	0.66	0.98	0.79
	CS	0.93	0.33	0.49

Table 2.1: Code-switched sentence detection performance using Anchoring: Strong Anchors, Weak Anchors and Weak Anchors with Language ID.

The top subtable from Table 2.1 shows the results we obtained for this task using our strong definition of anchor. Not surprisingly, we achieved very high precision, but very low recall. High precision and low recall is a secondary effect from the restrictiveness of the definition of anchor set and code-switched sentence, since anchors are not defined exclusively

in terms of L_1 and L_2, but from a large pool of languages. This means that the words in the anchor set are most likely to be very low-frequency words. Furthermore the fact that a sentence must have at least one anchor from both languages and none from all the other languages, guarantees that much of the data will be rejected as not code-switched even when bilingual speakers of the languages in question would agree that it is.

The middle subtable from Table 2.1 shows the results on the task using weak anchors as defined above. At the expense of 0.01 absolute precision points, recall is improved by almost 0.35 points.

The bottom subtable of Table 2.1 shows results using weak anchors and Language Id. Although with this method the recall drops 0.03 points with respect to the weak anchors, we achieve the advantage of being able to reduce the number of queries we need for the collection, and make the search less restrictive. In the next section of the Chapter we use weak anchors with the Language ID restriction to collect code-switched tweets.

2.4 Data Collection

We used Babler[1] [Mendels *et al.*, 2016] to collect code-switched data from Twitter. Babler is a tool designed for harvesting web-data for NLP and machine learning tasks. Babler's pipeline is launched by querying a seed word $s \in S$ using Twitter's API. The tweets retrieved by the query are later processed and passed through a set of filtering rules R which are predefined for the task. Tweets were not selected or filtered based on time period, topic, genre, dialect, and so on.

Following the definition of "weak anchor plus Language Id" given in section 2.3 we used the "weak" anchors to seed the Twitter API and the filtering rules R to enforce the LID restriction. To further reduce the number of required queries we also sorted our "weak" anchors by frequency. The weak anchors were computed from the GigaCorpus dataset of Broadcast News data. R uses Twitter's LID to only allow tweets that were seeded from a Spanish anchor and classified as English or vice versa. Although we required the Twitter API to return only exact matches to our seed terms, we found that in fact Twitter performs

[1]Babler is publicly available from `https://github.com/gidim/Babler`

stemming.

Our method differs from the prior art in two aspects. First, we derive our word lists from non-noisy pure monolingual corpora which reduces the risk of including out-of-language tokens. Second, instead of performing local filtration our method is implemented based only on API calls thus increasing our potential dataset to every public tweet available. Overall we collected 14,247 tweets that were seeded from Spanish weak anchors and classified as English by the Twitter API and 28,988 tweets that were seeded from English weak anchors and classified as Spanish.

2.5 Crowdsourcing Language Tags

While we designed our data collection pipeline to save only code-switched tweets, we next needed to test this, as well as to obtain manual annotations for our language modeling research.

From the more than forty-three thousand tweets that were collected, we randomly chose a subset of 8,285 tweets for our "Anchored" tweets corpus [2]. We crowdsourced language tags for every word in our Anchored tweet dataset. Each word was tagged as English (EN), Spanish (ES), Ambiguous between English and Spanish (AMBIG), Mixed English-Spanish (MIXED), Named Entity (NE), Foreign Word (FW), Other (OTHER) and Gibberish (UNK). "Named Entities" were defined as single proper names or part of a name or title that referred to persons, places, organizations, locations, brands, goods, initials, movie titles and song titles. A word is to be tagged as "Ambiguous" when it can be used in both English and Spanish, but there is not enough context to decide its use in the current tweet. A word is to be tagged "Mixed" when the word does not exist in Spanish or English, but consists of a combination of elements from both, e.g. the word "ripeado" which contains the English root "rip" and the Spanish morpheme "-ado". The category "Other" is to be used to tag punctuation, numbers, emoticons, retweet symbols, and other non-lexical items. Finally the "Gibberish" category is for tokens whose meaning cannot be identified.

[2] All the anchor wordlists, tweet IDs and their crowdsourced language tags are publicly available in `http://www.cs.columbia.edu/~vsoto/files/lrec_2018_package.zip`

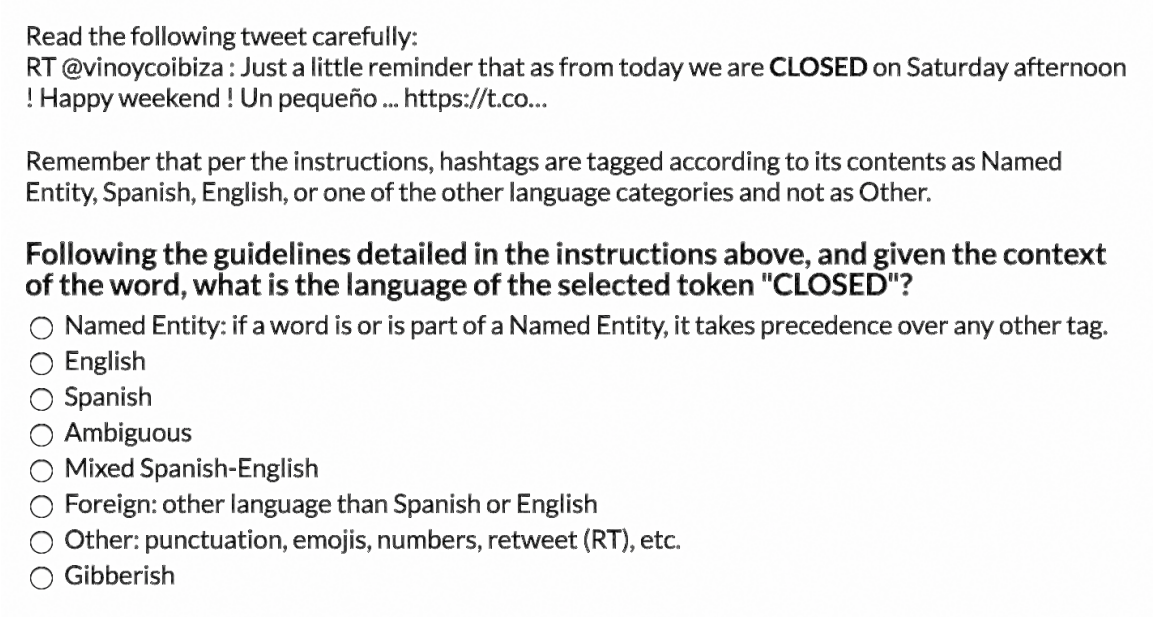

Figure 2.1: Word-level language annotation crowdsourcing task. Figure shows the interface on Crowdflower.

We used the guidelines designed for the annotation of the EMNLP 2016 Shared Task dataset, with some minor changes, including a large number of examples per language tag, and reminders to the annotators throughout the instructions and question statements that a) hashtags were to be tagged with the language tag of the words in the hashtag, and b) Named Entities had precedence over any other language tag; since these were the test questions they had the most difficulty with in our initial test.

We used Crowdflower to crowdsource language tags for our tweets. An example of the task our workers were asked to complete can be seen in Figure 2.1. Our workers were pre-screened using a quiz of twenty test questions. If three or more test questions were missed during the initial quiz, the worker was denied access to the task. Furthermore, workers were required to be certified for the Spanish language requirement in Crowdflower. Only workers from Argentina, Canada, Mexico, Spain, U.K. and U.S.A. were allowed access to the task. The task was designed to present 20 questions per page plus one test question used to assess workers' performance. When a worker reached an accuracy lower than 85% on these test questions, all their submitted judgments were discarded and the task made

Lang Tag	#Tokens	Avg. Conf
ES	40,208	0.97
EN	30,372	0.93
AMBIG	919	0.55
MIXED	129	0.54
NE	15,260	0.88
FW	1,815	0.77
OTHER	1,994	0.80
UNK	546	0.59

Table 2.2: Number of tokens and average confidence per Language ID tag from the crowdsourced annotations for the Anchored Twitter corpus.

subsequently unavailable. Every set of 19+1 judgments was paid 1 cent (USD).

In total, we collected three judgments per token. The average inter-annotator agreement was 92.33% and the average test question accuracy was 91.1%. These metrics demonstrate that the crowdsourced language labels are of high-quality. For every token for which we crowdsourced a language tag, Crowdflower computes the confidence on the language tag as the level of agreement between all the contributors that predicted that language tag weighted by the contributors' trust scores. The language tag with highest confidence is then chosen as aggregated prediction. Table 2.2 shows the average confidence per language tag across all tokens. It can be seen that workers struggled the most when tagging words as Mixed, Ambiguous or Gibberish.

2.6 Evaluation

2.6.1 Data Assessment

Given the crowdsourced LID labels, we can assess the quality of the retrieved anchored tweets by computing their degree of bilingualism and how frequently code-switching occurs within them. We compare these measures to the EMNLP 2016 CS Shared Task corpus [Molina *et al.*, 2016].

Metric	Workshop		Anchored
	Train-Dev	Test	Full
# Tweets (K)	14.4	10.7	8.5
# Tokens (K)	172.8	121.4	130.7
# Switches (K)	7.4	7.8	10.2
Avg. # Switches	0.52	0.73	**1.19**
Switched words (%)	4.30	6.42	**7.77**
Switched tweets(#)	4,116	4,617	5,958
Switched tweets(%)	28.56	43.09	**69.89**
0 switches (%)	**71.44**	56.91	30.11
1 switch (%)	12.86	21.38	39.57
2 switches (%)	11.34	16.65	19.53
3 switches (%)	2.50	2.88	5.81
4 switches (%)	1.27	1.66	3.32
5 switches (%)	0.29	0.33	0.84
6 switches (%)	0.20	0.17	0.43
7 switches (%)	0.05	0.02	0.23
8 switches (%)	0.03	0.00	0.12

Table 2.3: Code-switching statistics for the EMNLP 2016 Workshop and Anchored Twitter datasets. The bottom subtable shows the percentage of tweets that contain N code-switches.

The train and dev tweets from the 2016 Shared Task were the train and test sets from the 2014 Shared Task [Solorio *et al.*, 2014], whereas the test split was collected specifically for the 2016 task. The collection schemes used in 2014 and 2016 were explained in detail in Section 2.2. Table 2.3 provides the overall statistics describing this corpus in comparison to ours. We report the train-dev and test splits of the EMNLP 2016 Workshop Shared Task corpus separately since they were collected using different methods. As can be seen in Table 2.3, our subset of 8,525 tweets had an average of 1.19 code-switches per tweet, with 7.77% of words in a tweet being followed by a switch. 69.89% of our tweets contained at least one or more switches. In comparison, the Workshop corpus had an average of 0.61 code-switches per tweet, with 5.17% of tokens followed by a switch. Only 34.75% tweets contained at least one switch. The test set of the Workshop corpus shows greater degrees

	Workshop		Anchored
Lang Tag	Train-Dev	Test	Full
ES	24.51	63.44	34.44
EN	55.33	13.95	24.73
AMBIG	0.23	0.00	0.70
MIXED	0.04	0.00	0.10
NE	2.09	1.72	11.68
FW	0.01	0.02	1.39
OTHER	17.62	20.84	26.53
UNK	0.17	0.02	0.42

Table 2.4: Language composition for the EMNLP 2016 Workshop and Anchored Tweets datasets. Amounts are shown in percentages, at the token level.

of bilingualism and a better switching rate: Test corpus tweets averaged 0.73 code-switches per tweet, with 6.42% of tokens followed by a switch and contained 43.09% code-switched tweets overall. Based on these metrics alone, it would appear that our anchoring method improves over the earlier approach considerably.

Table 2.4 shows the language composition of the three datasets: Workshop training-dev, Workshop test, and the full Anchored dataset. From this table we can see that the train-dev portion of the workshop corpus has a majority (>55%) of English words, while the test split contains a large majority of Spanish words (>63.44%), perhaps due to seeding the collection of tweets on Spanish-language Radio accounts and followers/ees. In comparison, the Anchored corpus is more balanced, with 34.44 and 24.73% of Spanish and English tokens. It also has a higher rate of Named Entities and Other tokens. We believe this is due to the updated annotation guidelines that emphasized the subtleties involved in annotating Named Entities and Other tokens. While Table 2.4 compares the corpora by language composition, Table 2.5 examines the corpora by type of switch. The most frequent switch across datasets is Spanish to English (ES-EN), followed by English to Spanish (EN-ES). These account for 63.53%, 74.04% and 52.67% of switches for the Workshop Train-Dev, Workshop Test and Anchored datasets respectively. The next most common type of switch is an English word followed by a sequence of Other tokens and a Spanish word (EN-Other-ES), or Spanish

	Workshop		Anchored
Switch Type	Train-Dev	Test	Full
ES EN	32.06	45.68	29.81
EN ES	31.47	28.36	22.86
EN Other+ ES	15.99	12.28	14.83
ES Other+ EN	15.16	11.05	10.86
ES NE+ EN	1.44	0.99	4.06
EN NE+ ES	0.91	0.36	2.45

Table 2.5: Types of code-switching in the EMNLP 2016 Workshop and Anchored Tweets datasets. TAG+ indicates a sequence of one or more occurrences of that language tag.

Training	Word Accuracy (%)		Avg. F1-Score		Sentence Accuracy (%)	
Corpus	Workshop	Anchored	Workshop	Anchored	Workshop	Anchored
Workshop	95.93	82.09	0.4218	0.3978	67.91	14.20
Anchored	95.13	**91.86**	**0.4655**	**0.5937**	62.60	**40.13**
Combination	**96.91**	91.61	0.4328	0.5617	**73.53**	39.87

Table 2.6: Language tagging accuracy (left) and average f1-score (center) at the word level and language tagging accuracy at the sentence-level (right) for each training and testing combination.

followed by Other and then English (ES-Other-EN). These make up for 31.15%, 23.33% and 25.69% of the switches. Note that this type of switch can be indicative of inter-sentential code-switching if the Other token is a punctuation mark (like '!' in "Holaaaa mis niños bellos!!! I love you guys") or it can be indicative of intra-sentential code-switching if the other token is a Twitter mention, a quote, and so on (e.g "En cuestiones de Rock 'n' Roll I am pretty crossover"). Overall, the typing distribution is more balanced in the Anchored dataset, whereas the Workshop test set has a significant majority of ES – EN switches, due perhaps, again, to the way the collection of tweets was seeded.

Training	Word Accuracy (%)		Avg. F1-Score		Fragment Accuracy (%)	
Corpus	Workshop	Anchored	Workshop	Anchored	Workshop	Anchored
Workshop	85.46	78.96	0.3678	0.3802	84.29	61.67
Anchored	83.85	86.64	0.3617	0.4937	82.61	71.73
Combination	87.44	86.98	0.3722	0.5020	86.51	73.67

Table 2.7: Language tagging accuracy (left) and average f1-score (center) at the word level and language tagging accuracy at the fragment-level (right) for each training and testing combination on the subset of code-switched fragments.

2.6.2 Language Identification

Our second evaluation of the accuracy of our corpus consists of training and testing Language ID taggers on the new dataset and comparing its performance to a tagger trained on the Workshop data. We made use of a high-performing classification model from Jaech *et al.* [2016b]. The model did well on the English-Spanish code-switching 2016 Shared Task, especially considering that it was one of only two models that did not use external resources for training [Molina *et al.*, 2016]. The same model did well on a sentence level language identification task [Jaech *et al.*, 2016a].

We summarize the model architecture and its motivation here. For a full description see [Jaech *et al.*, 2016b]. The model is a hierarchical neural model with one level that operates on character sequences to build a representation for each word and a second level that operates on the sequence of word representations to predict the language tag for each word. In the first level, the model uses convolutional neural network layers to do a soft-version of n-gram matching. The output of this layer is a feature vector that provides a useful signal for the language of each word because languages tend to differ in their character n-gram distributions. The second level of the model is a bidirectional LSTM that takes as input the feature vectors from the previous layer and outputs the predicted tag for each word. The use of the LSTM allows the model to incorporate evidence from tokens far away in the word sequence.

We made one tweak that was not described in [Jaech *et al.*, 2016b]: the standard LSTM

was replaced with an LSTM that has coupled input and forget gates for a 25% reduction in the parameters in the bi-LSTM and a corresponding improvement in speed of computation [Greff *et al.*, 2017]. Operating on the word-level representations allows the LSTM to predict the correct tag for words whose language is ambiguous from just the character-level feature vectors based on the fact that adjacent words are more likely to belong to the same language.

We tuned the model hyper-parameters by training and testing on the train and dev splits of the Workshop dataset, effectively making the task more difficult for the model trained on the Anchored corpus. Table 2.6 shows the word-level and sentence-level accuracy and the average F1-score of the language ID tagset for each training/testing combination.

First, we trained our tagger on the Workshop data (Workshop Model, in Table 2.6) and observed that its performance on the Workshop test set is similar to that reported for this model in the Shared Task (95.93%). The performance of this tagger however sees a big drop of performance on word-level accuracy and sentence-level accuracy when tested on the Anchored test set. This demonstrates that a tagger trained on a corpus comprised of majority of monolingual sentences, with a lower degree of bilingualism and switching rates, has some difficulty generalizing to a more balanced corpus like the Anchored Tweets Corpus.

Second, we partitioned the Anchored corpus into train and test by randomly choosing 1,500 tweets for the test set and leaving the rest for training. We trained a new tagger on the Anchored dataset with the same hyper-parameter settings as the Workshop tagger and report its test performance on Table 2.6 as Anchored tagger. We observed that the performance of this model on the Workshop data is very good, despite the difference between the two datasets: the word-level accuracy only decreases by 0.8% accuracy points with respect to the Workshop model, whereas the sentence-level accuracy decreases by 5.31% points. However the F1-score value sees a relative improvement of 10.36%, which indicates that the new corpus is more similar to the Workshop test split than the Workshop train-dev split. The Anchored-trained tagger achieves 91.86% word-level accuracy on its own test set, with 0.5937 average F1-score value and 40.13% sentence-level accuracy. These results indicate that a tagger trained on the anchored corpus is able to generalize quite well on the same corpus, although overall the classification task is harder than on the Workshop

corpus: the best word-level and sentence-level accuracies in the Workshop test set are much higher than in the Anchored test set.

Finally, we trained a tagger on a combination of the Workshop and Anchored training sets. This combined tagger achieves the best word-level accuracy on the Workshop corpus (96.91%) as shown in the last row of Table 2.6. Similarly the combined tagger also achieves the best sentence-level accuracy on the Workshop test set (73.53%).

Overall, the Anchored tagger achieves the best results on the Anchored test set for every metric (91.86% word-level accuracy, 0.5937 average f1-score and 40.13% sentence-level accuracy), despite being trained on much less data (the anchored train set has 7,025 tweets, the workshop train set has 11,400 tweets and the combined train set has 18,425 tweets). It also achieves the best average f1-score on the Workshop test set (0.4655). The Combination tagger achieves the best word-level and sentence-level accuracy on the Workshop test set (96.91% and 73.53% respectively).

We next examine the performance of the three taggers on the subset of code-switched segments present in each test set in Table 2.7, where we define a code-switched segment as the minimal span of tokens where a point code-switch occurs. Notice that a segment can be longer than two tokens if there is a Named Entity, Other, Mixed or Ambiguous token in between. For example, from the sentence "I watched The Godfather y me encantó", the code-switched segment would be "watched The Godfather y" where "The Godfather" is a Named Entity.

From this table we can see that, in fact, taggers have most difficulty tagging words that occur in the context of a code-switch, since the accuracy of all three models on both test subsets of code-switched segments suffers a steep decline for the results shown for the complete test set in the left subtable of Table 2.6. In the case of the Workshop tagger, its accuracy has relative changes of -10.91 and -3.81% on the full workshop and anchored test sets respectively. The Anchored model sees even larger relative decreases of -11.86 and -5.68%. In comparison, the Combination model has the smallest relative decreases in accuracy, with -9.77 and -5.05%. The same trends can be observed for the average F1-Score and the fragment-level accuracy metrics.

Overall the best performing model is the one trained on the combined training sets,

followed by the Anchored model, which always gets better metric values on its own test set and achieves similar metric values on the Workshop test set when compared to the Workshop tagger. Notice though that the Anchored model was trained on less than 40% of the number of tweets in the Combined train set.

2.7 Conclusions

In this Chapter we presented a method that made use of *anchoring* and monolingual Language ID for detecting code-switched text. We relaxed strict anchoring constraints to query the Twitter API and retrieved code-switched tweets. We crowdsourced language tags for the tokens of 8,285 tweets and found that almost 70% of the collected tweets were indeed code-switched. These tweet exhibit a relatively balanced amount of Spanish and English text and a high amount of code-switching per tweet. The average number of code-switches per tweet in the corpus is 1.19 switches while 7.77% of the tokens are followed by a code-switch. These numbers compare favorably to the 2016 EMNLP Workshop Shared Task Code-Switched Twitter corpus, which was obtained with a different and more labor-intensive method. We evaluated the quality of our new Anchored corpus by training state-of-the-art language taggers and showed that a) a tagger trained on the original Workshop corpus exhibited a more considerable drop in accuracy when tested on the Anchored corpus; and b) a tagger trained on the Anchored corpus achieved very good accuracy on both test corpora. These results show great promise for automatic collection of other code-switched corpora for use in training language models and for other NLP and speech tasks.

Chapter 3

Crowdsourcing Part-of-Speech Tags for Code-Switched Data

3.1 Introduction

High-quality linguistic annotations are extremely valuable for any NLP task, and performance is often limited by the amount of high-quality labeled data available. However, little such data exists for code-switching. In this Chapter, we describe crowdsourcing universal part-of-speech tags for the Miami Bangor Corpus of Spanish-English code-switched speech.

With the advent of large scale machine learning approaches, the annotation of large datasets has become increasingly challenging and expensive. Linguistic annotations by domain experts are key to any language understanding task, but unfortunately they are also expensive and slow to obtain. One widely adopted solution is crowdsourcing. In crowdsourcing, naive annotators submit annotations for the same items on crowdsourcing platforms such as Amazon Mechanical Turk and Crowdflower. These are then aggregated into a single label using a decision rule like majority vote. Crowdsourcing allows one to obtain annotations quickly at lower cost. It also raises some important questions about the validity and quality of the annotations, mainly: a) are aggregated labels by non-experts as good as labels by experts? b) what steps are necessary to ensure quality? and c) how does one explain complex tasks to non-experts to maximize output quality? [Callison-Burch and Dredze, 2010].

The task of crowdsourcing part-of-speech tags is challenging insofar as part-of-speech tagsets tend to be large and the task is intrinsically sequential. This means that workers need to be instructed about a large number of categories and they need to focus on more than the word to tag, making the task potentially longer, more difficult, and thus, more expensive. More importantly, even though broad differences between part-of-speech tags are not hard to grasp, more subtle differences tend to be critically important. An example would be deciding whether a word like "up" is being used as a preposition ("He lives up the street") or a particle ("He lived up to the expectations.")

We present an annotation scheme for obtaining part-of-speech (POS) tags for code-switching using a combination of expert knowledge and crowdsourcing. Part-of-speech tags have been proven to be valuable features for NLP tasks like parsing, information extraction and machine translation [Och *et al.*, 2004]. They are also routinely used in language modeling for speech recognition and in the front-end component of speech synthesis for training and generation of pitch accents and phrase boundaries from text [Taylor *et al.*, 1998; Taylor and Black, 1998; Zen *et al.*, 2009; Hirschberg, 1990; Watts *et al.*, 2011].

We split the annotation task into three subtasks: one in which a subset of tokens are labeled automatically, one in which questions are specifically designed to disambiguate a subset of high frequency words, and a more general cascaded approach for the remaining data in which questions are displayed to the crowd-source worker following a decision tree structure. Each subtask is extended and adapted for a multilingual setting and the universal tagset. The quality of the annotation process is measured using hidden check questions annotated with gold labels. The overall agreement between gold standard labels and the majority vote is between 0.95 and 0.96 for just three labels and the average recall across part-of-speech tags is between 0.87 and 0.99, depending on the task.

The rest of the Chapter is organized as follows. Section 3.2 presents a summary of related work on crowdsourcing linguistic annotations and more specifically part-of-speech tags. Section 3.3 gives an overview of the Miami Bangor corpus, which we obtain annotations for. This corpus will be used throughout the rest of this dissertation. Section 3.4 explains our annotation scheme in detail. Section 3.5 shows our results and finally, Section 3.6 presents out conclusions.

3.2 Related Work

There is a large body of work on the topic of crowdsourcing linguistic annotations for language corpora. In [Snow *et al.*, 2008] the authors crowdsourced annotations in five different NLP tasks. To evaluate the quality of the new annotations they measured the agreement between gold and crowdsourced labels. Furthermore, they showed that training a machine learning model on the crowdsourced labels yielded a high-performing model. Callison-Burch [2009] crowdsourced translation quality evaluations and found that by aggregating non-expert judgments it was possible to achieve the quality expected from experts. In [Hsueh *et al.*, 2009] crowdsourcing was used to annotate sentiment in political snippets using multiple noisy labels. The authors showed that eliminating noisy annotators and ambiguous examples improved the quality of the annotations. Finin *et al.* [2010] described a crowdsourced approach to obtaining Named Entity labels for Twitter data from a set of four labels using both Amazon Mechanical Turk and CrowdFlower. They found that a small fraction of workers completed most of the annotations and that those workers tended to score highest inter-annotator agreements. Jha *et al.* [2010] proposed a two-step disambiguation task to extract prepositional phrase attachments from noisy blog data.

The aggregation scheme is a key component in a crowdsourcing task. Majority voting is widely used but is sensitive to noisy labels. In [Hovy *et al.*, 2013] the authors proposed MACE (Multi Annotator Competence Estimation), an aggregation scheme based on item-response models. MACE learns to identify which annotators are trustworthy and predict correct labels. Similarly in [Rodrigues *et al.*, 2014], a Conditional Random Field (CRF) is used for situations where multiple annotations are available but not actual ground truth. The algorithm proposed there was able to simultaneously learn the CRF parameters, reliability of the annotators and the estimated ground truth.

Previous research has tackled the task of crowdsourcing part-of-speech tags. The authors in [Hovy *et al.*, 2014] collected five judgments per word in a task which consists of reading a short context where the word to be tagged occurs, and selecting the part-of-speech tag from a drop-down menu. Using MACE [Hovy *et al.*, 2013] they obtained 82.6% accuracy and 83.7% when restricting the number of words to be tagged using dictionaries. In his M.S. thesis, Mainzer [2011] proposed an interactive approach to crowdsourcing part-of-

speech tags, where workers are assisted through a sequence of questions to help disambiguate the tags with minimal knowledge of linguistics. Workers following this approach for the Penn Treebank Tagset [Santorini, 1990] achieved 90% accuracy.

In this Chapter, we propose to adapt the monolingual annotation scheme from [Mainzer, 2011] to crowdsource Universal part-of-speech tags in a code-switching setting for the Miami Bangor Corpus. Our main contributions are the following: finding mappings to the universal part-of-speech tagset, extending a monolingual annotation scheme to a code-switching setting, creating resources for the second language of the pair (Spanish) from zero and creating a paradigm that others can adopt to annotate other code-switched language pairs.

3.3 The Miami Bangor Corpus

The Miami Bangor corpus is a conversational speech corpus recorded from bilingual Spanish-English speakers living in Miami, FL. It includes 56 files of conversational speech from 84 speakers. The corpus consists of 242,475 words (transcribed) and 35 hours of recorded conversation. 63% of transcribed words are English, 34% Spanish, and 3% are undetermined. The manual transcripts include beginning and end times of utterances and per word language identification.

The original Bangor Miami corpus was automatically glossed and tagged with POS tags using the Bangor Autoglosser [Donnelly and Deuchar, 2011a,b]. The autoglosser finds the closest English-language gloss for each token in the corpus and assigns the tag or group of tags most common for that word in the annotated language. These tags have three main problems: they are unsupervised, the tagset used is uncommon and not specifically designed for multilingual text, and often the autoglosser does not disambiguate between predicted tags (e.g. the Spanish token "sí" is simultaneously tagged as "yes.ADV.[or].himself.PRON", where "yes" and "himself" are the English glosses and "ADV" and "PRON" their part-of-speech tags). To overcome these problems we decided to a) obtain new part-of-speech tags through in-lab annotation and crowdsourcing and b) to use the Universal Part-of-Speech Tagset [Petrov *et al.*, 2012].

The Universal part-of-speech tagset is ideal for annotating code-switching corpora be-

cause it was designed with the goal of being appropriate to any language. Furthermore, it is useful for crowdsourced annotations because it is much smaller than other widely-used tagsets. Comparing it to the Penn Treebank part-of-speech tagset [Santorini, 1990; Marcus *et al.*, 1993], which has a total of 45 tags, the Universal part-of-speech tagset has only 17: Adjective, Adposition, Adverb, Auxiliary Verb, Coordinating and Subordinating Conjunction, Determiner, Interjection, Noun, Numeral, Proper Noun, Pronoun, Particles, Punctuation, Symbol, Verb and Other. A detailed description of the tagset can be found in `http://universaldependencies.org/u/pos/`.

3.4 Annotation Scheme

The annotation scheme we have developed consists of multiple tasks: each token is assigned to a tagging task depending on word identity, its language and whether it is present in one of three disjoint wordlists. The process combines a) manual annotation by computational linguists, b) automatic annotation based on knowledge distilled from the Penn TreeBank guidelines and the Universal Tagset guidelines, and c) and d) two language-specific crowdsourcing tasks, one for English and one for Spanish. The pseudocode of the annotation scheme is shown in Algorithm 1.

Table 3.1 shows the number and percentage of tokens tagged in each annotation task (second and third column) and the percentage of tokens that was annotated by experts in-lab, either because it was the manual task or because there was a tie in the crowdsourced task. In the next subsections we explain in detail each one of the annotation blocks. All the wordlists and sets of questions and answers mentioned but not included in the following sections are available in Appendices A, B, C, and D.

3.4.1 Automatically Tagged Tokens

For English, the Penn Treebank Annotation guidelines [Santorini, 1990] instructs annotators to tag a certain subset of words with a given part-of-speech tag. We follow those instructions by mapping the fixed Penn Treebank tag to a Universal tag. Moreover we expand this wordlist with a) English words that we found were always tagged with the same Universal

Algorithm 1: Pseudocode of the annotation scheme.

```
function RetrieveGoldUniversalTag (token, lang, tag);
  Input: A word token, lang ID lang and POS tag tag
if IsInUniqueLists(token, lang) then
    return RetrieveAutomaticTag(token, lang);
else if IsInManualAnnotationList(token, lang) then
    return RetrieveManualTag(token, lang);
else
    utag = Map2Universal(token, lang, tag);
    if IsInTSQList(token, lang) then
        utags = TokenSpecificQuestionTask(token, 2);
    else
        utags = QuestionTreeTask(token, lang, 2);
    return MajorityVote([utag, utags]);
```

Task	# Tokens	% Corpus	% by Experts
Automatic	156845	56.58	0.00
Manual	4,032	1.45	1.45
TSQ	57,248	20.65	0.93
English QT	42,545	15.34	0.32
Spanish QT	16,587	5.98	0.08
Total	277,257	100	2.78

Table 3.1: Breakdown of amount of corpus annotated per task.

tag in the Universal Dependencies Dataset and b) low-frequency words that we found only occur with a unique tag in the Bangor Corpus. Similarly, for Spanish, we automatically tagged all the words tagged with a unique tag throughout the Universal Dependencies Dataset (e.g. conjunctions like 'aunque', 'e', 'o', 'y', etc.; adpositions like 'a', 'con', 'de', etc.; and some adverbs, pronouns and numerals) and low frequency words that only occurred with one tag throughout the Bangor corpus (e.g. 'aquella', 'tanta', 'bastantes', etc.).

Given the abundance of exclamations and interjections in conversational speech, we collected a list of frequent interjections in the corpus and tagged them automatically as INTJ. For example: 'ah', 'aha', 'argh', 'duh', 'oh', 'shh'. Finally, tokens labeled as Named Entities or Proper Nouns in the original Miami Bangor Corpus were automatically tagged as PROPN.

3.4.2 Manually Tagged Tokens

We identified a set of English and Spanish words that we found to be particularly challenging for naive workers to tag and which occurred in the dataset in such low frequency that we were able to have them tagged in the lab by computational linguists. Note that a question specific to each one of these tokens could have been designed for crowdsourced annotations the way it was done for the words in section 3.4.3.1. The majority of these are tokens that needed to be disambiguated between adposition and adverb in English (e.g.'above', 'across', 'below', 'between') and between determinant and pronoun in Spanish (e.g. 'algunos/as', 'cuántos/as', 'muchos/as').

3.4.3 Crowdsourcing Universal Tags

We used crowdsourcing to obtain new gold labels for every word not manually or automatically labeled. We started with the two basic approaches discussed in [Mainzer, 2011] for disambiguating part-of-speech tags using crowdsourcing which we modified for a multilingual corpus. In the first task (subsection 3.4.3.1), a question and a set of answers were designed to disambiguate the part-of-speech tag of a specific token. In the second task (subsection 3.4.3.2), we defined two Question Trees (one for English and one for Spanish) that sequentially ask non-technical questions of the workers until the part-of-speech tag is

disambiguated. These questions were designed so that the worker needs minimal knowledge of linguistics. All the knowledge needed, including definitions, is given as instructions or as examples in every set of questions and answers. Most of the answers contain examples illustrating the potential uses for the token.

Two judgments were collected from the pertinent crowdsourced task and a third one was computed from applying a mapping from the Bangor tagset to the Universal tagset. The new gold standard was computed as the majority tag between the three part-of-speech tags.

3.4.3.1 Token-specific questions (TSQ)

In this task, we designed a question and multiple answers specifically for particular word tokens. The worker was then asked to choose the answer that is the most true in his/her opinion. Below is the question we asked workers for the token 'can' (note that users cannot see the part-of-speech tags when they select one of the answers):

In the context of the sentence, is 'can' a verb that takes the meaning of 'being able to' or 'know'?

- Yes. For example: 'I can speak Spanish.' (**AUX**)
- No, it refers to a cylindrical container. For example: 'Pass me a can of beer.' (**NOUN**)

We began with the initial list of English words and the questions developed in [Mainzer, 2011] for English. However, we added additional token-specific questions for words that a) we thought would be especially challenging to label (e.g. 'as', 'off', 'on') and b) appear frequently throughout the corpus (e.g. 'anything', 'something', 'nothing').

We designed specific questions for a subset of Spanish words. Just as for English, we chose a subset of most frequent words that we thought would be especially challenging for annotation by workers like tokens that can be either adverbs or adpositions (e.g.'como', 'cuando', 'donde') or determiners and pronouns (e.g. 'ese/a', 'este/a', 'la/lo'). We modified many of the questions proposed in [Mainzer, 2011], to adapt them to a code-switching

setting and to the universal part-of-speech tagset. For example, the token 'no' can be an Adverb and Interjection in Spanish, and also a Determiner in English. Also, some of our questions required workers to choose the most accurate translations for a token in a given context:

In the context of the sentence, would 'la' be translated in English as 'her' or 'the'?

- The ('La niña está corriendo' becomes 'The girl is running') (**DET**)
- Her ('La dije que parase' becomes 'I told her to stop') (**PRON**)

3.4.3.2 Annotations Following a Question Tree

In this task the worker is presented with a sequence of questions that follows a tree structure. Each answer selected by the user leads to the next question until a leaf node is reached, when the token is assigned a part-of-speech tag. We followed the basic tree structure proposed in [Mainzer, 2011], but needed to modify the trees considerably due again to the multilingual context. For example, the new Question Tree starts by first asking whether the token is an interjection or a proper noun. This is very important since any verb, adjective, adverb or noun can effectively be part of or itself be an interjection or proper noun. If the worker responds negatively, then they are asked to follow the rest of the tree. The resulting tree is slightly simpler than the one in [Mainzer, 2011]. This is mainly because we moved the Particle-Adverb-Adposition disambiguation from this task into the Token-Specific Questions task. On the other hand, we added question nodes designed to disambiguate between main verbs and auxiliary verbs. The following is an example of the annotation task following the English Question Tree:

Read the sentence carefully: "Sabes porque I plan to move in August but I need to find a really good job." In the context of the sentence, is the word 'good':

- A Proper Noun or part of a Proper Noun.

- A single word used as an exclamation that expresses acknowledgement or an emotional reaction.
- None of the above. ✓

In the context, 'good' is a:

- Noun, because it names a thing, an animal, a place, events or ideas.
- Adjective, because it says something about the quality, quantity or the kind of noun or pronoun it refers to. ✓
- Verb, because it is used to demonstrate an action or state of being.
- Adverb, because it tells the how, where, when, when or the degree at which something is done.

Could 'good' be a noun or a verb?

- It could be a Noun. For example, fun can be a noun as in ... or an adjective as in...
- It could be a Verb. For example, surprised can be a verb as in ... or an adjective as in ...
- No, it's definitely an Adjective. ✓

For the Spanish portion of the corpus, we modified the English subtasks still further, adapting them according to the syntactic properties of Spanish. One of the key differences from the English tree concerns verbs in their infinitival form. Users that choose to tag a token as verb are then asked to confirm that the infinitival form is not a noun, and if it is not, to decide whether a verb is acting as main verb or as an auxiliary verb (as a compound verb or periphrasis). Figure 3.2 shows an example of an annotation task using the Spanish Question Tree. More instances of part-of-speech tag disambiguation questions that we created for the Spanish Question Tree include Auxiliary Verb-Main Verb disambiguation, periphrasis detection, and Verb-Noun disambiguation.

Figure 3.1: Example of part-of-speech annotation task using the English Question Tree task.

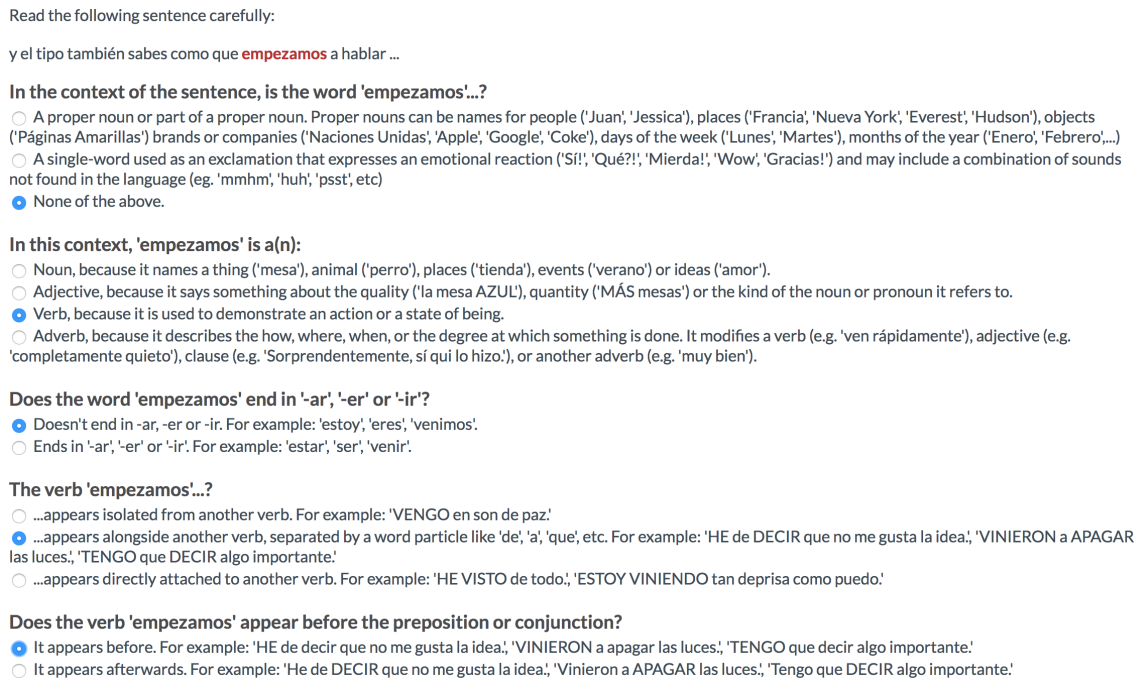

Figure 3.2: Example of part-of-speech annotation task using the Spanish Question Tree task.

3.4.3.3 Mapping Stage

We used the pre-annotated tag from the Bangor corpus as the third tag to aggregate using majority voting. To obtain it, we first cleaned the corpus of ambiguous tags, and then defined a mapping from the Bangor tagset to the Universal tagset. This mapping process was first published in [AlGhamdi *et al.*, 2016].

3.5 Results

We ordered two judgments per token for each of our tasks. Before they were allowed to begin the tasks, workers were pre-screened using a quiz of ten check questions (also referred to as test questions). These check questions are very simple questions that no worker should miss and ensure that the workers have the required knowledge to complete the task. If two or more check questions were missed during the initial quiz, the worker was denied access to the task.

Furthermore, workers were required to be certified for the Spanish language requirement in Crowdflower. Only workers from Argentina, Canada, Mexico, U.K., U.S.A. and Spain were allowed access to the task. The tasks for the workers were designed to present 9 questions per page plus one test question used to assess workers' performance. When a worker reached an accuracy lower than 85% on these test questions, all their submitted judgments were discarded and the task made subsequently unavailable. Every set of 9+1 judgments was paid 5 cents (USD) for the Token-Specific Questions task and 6 cents for the Question Tree tasks.

Table 3.2 shows the number of test questions for each task and of evaluation metrics to estimate the accuracy of the annotations obtained from the crowdsourcing workers. Taking into account all the judgments submitted for test questions, the majority voting tag had an accuracy of 0.97-0.98 depending on the task. These estimations are not expected to match the true accuracy we would get from the two judgments we obtained for the rest of non-test tokens, so we re-estimate the accuracy of the majority vote tag for every subset of one, two, three and four judgments collected, adding the initial Bangor tag. In this case we obtain an average accuracy ranging from 0.89-0.92 with just one token to 0.95-0.96 when using four

Task	TSQ	Eng QT	Spa QT
# Tokens	57.2K	42.5K	16.6K
# Test Questions	271	381	261
Avg. # Judgments per TQ	55.72	28.60	16.28
Accuracy	0.98	0.98	0.97
Avg. Acc of SJ per TQ	0.88	0.89	0.87
Avg. Agrmnt of SJ wrt MV	0.89	0.90	0.87
Accuracy(1+1)	0.89	0.92	0.91
Accuracy(2+1)	0.94	0.92	0.92
Accuracy(3+1)	0.94	0.96	0.96
Accuracy(4+1)	0.96	0.95	0.96

Table 3.2: Accuracy and Agreement measurements per annotation task.

tags.

The best accuracy estimates for our part-of-speech tags are for the option of two crowdsourced tags and the Bangor tag, for which we obtained accuracies of 0.92 to 0.94. When looking at non-aggregated tags, the average accuracy per token of single judgments (SJ) were observed to be between 0.87 and 0.88. Measuring the agreement between single judgments and the majority vote (MV) per token, the average agreement value is between 0.87 and 0.89.

We examined the vote split for every non-test token to obtain a measure of confidence for the tags. We see that we consistently obtained full-confidence crowdsourced tags on at least 60% of the tokens for each of the tasks, reaching 70% for the Spanish Question Tree task. The option for which one of the crowdsourced tags was different from the other two (marked as 2-1 Bangor) on the table occurred between 18% and 23% of the time depending on the task, whereas the split where the Bangor tag was different from the crowdsourced tags (marked as 2-1 CF) occurred only between 10.63 and 12.15% of the time. Finally the vote was split in three different categories only between 1.29% and 4.51% of the time. In those instances, the tie was broken by in-lab annotators. To further evaluate the performance of the annotation process by different tag categories, we examined the recall on the gold test questions. The recall across all tags and tasks is higher than 0.93 except for Interjections and Adjectives for the Spanish Question Tree and Adverbs for the English Question Tree.

Task	TSQ	English QT	Spanish QT
3-0	60.12	67.20	70.09
2-1 (Bangor)	23.20	19.74	17.98
2-1 (CF)	12.16	10.97	10.63
1-1-1	4.51	2.09	1.29

Table 3.3: Voting split per annotation task on the Miami Bangor corpus.

Looking at the failed test questions for Adverbs, it becomes apparent that workers had difficulty with adverbs of place that can also function as nouns, like: 'home', 'west', 'south', etc. For example 'home' in 'right when I got home' was tagged 24 times as a Noun, and only 5 as an Adverb.

3.6 Conclusions

We have presented a new scheme for crowdsourcing Universal part-of-speech tagging of Spanish-English code-switched data derived from a monolingual process which also used a different tagset. Our scheme consists of four different tasks (one automatic, one manual, and two crowdsourced). Each word in the corpus was sent to only one task based upon curated wordlists. For the crowdsourced tokens, we have demonstrated that, taking the majority vote of one unsupervised tag and two crowdsourced judgments, we obtained highly accurate predictions. We have also shown high agreement on the predictions: between 95 and 99% of the tokens received two or more votes for the same tag. Looking at the performance of each part-of-speech tag, our predictions averaged between 0.88 and 0.93 recall depending on the task.

Task	TSQ	Eng QT	Spa QT
ADV	0.98	0.2	1.0
ADJ	1.0	0.97	0.86
ADP	1.0	X	X
AUX	1.0	0.98	1.0
CONJ	1.0	X	X
DET	1.0	X	X
INTJ	1.0	1.0	0.78
NOUN	1.0	1.0	0.96
NUM	1.0	X	X
PART	1.0	X	X
PRON	0.93	X	X
PROPN	X	1.0	X
SCONJ	0.96	X	X
VERB	1.0	0.99	1.0
Average	0.99	0.88	0.93

Table 3.4: Recall per part-of-speech tag and annotation task.

Part II

Part-of-Speech and Language Modeling of Code-Switched Data

Chapter 4

Joint Part-of-Speech and Language ID Tagging for Code-Switched Data

4.1 Introduction

Part-of-speech tagging is a key component of any Natural Language Understanding system and one of the first that researchers employ to process data. As such, it is crucial that part-of-speech taggers be able to process CS content. Monolingual part-of-speech taggers stumble when processing CS sentences due to out-of-vocabulary words in one language, confusable words that exist in both language lexicons, and differences in the syntax of the two languages.

For example, when running monolingual English and Spanish taggers on the CS English-Spanish shown in Figure 4.1, the English tagger erroneously tagged most Spanish tokens, and similarly the Spanish tagger mistagged most English tokens. A tagger trained on monolingual English and Spanish sentences (EN+ES tagger) fared better, making only two mistakes: on the word "when", where the switch occurs (confusing the subordinating conjunction for an adverb), and the word "in" (which exists in both vocabularies). A tagger trained on CS instances of English-Spanish, however, was able to tag the whole sentence

Words:	Ella	lo	había	leído	when	she	was	in	third	grade
Translation:	*She*	*it*	*had*	*read*	-	-	-	-	-	-
Gold:	PRON	PRON	AUX	VERB	SCONJ	PRON	VERB	ADP	ADJ	NOUN
EN Tagger:	NOUN	ADV	NOUN	VERB	ADV	PRON	VERB	ADP	ADJ	NOUN
ES Tagger:	PRON	PRON	AUX	VERB	PROPN	PROPN	PROPN	ADP	X	PROPN
EN+ES Tagger:	PRON	PRON	AUX	VERB	ADV	PRON	VERB	ADV	ADJ	NOUN
CS Tagger:	PRON	PRON	AUX	VERB	SCONJ	PRON	VERB	ADP	ADJ	NOUN

Figure 4.1: Example of an English-Spanish code-switched sentence. The figure shows the original code-switched sentence, English translations of each token, gold part-of-speech tags and the tagging output of an English tagger, of a Spanish tagger, of a tagger trained on English and Spanish sentences, and of a tagger trained on a corpus of code-switched sentences, in that order. Errors made by each tagger are underlined.

correctly.

In this Chapter, we propose an approach to part-of-speech tagging of code-switched English-Spanish data based on recurrent neural networks. We test our model on known monolingual benchmarks to demonstrate that our neural part-of-speech tagging model is on par with state-of-the-art methods. We next test our code-switched methods on the Miami Bangor corpus of English-Spanish conversation, focusing on two types of experiments: part-of-speech tagging alone, for which we achieve 96.34% accuracy, and joint part-of-speech and language ID tagging, which achieves similar part-of-speech tagging accuracy (96.39%) and very high language ID accuracy (98.78%). Finally, we show that our proposed models outperform other state-of-the-art code-switched taggers.

The rest of this Chapter is organized as follows. In Section 4.2 we present an overview of previous work on part-of-speech tagging and language tagging on code-switched data. Section 4.3 gives a short introduction to recurrent neural networks and long short-term memory networks, and Section 4.4 contains the details of the the classification models used in this Chapter. In Section 4.5 we give an overview of the corpora used for our experiments. Section 4.6 details the methodology followed to design our experiments and Section 4.7 contains the experiments and results. In Section 4.8 we present an error analysis of our taggers and finally, Section 4.9 concludes the Chapter.

4.2 Related Work

A variety of tasks have been studied in CS data. For language identification (LID), Rosner and Farrugia [2007] proposed a word-level Hidden Markov Model and a character-level Markov Model to revert to when a word is out-of-vocabulary, and tested these on a corpus of Maltese-English sentences, achieving 95% accuracy. Working on a Bengali-Hindi-English dataset of Facebook posts, Barman *et al.* [2014] employed classifiers using n-gram and contextual features to obtain 95% accuracy.

In the first statistical approach to POS-tagging on CS data, Solorio and Liu [2008b] collected the Spanglish corpus, a small set of 922 English-Spanish sentences. They proposed several heuristics to combine monolingual taggers with limited success, achieving 86% accuracy when choosing the output of a monolingual tagger based on the dictionary language ID of each token. However, an SVM trained on the output of the monolingual taggers performed better than their oracle, reaching 93.48% accuracy. On the same dataset, Rodrigues and Kübler [2013] compared the performance of a POS-tagger trained on CS sentences with a dynamic model that switched between taggers based on gold language identifiers; they found the latter to work better (89.96% and 90.45% respectively). Note, however, that the monolingual taggers from [Solorio and Liu, 2008b] were trained on other larger corpora, while all the models used in [Rodrigues and Kübler, 2013] were trained on the Spanglish corpus.

Jamatia *et al.* [2015] used CS English-Hindi Facebook and Twitter posts to train and test part-of-speech taggers. They found a Conditional Random Field model to perform best (71.6% accuracy), and a combination of monolingual taggers similar to the one in [Solorio and Liu, 2008b] achieved 72.0% accuracy. Again using Hindi-English Facebook posts, Vyas *et al.* [2014] ran Hindi and English monolingual taggers on monolingual chunks of each sentence. Sequiera *et al.* [2015] tested algorithms from [Solorio and Liu, 2008b] and [Vyas *et al.*, 2014] on the Facebook dataset from [Vyas *et al.*, 2014] and the Facebook+Twitter dataset from [Jamatia *et al.*, 2015], and found that [Solorio and Liu, 2008b] yielded better results. Similarly, the authors in [Barman *et al.*, 2016] compared the methods proposed in [Solorio and Liu, 2008b] and [Vyas *et al.*, 2014] on a subset of 1,239 code-mixed Facebook posts from [Barman *et al.*, 2014] and found that a modified version of [Solorio and Liu,

2008b] performed best. They also experimented with performing joint part-of-speech and LID tagging using 2-level factorial Conditional Random Field and achieved statistically similar results.

In [AlGhamdi *et al.*, 2016], the authors tested seven different part-of-speech tagging strategies for CS data: four consisted of combinations of monolingual systems and the other three were integrated systems. They tested them on MSA-Egyptian Arabic and English-Spanish. The first three combined strategies consisted of running monolingual part-of-speech taggers and language ID taggers in different order and combining the outputs in a single multilingual prediction. The fourth approach involved training an SVM on the output of the monolingual taggers. The three integrated approaches trained a supervised model on a) the Miami Bangor corpus (which contains switched and monolingual utterances); b) the union of two monolingual corpora (Ancora-ES and Penn Treebank); c) the union of the three corpora. The monolingual approaches consistently underperformed compared to the other strategies. The SVM approach consistently outperformed the integrated approaches. However, this method was trained on both monolingual and multilingual resources – the Penn Treebank Data for the English model, and the Ancora-ES dataset for the Spanish model. In Section 4.7.4, we run experiments in similar conditions to the integrated approaches from [AlGhamdi *et al.*, 2016], to which we will compare our work.

The main contributions of this Chapter over the previous research on part-of-speech tagging for CS data, are the following: a) Our tagger is a bi-directional LSTM that achieves part-of-speech tagging accuracy comparable to state-of-the-art taggers on benchmark datasets like the Wall Street Journal corpus and the Universal Dependencies corpora. It is the first such model used to train code-switched part-of-speech taggers; b) Our model can simultaneously perform part-of-speech and LID tagging without loss of part-of-speech tagging accuracy; c) We run experiments on the Miami Bangor corpus of Spanish and English conversational speech. However, unlike AlGhamdi *et al.* [2016] which used part-of-speech tags which were obtained from an automatic tagger and then mapped to a deprecated version of the Universal part-of-speech tagset, our experiments are run on newly crowd-sourced Universal part-of-speech tags [Soto and Hirschberg, 2017], which were obtained with high accuracy and inter-annotator agreement.

4.3 Recurrent Neural Networks and LSTMs

A recurrent neural network (RNN) is a feed-forward NN with cyclical connections. Whereas a feed-forward NN can only map from input to output vectors, an RNN is able to map sequences of input vectors to sequences of output vectors. The recurrent connections in an RNN allow a memory of previous inputs to persist in the network's internal state s, therefore influencing the network's subsequent output. A simple RNN with one hidden layer would be modeled as:

$$s_t = f(Ux_t + Ws_{t-1}) \tag{4.1}$$

$$o_t = g(Vs_t) \tag{4.2}$$

where x_t and o_t are the input and output of the network at time t, s_t is the network's internal state at time t, U, W and V are the RNN weight parameters and $f()$ and $g()$ are differentiable functions like tanh, ReLU or softmax.

Standard unidirectional RNNs process sequences in temporal order, using only past context. A bi-directional RNN (Bi-RNN) goes through every training sequence forward and backward using two separate recurrent hidden layers, both of which are connected to the same output layer (the two hidden layers are self-connected but not interconnected). The output layer, therefore, has access to all the context for every example in the sequence. Bi-RNNs generally outperform RNNs [Graves, 2012] and have been used with success in many tasks such as machine translation [Sundermeyer *et al.*, 2014], handwriting recognition [Liwicki *et al.*, 2007] and protein structure prediction [Baldi *et al.*, 1999].

One of the major weaknesses of RNNs is their ineffectiveness in modeling long-term dependencies [Bengio *et al.*, 1994]. Long Short-Term Memory networks (LSTMs) are one type of RNN specifically designed to address this problem [Hochreiter and Schmidhuber, 1997]. The key to LSTM's effectiveness, and its main difference with respect to the rest of mainstream RNNs, lies in the function used to compute its cell state. LSTMs make use of three "gates" to control the cell state: input gate layer, output gate layer and forget gate layer. Broadly speaking, at every time t, the forget gate layer decides what information to discard from the previous cell state, the input gate layer decides what information to add to the cell state, and the output gate layer decides what information to output from the

cell state. LSTMs have been applied with great success to a variety of sequence learning problems like language modeling [Sundermeyer *et al.*, 2012], POS tagging [Huang *et al.*, 2015], acoustic modeling [Sak *et al.*, 2014], machine translation [Sutskever *et al.*, 2014], and image captioning [Vinyals *et al.*, 2015].

4.4 A Model for Neural Part-of-Speech Tagging

For our experiments we used a bi-directional LSTM network similar to the one proposed in [Wang *et al.*, 2015] with the following set of features: 1) word embeddings; 2) prefix and suffix embeddings of one, two and three characters; and 3) four boolean features that encode whether the word is all upper case, all lower case, formatted as a title, or contains any digits. In total, the input space consists of seven embeddings and four boolean features. For the embeddings, we computed word, prefix and suffix lexicons, excluding tokens that appear less than five times in the training set, and then assigned a unique integer to each token. We also reserved two integers for the padding and out-of-lexicon symbols.

We present two architectures for part-of-speech tagging and one for joint part-of-speech and LID tagging. In the most basic architecture the word, prefix and suffix embeddings and the linear activation units are concatenated into a single layer. The second layer of the network is a bidirectional LSTM. Finally, the output layer is a softmax activation layer, whose i-th output unit at time t represents the probability of the word w_t being the part-of-speech POS_i. We refer to this model as Bi-LSTM part-of-speech Tagger for the rest of this chapter and in our tables. For the second model, given the multilingual nature of our experiments, we modified the input space of our Bi-LSTM tagger to make use of the language ID information in our corpus. We added five more boolean features to represent the language ID and add six linear activation units in the first hidden layer, which are then concatenated with the rest of linear activation units and word embeddings in the basic model. This model is referred to as Bi-LSTM part-of-speech tagger + LID features.

Finally, our third model simultaneously tags words with part-of-speech and language labels. The architecture of this model can be seen in Figure 4.2 and follows the Bi-LSTM part-of-speech architecture very closely adding a second output layer with softmax acti-

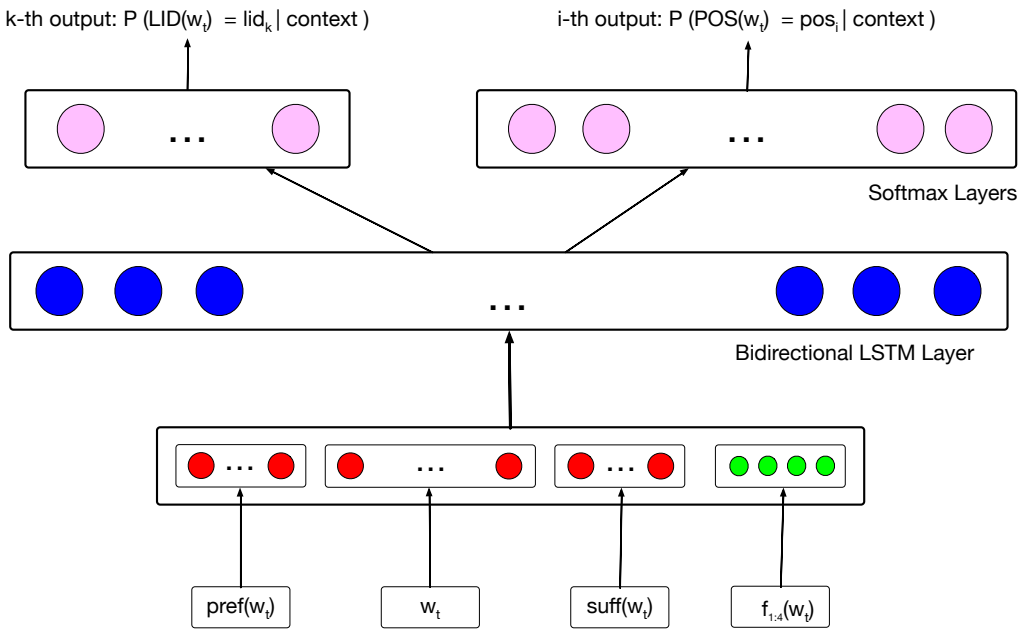

Figure 4.2: The joint POS+LID tagging model. This figure shows only one prefix and one suffix embedding for clarity of presentation.

vations for LID prediction. Note that the part-of-speech and LID output layers are independent and are connected by their weight matrices to the hidden layer, and both loss functions are given the same weight. This model is referred to as joint POS+LID tagger. We implemented our code using the library for deep learning Keras [Chollet, 2015], on a Tensorflow backend [Abadi *et al.*, 2015].

4.5 Datasets

Throughout our experiments we used three corpora for different purposes. The Wall Street Journal (WSJ) corpus was used to demonstrate that our proposed Bi-LSTM part-of-speech tagger is on par with current state-of-the-art English part-of-speech taggers. The Universal Dependencies (UD) corpus was used to train baseline monolingual part-of-speech taggers in English and Spanish that we can use to test on our CS data since both employ the Universal part-of-speech tagset [Petrov *et al.*, 2012]. The Miami Bangor corpus, which contains instances of inter- and intra-sentential CS utterances in English and Spanish, was

Corpus	Split	# Sents	# Toks
WSJ	Train	38.2K	912.3K
	Dev.	5.5K	131.7K
	Test	5.5K	129.7K
UD-EN	Train	12.5K	204.6K
	Dev.	2K	25.1K
	Test	2K	25.1K
UD-ES	Train	14.2K	403.9K
	Dev.	1.6K	43.5K
	Test	274	8.4K
Miami Bangor	Full	42.9K	333.1K
	Train	38.7K	300.3K
	Test	4.2K	32.8K
	Train Inter-CS	36.0K	267.3K
	Test Intra-CS	285	3.6K

Table 4.1: Corpora and splits used for POS tagging experiments.

used for training and testing CS models and comparing these to monolingual models. Table 4.1 shows the number of sentences/utterances and tokens in each dataset split. For the MB corpus, Inter-CS refers to the subset of monolingual sentences and Intra-CS refers to the subset of CS sentences.

4.5.1 Wall Street Journal Corpus

The WSJ corpus [Marcus *et al.*, 1999] is a monolingual English news corpus comprised of 49208 sentences and over 1.1 million tokens. It is tagged with the Treebank tagset [Santorini, 1990; Marcus *et al.*, 1993], which has a total of 45 tags. We used the standard training, development and test splits from [Collins, 2002] which span sections 0-18 19-21 and 22-24 respectively.

4.5.2 Universal Dependency Corpora

Universal Dependencies (UD) is a project to develop cross-linguistically consistent treebank annotations for many languages. The English UD corpus [Silveira *et al.*, 2014] is built from the English Web Treebank [Bies *et al.*, 2012]. The corpus contains data from web media sources, including web logs, newsgroups, emails, reviews and Yahoo! answers. The trees were automatically converted into Stanford Dependencies and then hand-corrected to Universal Dependencies. The corpus contains 16,622 sentences and over 254K tokens. The Spanish UD corpus [McDonald *et al.*, 2013] is built from the content head version of the Universal Dependency Treebank v2.0, to which several token-level morphology features were added. It is comprised of news blog data and has a total of 16,013 sentences and over 455k tokens.

4.6 Methodology

For the experiments involving the Bangor corpus, we performed 4-fold cross-validation (CV) on the training corpus to run grid search and obtain the best learning rate and decay learning rate parameter values. For the experiments on WSJ and UD, we used the official development set. The performance of the best parameter values is reported as "Dev" accuracy. We then trained a model using the best parameter values on the full train set and obtained predictions for the test set (reported as "Test"). When pertinent we also report results on the subset of intra-sentential CS utterances of the test set (reported as "Intra-CS Test").

During cross-validation, each model was trained for a maximum number of 75 epochs using batches of 128 examples. We used early stopping to halt training when the development part-of-speech accuracy had not improved for the last three epochs, and kept only the best performing model. However, when training the final model, we stopped training after the number of epochs that the best model trained for during cross-validation. The loss function used is categorical cross-entropy and we used ADAM [Kingma and Ba, 2015] with its default β_1, β_2 and ϵ parameter values as the stochastic optimization method.

The word embeddings [Bengio *et al.*, 2003] we used were trained with the rest of the

network during training following the Keras implementation [Gal and Ghahramani, 2016]. The size of the embedding layers is 128 for the word embeddings and 4, 8 and 16 for the prefix and suffix embeddings of length 1, 2 and 3 respectively. The Bi-LSTM hidden layer has 200 units for each direction.

Finally, we ran McNemar's test [McNemar, 1947] to show significant statistical difference between pairs of classifiers when the accuracy of the classifiers is similar, and report statistical significance for p-values smaller than 0.05.

4.7 Experiments & Results

In this section, we present our experiments using the three Bi-LSTM models introduced in Section 4.4 and the datasets from Section 4.5. Our goal is a) to show that the basic Bi-LSTM part-of-speech tagger performs very well against known part-of-speech tagging benchmarks; b) to obtain baseline performances for monolingual taggers when tested on CS data; and c) to train and test the proposed models on CS data and analyze their performance when trained on different proportions of monolingual and CS data.

4.7.1 WSJ results

We begin by evaluating the performance of the Bi-LSTM part-of-speech tagger on the benchmark WSJ corpus to show that it is on par with current state-of-the-art English part-of-speech taggers. We trained taggers on three incremental feature sets to measure how much each feature adds. Using only word embeddings we achieved 95.14% accuracy on the test set; adding word features increased accuracy to 95.84%; and adding the prefix and suffix embeddings further increased accuracy by up to 97.10%. This demonstrates that our tagger is on par with current state-of-the-art systems which report 97.78% [Ling *et al.*, 2015], 97.45% [Andor *et al.*, 2016], 97.35% [Huang *et al.*, 2012], 97.34% [Moore, 2014] and 97.33% [Shen *et al.*, 2007] accuracy on their standard test set. Systems most similar to our Bi-LSTM tagger with basic features reported 97.20% in [Collobert *et al.*, 2011] and 97.26% [Wang *et al.*, 2015].

	UD		MB	
Training	Dev	Test	Test	CS Test
UD EN	94.53	94.78	69.97	56.20
UD ES	96.20	95.02	45.13	55.32
UD EN&ES	94.88	94.25	88.17	87.18

Table 4.2: Bi-LSTM part-of-speech tagging accuracy when trained on the Universal Dependency corpora. The left subtable shows the accuracy on the UD dev and test sets. The right subtable shows the accuracy on the MB test set and on the subset of CS utterances.

4.7.2 Universal Tagset Baseline

In the second set of experiments we trained baseline monolingual Spanish and English taggers on the UD corpora: one monolingual Spanish and one monolingual English tagger, and one tagger trained on both corpora. The goal of these experiments was to obtain taggers trained on the Universal tagset that we could use to obtain a baseline performance of monolingual taggers on the CS Bangor corpus. The results are shown in Table 4.2. The accuracy of the baseline UD taggers is slightly worse than the WSJ taggers, probably due to the smaller size of the UD datasets. The accuracy of the taggers on their own test sets is 94.78% and 95.02% for English and Spanish respectively. In comparison, Stanford's neural dependency parser [Dozat *et al.*, 2017] reports accuracy values of 95.11% and 96.59% respectively.

In order to approximate how a monolingual tagger trained on established datasets performs on a conversational CS dataset, we tested the baseline UD taggers on the MB test set and observed a dramatic drop in accuracy, due perhaps to the difference in genre (web blog data vs. transcribed conversation) and the bilingual nature of the Miami corpus. Note that, when training on both EN and ES UD, the Bi-LSTM taggers reached 88.17% accuracy, from only 69.97 and 45.13% by the monolingual taggers. When looking at the multilingual subset of sentences from the test set (CS Test in Table 4.2), we observe that the English model decreases in accuracy further, whereas the Spanish tagger has better performance. This is due to the CS sentences having more Spanish than English.

Training	Task	Dev	Test	CS Test
MB	Tagger	96.27	96.34	96.10
	Tagger+LID	96.35	96.49	**96.44**
	Joint Tagger	96.30	96.39	95.97
MB + UD	Tagger	96.34	96.47	95.99
	Tagger+LID	**96.40**	**96.63**	**96.44**
	Joint Tagger	96.39	96.61	96.35
MB Inter-CS	Tagger	96.24	96.03	95.27
	Tagger+LID	96.26	96.16	95.55
	Joint Tagger	96.25	96.11	95.22

Table 4.3: POS tagging accuracy (%) on the MB corpus. Underlined font indicates best result in test set by each training setting across different tagging models. Bold results indicate best overall result in that test set.

4.7.3 Miami Bangor Results

In the third set of experiments we trained the three proposed models (Bi-LSTM tagger, Bi-LSTM tagger with LID features and joint part-of-speech and LID tagger) on: a) the full MB corpus, b) the joint MB and UD ES&EN corpora, and c) instances of inter-sentential CS utterances from the MB corpus. part-of-speech and LID accuracy results are shown in Table 4.3 and Table 4.4 respectively.

When training on the full MB corpus (top subtable from table 4.3), the part-of-speech tagger achieved 96.34% accuracy, a significant improvement from the 88.17% of the UD EN&ES. The improvement holds up on the subset of CS utterances, achieving 96.10% accuracy. Adding the LID features improved performance by 0.15 and 0.34 absolute percentage points. In both cases these differences are statistically significant ($p = 0.03$). Furthermore, when running joint part-of-speech and LID tagging, we see that tagging accuracy decreased slightly with respect to the part-of-speech tagger with LID features. This result reaffirms the contribution of the LID features. The difference in performance between the joint tagger and the basic tagger is slightly higher but not statistically significant ($p \sim 0.5$), showing that joint decoding does not harm overall performance. The best part-of-speech tagging

accuracy was always achieved by the Bi-LSTM tagger with LID features on both Test and CS Test; however, the joint Tagger was very close at no more than 0.1 percentage points on Test. When adding the UD corpora during training (middle subtable from Table 4.3) we see some improvements for the three models (0.13, 0.14 and 0.22 absolute percentage points respectively), and once again the difference in performance between the basic tagger and the tagger with LID features is statistically significant ($p < 0.05$).

We performed statistical tests to measure how different the models trained on MB were from the models trained on MB+UD and found that the addition of more monolingual data only made a difference for the joint tagger ($p < 0.01$) when looking at the performance on the Test set. On the CS test set, these models achieved about the same performance in part-of-speech tagging with a slight decrease for the basic tagger (-0.11 points, not significant) and a slight increase in accuracy for the joint tagger (0.38 percentage points, again not significant). Thus, it is clear that our model is able to learn from a few CS examples – even when many more monolingual sentences, from a different genre, are added to the train set.

Finally, we trained models on the subset of monolingual English and Spanish sentences from the MB training set to measure how a model trained on the same genre would be able to generalize to unseen intra-sentential CS sentences (bottom subtable from Table 4.3, marked as Inter-CS). This model would be closer to an in-genre inter-sentential CS tagger, tested on intra-sentential CS. Compared to the models trained on UD EN&ES, this model performed much better: 96.03% compared to 88.17% on the MB test set. This is mainly due to the fact that the UD corpus is *not* conversational speech. When comparing this result to the taggers trained on the full MB corpus, it can be seen that these new models achieved the lowest test accuracy across all models, probably due to the lack of bilingual examples in their training set. The difference in performance is more pronounced on the subset of CS utterances. Again, we ran statistical tests to compare these three new taggers to the taggers trained on the full MB corpus, and we found that their differences were statistically significant in all three cases ($p < 0.001$).

With respect to the LID accuracy of the joint Tagger, the best model is the one trained on the MB corpus, followed very closely by the model trained on MB and UD data. In both cases, the test set accuracy is above 98.49%. The accuracy on the CS test subset

Training	Dev	Test	CS Test
MB	98.82	98.78	98.01
MB + UD EN&ES	98.60	98.49	97.93
MB Inter-CS Subset	98.53	97.99	90.25

Table 4.4: LID tagging accuracy by the Bi-LSTM joint POS+LID Tagger on the MB corpus. Underlined results indicate best overall result in that test set.

is sightly lower at 98.01% and 97.93%. The monolingual Bangor tagger sees a decrease in test accuracy (97.99%) and a bigger drop, down to 90.25%, on the CS subset. All the differences in performance between every pair of the three LID taggers are statistically significant ($p < 10^{-5}$).

4.7.4 Comparison to Previous Work

We compare the performance of our models to the Integrated and Combined models proposed in [AlGhamdi *et al.*, 2016]. In that paper, part-of-speech tagging results are reported on the MB corpus, but using a preliminary mapping to the first iteration of the Universal tagset (12 tags, as opposed to the current 17); furthermore, the train and test splits were different. Therefore, we decided to replicate their experiments using their code and our data configuration, and compare them to our own classifiers.

With respect to their "Integrated" models, INT3:AllMonoData+CSD, which is their Tree Taggers trained on all available monolingual and code-switched data, can be compared to our Bi-LSMT POS Tagger trained on the full MB set and UD EN&ES. In this setting, our model outperformed the TreeTagger by more than 4 absolute points (ours at 96.47% compared to 92.33%). Similarly, their Tree Tagger trained exclusively on monolingual data (referred to as INT2:AllMono in their paper) performed significantly worse than our Bi-LSMT Tagger trained on the same data (ours at 88.17% compared to 84.47%). Finally, when trained exclusively on code-switched data, their model (INT1:CSD) underperformed compared to ours (92.71% versus 96.34%).

For their "Combined" models, COMB4:MonoLT-SVM trained two monolingual taggers on the UD-EN and UD-ES corpora and then a SVM on top from the output of the taggers

	EN	ES	EN&ES	Bangor
OOV	40.9	32.7	10.7	7.9
SAcc.	2.5	4.2	21.8	60.7
WAcc.	56.2	55.3	87.2	96.1
CSFAcc.	10.9	12.6	57.5	84.2
CSFWAcc.	12.6	16.1	63.3	86.7
AvgMinDistCSF	4.0	5.4	3.9	3.5
%ErrorsInCSF	26.9	24.3	32.5	36.9

Table 4.5: Out-of-vocabulary (OOV) rate, sentence (Sacc) and word accuracy (Wacc) at the sentence level, fragment (CSFAcc) and word accuracy (CSFWacc) at the fragment level, average minimum distance from tagging error to CSF (AvgMinDistCSF), and percentage of errors that occur within a CSF (%ErrorsInCSF).

on the MB corpus. We do not perform system combination in this Chapter, but in terms of data, this model would be most similar to our part-of-speech tagger trained on Miami and EN&ES UD, in which we reached 96.47% compared to their 92.20%. Furthermore, we note that our joint POS+LID tagger also has better part-of-speech accuracy than its counterparts Integrated systems from [AlGhamdi *et al.*, 2016] in addition to performing LID tagging.

4.8 Error Analysis

In this section we aim to analyze the performance of the POS taggers on the CS sentences of the Bangor test set and more specifically, on the CS fragments (CSF) of those test sentences. We define a CSF as the minimum contiguous span of words where a CS occurs. Most often a CSF will be two words long, spanning a Spanish token and an English one or vice versa, but it is also possible for fragments to be longer than that, given that a Mixed or Ambiguous token can occur within a fragment. The average CSF length in the Bangor test set is 2.16. We compared the performance of the UD-EN, UD-ES, UD-EN&ES and the Bangor-trained taggers on the Bangor CS Test set to understand the difference in errors made by monolingual and CS taggers. Table 4.5 shows the following measures: OOV rate, part-of-

speech tagging accuracy at the sentence and word level, part-of-speech tagging accuracy in CS fragments at the fragment and word level, the average distance from a part-of-speech tagging error to the nearest CSF (AvgMinDistCSF) and the percentage of part-of-speech tagging errors that occur within the boundaries of a CS utterance (%ErrorsInCSF). All measures were computed on the CS subset of test sentences of the Bangor corpus using the basic part-of-speech taggers trained on UD-EN, UD-ES, UD EN&ES and the Bangor corpus. In the table, we see that the multilingual models have much lower OOV rates, which translates into much higher sentence-level and word-level part-of-speech tagging accuracy. The CS Bangor-trained model fares much better than the UD EN&ES model in terms of word-level accuracy (96.1 versus 87.2%), especially when looking at the sentence-level accuracy (60.7 versus 21.8%), because the Bangor model is able to deal with code-switches. When looking at the tagging accuracy on the CS utterances the relative gains at the word level are even larger. This demonstrates that training on CS sentences is an important factor to achieving high-performing part-of-speech tagging accuracy.

It can also be seen from the table that, as the models achieve CS tagging accuracy, tagging errors are still concentrated near or within CSFs – for the UD EN&ES and Bangor models, AvgMinDistCSF and %ErrorsInCS decrease as the CSF-level accuracies increase. This shows that, even as the models improve at tagging CS fragments, CS fragments still remain the most challenging aspect of the task.

4.9 Conclusions

In this Chapter, we have presented a neural model for part-of-speech tagging and language identification on CS data. The neural network is a state-of-the-art bidirectional LSTM with prefix, suffix and word embeddings and four boolean features. We used the Miami Bangor corpus to train and test models and showed that: a) monolingual taggers trained on benchmark training sets perform poorly on the test set of the CS corpus; b) our CS models achieve high part-of-speech accuracy when trained and tested on CS sentences; c) expanding the feature set to include language ID as input features yielded the best performing models; d) a joint POS and language ID tagger performs comparably to the part-of-speech tagger

and its LID accuracy is higher than 98%, and e) a model trained on instances of in-genre inter-sentential CS performs much better than the monolingual baselines, but yielded worse test results than the model trained on instances of inter-sentential and intra-sentential code-switching. Furthermore, we compared our results to the previous state-of-the-art part-of-speech tagger for this corpus and showed that our classifiers outperform them in every configuration.

Chapter 5

Lexical, Syntactical and Conversational Factors in Code-Switching

5.1 Introduction

One of the often asked but unresolved questions regarding code-switching is whether there are particular conditions that facilitate or "trigger" its occurrence. Some linguistics literature on code-switching has proposed that a) cognates, defined as words in two different languages with the same etymology and similar spelling and meaning, are more likely to precede a code-switch; and that b) there are syntactic constraints to code-switching. However, there has been little research validating these proposals empirically.

Obtaining definitive statistical proof as to what factors elicit or trigger code-switching is not only important to further understand the dynamics behind language switching, but will also help us better model code-switching in language and speech technologies. Expert knowledge of linguistic factors to code-switching could have enormous impact in language modeling, word-level language identification, code-switching prediction, speech synthesis, speech recognition and dialog systems.

As with many aspects to code-switching, there are two main challenges to finding defini-

tive statistical proof of which factors influence code-switching. The first one is that code-switching seems to be heavily dependent on the two languages at hand, specifically lexical similarity and syntactical compatibility. The second one is the lack of a large enough corpus with enough natural code-switching in it. There is a wealth of studies in linguistics that have looked into factors that influence code-switching, but all of them have lacked, in our opinion, a corpus large enough to draw strong statistical proof.

In this chapter, we test the following hypotheses proposed in linguistic code-switched literature: first, that cognate stimuli are directly correlated with code-switching; second, that syntactic information facilitates or inhibits code-switching; and third that speakers *entrain* to one another in code-switching in conversation between bilinguals. We use statistical significance tests on the Miami Bangor corpus of code-switched English-Spanish conversation, and find that a) there is strong statistical evidence that cognates and switches occur simultaneously in the same utterance and that cognates facilitate switching when they precede a code-switch; b) there is strong statistical evidence of the relationship between part-of-speech tags and code-switching; and c) speakers tend to show converging entrainment behavior with respect to their rate of code-switching in conversation.

The remainder of the Chapter is organized as follows. Section 5.2 describes previous work on the relationship between cognate words, part-of-speech tags, entrainment, and code-switching. In Section 5.3, we describe the list of English-Spanish cognate words that we collected from the Internet. Section 5.4 describes the analysis of cognate influence on code-switching. Section 5.5 discusses the role of part-of-speech tags in code-switching. Section 5.6 discusses entrainment in the Miami Bangor corpus. Finally, Section 5.7 presents our conclusions and plans for future research.

5.2 Related Work

On the topic of eliciting code-switching, Michael Clyne proposed his triggering hypothesis which has been reformulated during the years [Clyne, 1967, 1980, 2003]. This hypothesis claims that code-switching can be facilitated by words that exist in both languages with similar form and meaning if those words occur immediately preceding or immediately

following a code-switch. Those words are said to include lexical transfers, bilingual homophones and proper nouns. Clyne's triggering hypothesis states that trigger words facilitate code-switching but does not imply direct causality, since it has also been observed that syntactic, prosodic and sociolinguistic factors also play a role. Broersma and Bot [Broersma and De Bot, 2006] evaluated this triggering hypothesis on a corpus of Dutch-Moroccan Arabic transcribed conversations and proposed alternative hypotheses based on modern speech production models. Although they were able to confirm and reject some aspects of Clyne's hypothesis, the corpus used in their analysis is severely limited by its size: 3 speakers, 318 clauses, 1,723 words, of which 60 include instances of code-switching.

In this Chapter, we test the triggering hypothesis for code-switching on a much larger corpus of English-Spanish speech following the methodology proposed in [Broersma and De Bot, 2006]. Our findings confirm some aspects of the hypothesis with much higher statistical power than Broersma and Bot's findings [Broersma and De Bot, 2006].

On the topic of syntax and code-switching, there has been much research mainly focusing on the study of how multiple monolingual grammars interact to produce mixed speech [Woolford, 1983] and whether they work together in a symmetric relationship [Sankoff and Poplack, 1981] or whether one (embedded) language is subsumed by the other (matrix) language [Joshi, 1982; Myers-Scotton, 1997]. Part-of-speech tags have played a role in many of these theories, typically being used to identify constraints that researchers have observed in their data. In this Chapter, we test the significance of the statistical relationship between code-switching and part-of-speech tags and inspect the role of different part-of-speech tags in the triggering process. Another contribution of this Chapter is an analysis of speaker entrainment on the code-switching rate we observe in the Miami Bangor corpus. While Fricke and Kootstra [2016] have investigated lexical priming in entrainment, no research has been done on longitudinal entrainment and code-switching.

5.3 Data

We use the Miami Bangor corpus (Section 3.3) for the statistical analysis carried out in this Chapter. We use the following naming convention throughout the rest of the Chapter: code-

switched word is the first word where a change of language occurs, the word preceding a code-switch is the word that occurs immediately before a code-switched word. Similarly, the word following a code-switch is the word that occurs immediately afterwards. For example, in the sentence 'Mis papás were so happy to see you', 'were' is the code-switched word and 'papás' and 'so' are the words immediately preceding and following the code-switch respectively.

A list of English-Spanish cognate pairs were collected from a variety of online sources[1]. We pre-processed the list of cognates first automatically and then manually to remove determiners, break cognate compound words into single words and remove duplicates. Not counting masculine/feminine duplicates, a total of 3,432 cognate word pairs were collected, of which 1,305 appear on the Miami Bangor corpus. This corpus can be obtained from GitHub[2].

5.4 Code-Switching and Cognate Words

In this section we analyze the statistical relationship between code-switching and cognate words on the Miami Bangor corpus, testing the triggering hypothesis. All the tests are performed on the surface-level form of the cognates words, and multiple word senses are not accounted for.

First, we observe that there is a strong statistical relationship between code-switched utterances and the presence of cognates in those utterances: Table 5.1 shows the contingency table for all the utterances in the corpus split in utterances with and without cognates and monolingual and code-switched utterances. The results of a χ^2 test returns a highly significant p-value that rejects the hypothesis that both distributions are independent. The percentage of code-switched utterances in each group (last row of the table) confirms that utterances that contain a cognate are more likely to be in code-switched utterances than in monolingual utterances.

[1] http://nlp.cs.berkeley.edu/projects/historical.shtml; http://spanishcognates.org/; http://www.colorincolorado.org/sites/default/files/Cognatelist.pdf; https://www.duolingo.com/comment/5508808/The-Most-Useful-Spanish-Cognates

[2] https://github.com/vsoto/cognates_en_es

$\chi^2 = 309.63$		Cognate	
$p < 10^{-68}$		no	yes
CS	no	20,029	18,767
	yes	1,037	1,937
	% yes	4.92	9.36

Table 5.1: Number of code-switched and monolingual utterances split by utterances that contain a cognate or not.

$\chi^2 = 0.14$		Cognate	
$p = 0.71$		No bordering	Precedes
CS	no	206,005	28,901
	yes	3,256	466
	% yes	1.56	1.59

Table 5.2: Number of code-switched words and percentage of code-switched words split by words preceding a cognate and words not bordering cognates.

Next, we replicated the experiments from [Broersma and De Bot, 2006] in Tables 5.2, 5.3, 5.4, 5.5 and 5.6. For all tables, we present contingency tables for the two groups being compared (one always code-switched words and the other some aspect of immediately adjacent cognates), plus the percentage of code-switched words for the second group, and the results of a χ^2 test on the contingency table, including the test's statistic value (χ^2) and its p-value p. Table 5.2 shows that there is no significant statistical relationship between words that precede a cognate and code-switching when compared to words that do not border on cognates.

Table 5.3 shows that there **is** a strong statistical relationship between code-switched words and words that follow cognates, when compared to words that do not border on cognates. Furthermore, it can be seen that the percentage of CS words increases for the group of words that immediately follow cognates.

A variation of the same test, Table 5.4, shows that there is a strong statistical relation-

$\chi^2 = 26.55$ $p < 10^{-6}$		Cognate	
		No bordering	Follows
CS	no	206,005	26,812
	yes	3,256	540
	% yes	1.56	1.97

Table 5.3: Number of code-switched words and percentage of code-switched words split by words following a cognate and words not bordering cognates.

$\chi^2 = 26.63$ $p < 10^{-6}$		Follows a Cognate	
		no	yes
CS	no	230,768	26,812
	yes	3,653	540
	% yes	1.56	1.97

Table 5.4: Number of code-switched words and percentage of code-switched words split by words following and not following a cognate.

ship between code-switching and words that follow cognates when compared to words that do not follow cognates. Ignoring the restriction that words are not followed by cognates, the result of the test is the same, which suggests that cognates that follow code-switching have no effect on them. This is further confirmed in Table 5.5, which shows that there is no statistical relationship between code-switching and the disjoint sets of words that border on cognates and words that only follow cognates.

From these experiments we can confidently conclude that cognates immediately preceding a code-switch help facilitate the switch and cognates immediately following a code-switch do not have a meaningful impact on it. Furthermore from Table 5.5 we conclude that code-switching does not occur significantly more often when words are immediately preceded and followed by cognates. Overall, it can be observed that the same results obtained for Dutch-Moroccan Arabic in [Broersma and De Bot, 2006] translate to the English-Spanish Miami Bangor corpus with much higher statistical power. We also examined the statistical

$\chi^2 = 2.67$		Cognate	
$p = 0.1$		Follows	Bordering
CS	no	22,674	4,138
	yes	471	69
	% yes	2.03	1.64

Table 5.5: Number of code-switched words and percentage of code-switched words split by words that border on two cognates and words that only follow a cognate word.

$\chi^2 = 26.23$		Cognate	
$p < 10^{-6}$		no	yes
CS	no	222,703	34,877
	yes	3,740	453
	% yes	1.65	1.28

Table 5.6: Number of code-switched words and percentage of code-switched words split by cognate and non cognate words.

relationship between code-switched words being cognate words (Table 5.6) and found that there is a strong statistical relationship between both variables. However, surprisingly, we found that code-switched words are overall less likely to be cognates than other words.

5.5 Code-Switching and Part-of-Speech Tags

The second set of experiments examines the relationship between code-switching and part-of-speech tags. Here we examine the role that part-of-speech categories play when immediately preceding and following a code-switch, and when they are themselves a code-switch. We started by measuring the statistical relationship between the tagset and the code-switched words. In order to do so, we creates three contingency tables for the counts of all part-of-speech tags and whether they occur in one of the mentioned positions, and ran a χ^2 test on them. Results for the three tests are shown in Table 5.7. It can be observed that, in the three cases, the null hypothesis that the part-of-speech tag distribution and the

	POS		
	Preceding	Current	Following
χ^2	1,817.8	795.0	35.39
p-value	0.0	$< 10^{-158}$	< 0.01

Table 5.7: Statistical significance results of performing the χ^2 test of all part-of-speech tags in three pairs of groups: words preceding a code-switch, code-switched words, and words following a code-switching.

code-switching distribution are independent can be rejected. Specifically, part-of-speech tags seem to have a statistically strong relationship to the words preceding a code-switch and the code-switched words themselves.

	ADJ	ADP	ADV	AUX	CONJ	DET	INTJ	NOUN	NUM	PART	PRON	PROPN	SCONJ	VERB
POS (%)	4.1	6.97	8.11	3.25	4.4	8.81	5.94	11.04	1.51	2.58	15.98	2.49	3.88	20.00
POS(t-1)|CS(t) (%)	3.84	5.08	7.66	**0**.33	5.29	**13**.90	**9**.23	**18**.89	**0**.79	**0**.60	**4**.46	**4**.41	**6**.01	17.72
POS(t)|CS(t) (%)	5.03	7.23	7.51	2.12	4.89	7.27	5.13	**21**.23	1.48	**0**.38	17.10	2.89	**6**.32	**11**.42
POS(t+1)|CS(t) (%)	4.17	6.78	6.98	3.28	**2**.59	10.09	2.27	**14**.71	1.67	2.21	15.66	2.93	3.59	22.21
CS(t)|POS(t-1) (%)	2.37	1.21	1.86	**0**.17	1.99	2.57	**3**.94	**4**.44	1.01	**0**.38	**0**.50	**4**.36	2.57	1.63
CS(t)|POS(t) (%)	1.97	1.66	1.48	1.05	1.78	1.32	1.38	**3**.08	1.57	**0**.24	1.71	1.85	**2**.61	**0**.91
CS(t)|POS(t+1) (%)	1.43	1.39	1.34	1.52	1.82	1.65	1.41	1.81	1.61	1.16	1.62	1.78	1.58	1.61
CS(t), POS(t-1)	✓	✓✓		✓✓✓		✓✓	✓✓✓	✓✓✓	✓	✓✓	✓✓✓	✓✓✓	✓✓	
CS(t), POS(t)	✓			✓✓		✓✓		✓✓✓		✓✓			✓✓	✓✓✓
CS(t), POS(t+1)			✓					✓		✓✓				

Table 5.8: The First subtable shows the percentage of part-of-speech tags in the Miami Bangor corpus. The Second shows the % part-of-speech preceding, on, and following a code-switched word. The Third shows the percentage of words that are code-switched for each part-of-speech tag category preceding, on or following code-switched words. The Bottom subtable shows the significance of running χ^2 statistical tests on each group of part-of-speech tag and code-switched words. One check mark indicates $p < 0.01$, two indicate $p < 1 \times 10^{-4}$ and three indicate $p < 1 \times 10^{-18}$.

In order to study the role that specific tags play in eliciting code-switching, we started by comparing the tagging distribution over the whole corpus (top subtable on Table 5.8) with the tagging distribution of the words neighboring a code-switch (second subtable on Table 5.8). Some things are immediately clear: auxiliary verbs are very unlikely to precede code-switching; determiners and interjections are very likely to precede a code-switch; nouns appear more frequently as code-switched or neighboring a code-switch than on the rest of the corpus; particles are unsurprisingly not involved in code-switching; pronouns very rarely precede a code-switched word; and verbs are less likely to be code-switched.

We also studied which tags are more likely to precede, occur in, or follow a code-switch by examining the rows from the third subtable in Table 5.8. It can be observed that Proper Nouns, Nouns and Interjections are the tags most likely to trigger a code-switch; Nouns and Subordinating Conjunctions are the two categories that are more often switched. Moreover, we observe that the tags following a code-switch are all comparably likely to be switched (third row).

To end this section, we examined the statistical relationship between specific tags and code-switching by running a χ^2 test on the contingency tables populated by the counts of specific tags when preceding, occurring in, or following a code-switch. These results are shown in the bottom subtable of Table 5.8, where ✓indicates that the p-value of the statistical test is significant. The remaining (empty) cells have p-values larger than 0.01. The first observation we make from the first row of the subtable is that most of the part-of-speech tags have a strong statistical relationship when preceding a code-switch (first row), whether because they precede a code-switch more often (DET, INTJ, NOUN, PROPN, SCONJ) or less often (ADJ, ADP, AUX, NUM, PART, PRON). With respect to the code-switched words themselves, the second row shows that ADJ, NOUN and SCONJ significantly increase their presence in code-switching compared to AUX, DET, PART and VERB.

Some of these results might be expected: a code-switch between an auxiliary verb and a verb would be highly disruptive. Similarly, a switch is not likely to occur right after a pronoun, since most often pronouns are followed by verbs that need to agree on person and number. Indeed, the statistical relationship between verbs and code-switched words is very strong; the percentage of verbs that are switched is much smaller than the overall

percentage of verbs in the corpus.

Both pronouns and nouns have a strong relationship with code-switching when immediately preceding the switch, albeit in different ways. Whereas nouns are very likely to precede a switch (18.89% of the tokens preceding a switch in the corpus are nouns), pronouns are much less likely to occur before a switch than in general (4.46% of the words before a switch are pronouns, compared to their percentage of 15.98% throughout the corpus). This fact is counter-intuitive since pronouns substitute for nouns and noun phrases and both must agree with following verbs in person and number. So, it is not immediately clear why they behave so differently with respect to code-switching. However this finding agrees with previous research on pronoun-verb code-switching [Woolford, 1983] that states that even though most often such switches are banned [Barkin and Rivas, 1979] they can still occur [Sankoff and Poplack, 1981], depending, among other things on the length of the noun phrase they represent.

Another unexpected observation comes from the disparity between coordinating and subordinating conjunctions. We observe from the second subtable that the fraction of subordinating conjunctions that appear preceding or on a code-switch is higher than the number in the corpus as a whole, and, while the same can be said about coordinating conjunctions, the increase is not significant. Indeed conjunctions seem to be the ideal place to facilitate a switch, since they can often start a new sentence. We hypothesize that the reason for this difference is that the "and/y" coordinating conjunctions, which make up the majority of that tag category, are most often used for pairing objects in which case the switch could be disruptive.

5.6 Code-Switching and Entrainment

In this section, we analyze the Miami Bangor corpus for evidence of entrainment in code-switching between conversational partners throughout the conversation. Entrainment is the phenomenon of conversational partners becoming similar to each other in their behaviors in dialogue. It has been found to occur in multiple dimensions of spoken language, including acoustic-prosodic [Levitan *et al.*, 2012], linguistic style [Danescu-Niculescu-Mizil

et al., 2011], and syntactic structure [Reitter and Moore, 2006]. Importantly, entrainment has been associated with positive conversation outcomes, such as likability [Chartrand and Bargh, 1999], naturalness, and task success [Nenkova *et al.*, 2008]. To measure entrainment in code-switching, we measure *convergence* (becoming more similar over time) between the conversational partners in the frequency of their code-switching behavior.

Earlier work on priming in code-switching [Fricke and Kootstra, 2016] investigated structural priming effects as they relate to code-switching, also in the Miami Bangor corpus. They found that the probability of an utterance featuring code-switching was higher when the previous utterance contained a code-switch. We further analyzed the Miami Bangor corpus for evidence of entrainment in code-switching behavior beyond utterance-to-utterance priming. We measured convergence between the conversational partners frequency of code-switching, and the degree to which the **amount** of code switched segments produced by each speaker became more like that of their partner through the conversation as a whole. In total, we analyzed 37 conversations from the Miami Bangor corpus, excluding those with more than 2 speakers, conversations for which we only have the dialogue of 1 speaker, and conversations lacking code-switching entirely.

Convergence was calculated by using a Pearson-R correlation analysis on each speakers code-switching ratio (total number of code-switches normalized by total number of tokens) for each speaker turn. A significant positive correlation is indicative of convergence, the pairs code switching frequency becomes more similar to each other, while a significant negative correlation is indicative of divergence, the pairs code-switching frequency becomes more different over the course of the conversation. Out of 37 pairs, 32 show significant correlations in convergence or divergence of code-switching ratio. A total of 28 conversations are converging, of which 10 were weakly converging ($0 < r < 0.5$), 7 were moderately converging ($0.5 \leq r < 0.7$), and 11 were strongly converging ($r \geq 0.7$). The other 4 conversations showed diverging patterns: 2 weakly diverging ($0 > r > -0.5$) pairs and 2 moderately diverging ($-0.7 < r \leq -0.5$) pairs.

We know from previous studies that the introduction of code-switching may immediately prime a code-switch in the following utterance. Here we find interlocutors adapting to each others code-switching rates over the course of a conversation. Fricke and Kootra [Fricke

and Kootstra, 2016] speculate that other factors must be prompting code-switching beyond mechanistic language priming due to the infrequency of code-switching in the Miami Bangor corpus. Based on our findings, we propose entrainment as one such high-level mechanism driving code-switching behavior.

5.7 Conclusions

In this Chapter, we have presented a thorough analysis of the relationship between code-switching and cognate words, and code-switching and part-of-speech tags for English and Spanish. We presented statistical evidence that there is a strong relationship between code-switching and part-of-speech tags, and we examined the specific tags that occur more and less frequently in the vicinity of a switch. We confirmed that cognate words facilitate code-switching when immediately preceding the code-switch, but have no effect on it when they immediately follow the switch. Finally, we have demonstrated that speakers entrain to one another in the rate at which they code-switch, a finding that may provide further socio-linguistic insight into the social aspects of code-switching.

Chapter 6

Improving Code-Switched Language Modeling Using Cognate Features

6.1 Introduction

In the NLP field, there has been some prior research on the task of language modeling for code-switched data, often using manually labeled language identification and syntactic information. These features, while useful, are difficult to obtain both in terms of expense and in the difficulty of training annotators.

Current state-of-the-art models in NLP, like bi-directional LSTMs, encoder-decoders, and transformers ingest huge amounts of data in order to be able to obtain optimal configurations for their parameters. For the past few years, the trend in NLP has become to avoid explicit feature engineering in favor of using character, word or even sentence embeddings to obtain continuous representations of language units and then using deep neural networks on top of these embeddings to perform unsupervised feature extraction. In the context of code-switching, there are two reasons to avoid this strategy. The first one is that these models shed no light on the nature of code-switching and the task at hand. The second one is that, for code-switching, the amount of data needed to effectively train these systems is

often unobtainable.

In this Chapter, we focus on how information about cognate words can improve language modeling performance of code-switched English-Spanish (EN-ES) language. We have found that the degree of semantic, phonetic or lexical overlap between a pair of cognate words is a useful feature in identifying code-switching in language. We derive a set of orthographic, phonetic and semantic features from a list of of EN-ES cognates and run experiments on a corpus of conversational code-switched EN-ES. First, we show that there exists a strong statistical relationship between these cognate-based features and code-switching in the corpus. Secondly, we demonstrate that language models using these features obtain similar performance improvements to manually tagged features including language and part-of-speech tags. We conclude that cognate features can be a useful set of automatically-derived features that can be easily obtained for any pair of languages. Better LMs for code-switched data can thus be developed without the need for large amounts of manually-labeled training data, thus leading to improvements in speech and language processing of code-switching in many more language pairs.

The remainder of the Chapter is organized as follows. Section 6.2 describes previous work on language modeling for code-switched language. Section 6.4 outlines the cognate-based features we are proposing. Section 6.5 gives a short introduction to the Factored Language Model (FLM) approach we are using for our experiments. Sections 6.6 and 6.7 describes our experiments. Finally, Section 6.8 presents our conclusions and plans for future research.

6.2 Related Work

Work on computational approaches to modeling code-switching has been increasing in the last few years. Most efforts have focused on language identification and code-switching detection [Solorio *et al.*, 2014; Molina *et al.*, 2016], but there has also been some research on language modeling [Li and Fung, 2012; Adel *et al.*, 2013b; Li and Fung, 2014], part-of-speech tagging [Solorio and Liu, 2008b; Rodrigues and Kübler, 2013; Jamatia *et al.*, 2015; AlGhamdi *et al.*, 2016] and even speech recognition [Ahmed and Tan, 2012; Lyudovyk and

Pylypenko, 2014]. While some of this research has attempted to incorporate existing linguistic theories of code-switching [Li and Fung, 2012, 2014], the vast majority have focused on standard machine learning approaches. Ultimately, even if some of these models successfully solve the task they are trained for, they shed little insight on the intrinsic mechanics of code-switching and why and how it takes place.

In the last decade there has been increasing interest in tackling the problem of modeling code-switched language in the computational linguistics community. Most efforts have focused on applying machine learning methods to the task of language modeling. The first example of a statistical language model (SLM) applied to code-switched data was presented in [Franco and Solorio, 2007]; the authors trained 2-,3-,4- and 5-grams on a very small corpus (unlabeled for language ID), obtaining perplexity values ranging from 49.40 to 50.95. Li and Fung [2012] is the first example of an SLM to incorporate a syntactical constraint ("the equivalence constraint" [Sankoff and Poplack, 1981] which states that "the order of constituents immediately adjacent to the code-switching point must be the same in both language's grammars") from the linguistics community. This work achieved a word error rate of 35.2% and 45.9% in two conversational speech corpora. In [Li and Fung, 2014] the same authors incorporated the Functional Head Constraint [Belazi *et al.*, 1994], which states that code-switching cannot occur between a functional head and its complement, and achieved further improvements in word error rates of 33.70% and 43.58% on the same corpora.

Adel *et al.* [2013b] performed LM experiments using FLMs and RNNs on the SEAME corpus of English and Mandarin code-switching. They found that RNNs achieved better results than FLMs and demonstrated that LID and part-of-speech tags are useful features for code-switched LM. However, their perplexity values were very high (239.21 for the best single model and 192.08 for the best combined model). In a similar vein as the work we present here, Adel *et al.* [2013a] presented an analysis that shows that certain words and part-of-speech tags are more likely to precede a code-switch; however their proposed RNN model for LM ended up using only POS classes and words as input, without any attempt to flag what *type* of words were useful.

Winata *et al.* [2018] proposed that a multi-task learning approach to POS tagging and

Split	# Sentences	# Tokens
Full	42.9K	321,630
Train	36,710	274,863
Dev	2,000	15,588
Test	4,200	31,179

Table 6.1: Partition of Miami Bangor corpus used for Language Modeling experiments. Table shows number of sentences and tokens in the full Miami Bangor Corpus and each of its splits.

LM can help improve LM performance and showed relative perplexity improvements of 9.7% on the SEAME corpus. Similarly, Chandu *et al.* [2018] achieved some improvements on the joint task of LID tagging and LM.

In this Chapter, we continue to investigate how cognate-based features, part-of-speech tags and LID tags effect code-switching, and specifically how they can help improve performance on the task of language modeling.

6.3 Data

In this research we use the Miami Bangor corpus detailed in Section 4.5 and the list of EN-ES cognate words from Section 5.3.

The Miami Bangor corpus was split into train, development and test sets for the experiments presented in Section 6.6. The size of each split is shown in Table 6.1.

6.4 Feature Engineering

6.4.1 Feature Extraction

We used the list $\mathcal{L} = \{(e^k, s^k)\}$ of English-Spanish pairs of cognate words described in [Soto *et al.*, 2018], which can be obtained from Github.[1]

[1] `https://github.com/vsoto/cognates_en_es`

Each entry in the list consists of an English word e^k and a Spanish word s^k of the same cognate (e.g. "mathematics" and "matemáticas"). The list contains a total of 3,423 cognate word pairs, of which a total of 1,305 appear at least once in the Miami Bangor corpus. For each of these word pairs (e^k, s^k), we extracted the following set of features, $f_l^k = f_l(e^k, s^k)$ quantifying the difference between cognate pairs in terms of orthography, pronunciation, and meaning:

Orthographic features: To compute these features we measured the distance or similarity between the sequence of letters of the pair of cognates. Distances used include the DamerauLevenshtein (DLD), Hamming (HD), and Levenshtein (LD) distances. We also computed the Jaro (JS) and Jaro-Winkler (JWS) similarities. We also included a 'perfect cognate' feature which is 1 if the spelling is identical in Spanish and English (not accounting for tildes) and 0 otherwise. For example, for the cognate pairs "mathematics" and "matemáticas", the Levenshtein distance is 0.18, and "actor" is a perfect cognate.

Pronunciation features: These features reflect how different the pronunciation between a pair of words is. We obtained the pronunciations from the CMU English pronunciation dictionary and a Spanish pronunciation dictionary. For the pronunciation entries of words not found in these dictionaries, we trained grapheme-to-phoneme systems using the CMU Sphinx sequence-to-sequence system described in [Yao and Zweig, 2015]. Once all the pronunciations were obtained, we computed the distance between both pronunciations using the Binary (BD), Hamming (HD), Jaccard (JD) and Levenshtein (LD) distances.

Semantic features: These features are intended to reflect how close in meaning the two words in each cognate pair are. We used the MUSE bilingual word embeddings [Conneau *et al.*, 2018] and computed the Euclidean (EUC) distance and the Cosine (COS) Similarity between the cognate pairs. Only 15 cognate words that appeared in the Miami Bangor corpus were not covered by the bilingual embeddings.

6.4.2 Feature Normalization

All the features not naturally bounded to $[0, 1]$ were normalized by the feature's maximum possible value, which for most distances is the maximum sequence length of one of the cognates in the pair. All the distance features were transformed into similarities using a

simple transformation $sim = 1 - dist$.

6.4.3 Statistical Relationship between Code-switching and Cognate Features

To analyze the relationship between code-switching and the cognate-based features, and to determine if these features can be predictive of code-switching behavior, we first looked at how similar the distribution of the features is when looking at the words surrounding a code-switch compared to those in the rest of the utterance.

To do so, we ran the Kruskal-Wallis statistical test to compare the distribution of features with respect to their position relative to a (labeled) code-switch. Kruskal-Wallis tests the null hypothesis that the population medians for two or more groups are equal, which can be rejected with a sufficiently small p-value. If the distributions (medians) of two subgroups of feature values are different enough, these features will be potentially usable for code-switch detection and language modeling.

To run the statistical significance tests we assign feature values f_l to every word w_i in an utterance: If the word w_i is a cognate (e^k, s^k) present in the list of cognates, the word is given the feature value $f_l(w_i) = f_l(e^k, s^k)$; otherwise, the word is assigned the minimum possible value for that feature, which is zero. For example, for the phrase "very simpático", where "very" is not a cognate and "simpático" is a cognate in the list, we would assign a zero to the first word and the pertinent feature value to the second word. We ran this statistical test for each feature described in Section 6.4.1 and in three different modalities to compare the feature distributions of a) code-switched words and the rest of words in an utterance; b) words that immediately precede a code-switch and the rest of words in an utterance and c) words that immediately follow a code-switch and the rest of the words in the utterance.

Results of these tests are presented in Table 6.2. In this table, each row contains the results from the Kruskal-Wallis test for a given feature and each column specifies the distributions that are being compared. Column 3 compares the feature distributions of the words immediately preceding a code-switch and the rest of the words in the corpus. Column 4 compares the feature distributions of the code-switched words and the rest of the words

in the corpus; and column 5 compares the feature distributions of the words immediately following a code-switch and the rest of the words in the corpus. Check marks indicate p-values $p < 0.001$.

Following the same trend that we observed in [Soto *et al.*, 2018], the p-values confirm that all engineered features values have different median values when they **precede** a code-switch and when they **are** code-switched; however, they do not present statistical differences when they immediately **follow** a code-switch.

For the orthographic features, all show significantly different distributions when the word they are calculated from precedes ($10^{-19} < p < 10^{-15}$) or is itself a code-switch ($10^{-20} < p < 10^{-8}$). Similarly for the pronunciation features, p-values range from $10^{-22} < p < 10^{-10}$ for feature values of code-switched words and $10^{-18} < p < 10^{-15}$ for feature values for words immediately preceding a code-switch (Hamming and Levenshtein distance). Tests run on semantic features return smaller p-values when focused on words preceding a switch ($10^{-7} < p < 10^{-4}$) but similar power on code-switched words ($10^{-22} < p < 10^{-20}$). Overall, the largest differences were always found on the code-switched word (perfect spelling, binary distance on pronunciation entries and cosine similarity on word embeddings).

6.5 Factored Language Models

Factored Language Models (FLMs) [Bilmes and Kirchhoff, 2003] are language models that encode each word w_i in a sentence as a vector of k factors $w_i = (f_i^1, \ldots, f_i^k) = f_t^{1:k} = F_i$, where each factor can be a feature of the word, i.e. the language of the word or its part-of-speech tag. An FLM is a directed graphical model where $p(F_t|F_{t-l}, \ldots, F_{t-1})$ can be factored into probabilities of the form $p(f|f_1, \ldots, f_N)$. An FLM is described by its backoff graph, which shows the various backoff paths from the parent node $p(F|F_1, \ldots, F_N)$ to the child node $P(F)$. Given a chosen backoff graph topology, FLMs can be trained using the Generalized Parallel Backoff algorithm, which allows the language model to back off on a single path or on multiple parallel paths simultaneously during runtime.

For the experiments presented in this Section, we used the FLM implementation in the SRILM toolkit [Stolcke, 2002; Stolcke *et al.*, 2011], which allows for fast training and

Group	Feat.	Prec	CS	After
Orthography	DL	✓	✓	-
	Hamming	✓	✓	-
	Jaro	✓	✓	-
	Jaro-Winkler	✓	✓	-
	Levenshtein	✓	✓	-
	Perfect	✓	✓	-
Pron.	Binary	-	✓	-
	Hamming	✓	✓	-
	Jaccard	-	✓	-
	Levenshtein	✓	✓	-
Semantic	Cosine	✓	✓	-
	Euclidean	✓	✓	-

Table 6.2: Statistical significance results of running the Kruskal-Wallis test by ranks of all the features split into two groups. Three pairs of groups are tested: words preceding a code-switch and the rest of words, code-switched words and the rest of words; and words following a code-switch and the rest. Check marks indicate that there is a statistically significant difference between the distribution of the features values of the two groups.

Model	PP
W	73.57
W + LID	68.88
W + POS	68.87
W + LID + POS	59.28

Table 6.3: Test set perplexity of Factored Language Models trained on word trigrams and language identifiers and part-of-speech tags.

evaluation of FLMs. Some of the key implementation issues when using FLMs are the choice of factors to use in the model and the design of the backoff graph. Many factors go into the design of the backoff graph, including: the topology of the graph (including the number of backoff graph nodes, and the dependencies between them) and the discounting, smoothing and combination options for each node. Given all these design factors, finding the optimal FLM structure for a given corpus is a highly intractable problem. We use GA-FLM [Duh and Kirchhoff, 2004], a genetic algorithm that searches over the space of possible FLMs structures optimizing for development set perplexity. Specifically, for each of FLMs trained on the next section, the GA-FLM was run on 10 generations, each one with a population size of 100, with a cross-over probability of 0.9 and a mutation probability of 0.01.

6.6 Experiments & Results

We started by training FLMs using exclusively word tokens and the gold features we have on the Miami Bangor corpus: LID and part-of-speech tags. All FLMs were trained using features of two previous words (3-grams). Table 6.3 shows the perplexity achieved by the baseline tri-gram language models and by the same language models when adding the gold LID and part-of-speech features to the Bangor Corpus. The addition of the LID and part-of-speech tags separately helped achieve similar improvements, from 73.57 down to 68.88 and 68.87 respectively. When used together the perplexity dropped much further to 59.28, proving that the two features are complementary and equally useful for language modeling.

Table 6.4 shows the performance of a trigram FLM when adding the cognate-based features. The top subtable shows the LM performance when adding the cognate and perfect cognate flags. In both cases perplexity improved with respect to the 73.57 baseline, but none of the features were as useful as LID or part-of-speech tags for LM. The next three subtables show the perplexity of the LM when adding just one of the orthographic, pronunciation, or semantic cognate-based features. For the lexical features, the best performance was achieved when using the Jaro-Winkler distance between the English and Spanish cognates (65.35); for the pronunciation features, the best performance was achieved when using the Hamming distance (65.99); and for the semantic features, both the cosine and euclidean distances performed similarly (66.02). Comparing tables 6.3 and 6.4, the cognate-based features can achieve better perplexity performance than the LID and part-of-speech tags features when used separately. This is important because LID and part-of-speech tags for this corpus were crowdsourced and expensive to obtain. However, no cognate-based feature helps achieve similar performance to the combination of the manual LID and part-of-speech tags.

Table 6.5 shows the perplexity performance of the FLM models when adding a combination of the cognate-based features. For each category (LEX, PRON and SEM) the best performing feature from Table 6.4 was chosen. The table shows that the combination of PRON+SEM, and the combination of LEX+PRON+SEM helps improve the perplexity achieved by the models shown in Table 6.4, although the gains are very small. We hypothesize that the combination of LEX and PRON features may not offer perplexity gains since the features are computed very similarly (the first as the string distance and the second as the distance between two phone sequences) whereas adding the SEM feature always helps improve performance. However, the addition of all cognate-based features does not bring performance improvements comparable to the addition of LID and part-of-speech tags (64.51 compared to 59.28).

We concluded these experiments by examining how much gain we could obtain from adding the cognate-based features to the LID and part-of-speech tags, which obtained a perplexity of 59.28 (see Table 6.3). We see that adding any subset of cognate features adds value to the W+L+P model, with perplexity numbers ranging from 58.30 to 59.17, although

Model	PP
W + Cognate	70.17
W + Perfect Cognates	71.71
W + LEX(JWS)	**65.35**
W + LEX(LD)	65.88
W + LEX(DLD)	66.02
W + LEX(JS)	67.02
W + LEX(HD)	72.01
W + PRON(JD)	66.42
W + PRON(HD)	**65.99**
W + PRON(BD)	70.14
W + PRON(LD)	66.42
W + SEM(EUC)	66.02
W + SEM(COS)	66.02

Table 6.4: Test set perplexity of Factored Language Models trained on word trigrams and each of the cognate-based features.

Model	PP
W + LEX + PRON	66.23
W + LEX + SEM	65.90
W + PRON + SEM	64.95
W + LEX + PRON + SEM	64.51

Table 6.5: Test set perplexity of Factored Language Models using a combination of two or the three cognate-based features.

Model	PP
W + C + L + P	58.85
W + C + L + P + PRON	58.32
W + C + L + P + SEM	58.32
W + C + L + P + LEX	58.75
W + C + L + P + LEX + PRON	58.30
W + C + L + P + LEX + SEM	59.17
W + C + L + P + PRON + SEM	60.01
W + C + L + P + LEX + PRON + SEM	58.84

Table 6.6: Test set perplexity of FLMs using cognate flags, LID and part-of-speech tags plus one set of one, two, or three cognate-based features.

these improvements are very small.

6.7 Cross-Lingual Feature Transfer

So far in this Chapter, we have studied the relationship between code-switching and cognate words and part-of-speech roles. We have also proved that cognate-based features offer significant performance improvements over gold-label features like part-of-speech tags or language identifiers. All these results have been obtained from experiments run on the Miami Bangor corpus of English-Spanish code-switched language. The next questions we want to tackle are: 1) How much of these findings hold from English-Spanish to a different language pair? and 2) Can we transfer cognate-based features across language pairs to obtain improvements on language modeling?

Code-switching between two languages seems to be heavily dependent on lexical overlap [Clyne, 1980; Broersma and De Bot, 2006] and syntactical compatibility [Woolford, 1983; Belazi *et al.*, 1994], and there seems to be consensus that there is not a single unified theory to the way two languages switch [Clyne, 1987], despite some efforts in that direction [Myers-Scotton and Jake, 2009]. While we certainly do not expect a pair of languages like English and Mandarin Chinese to behave as English and Spanish do, it would be reasonable

to expect that, for example, English and another Romance language behave similarly to English and Spanish.

In this research, we chose English and French as the next language pair to study and a) confirmed that cognate-based features are again a valuable addition to the task of language modeling; and b) confirmed that we can use cognate-based features computed for English-Spanish code-switching and for English-French and obtain similar language modeling improvements. We began by collecting a small dataset of code-switched English-French sentences from the Hansard corpus.

Then we showed that the statistical relationship between cognates and code-switching in English-French is similar to that of English-Spanish. We trained language models on code-switched language and showed that cognate-based features obtained from the Hansard offer language modeling performance gains. Finally, we used cross-lingual features for English-French obtained in English-Spanish.

6.7.1 Data Collection for English-French Code-Switched Sentences

The Hansard corpus is a collection of parallel documents in English and French of the proceedings of the Canadian parliament. The corpus is divided into three different sets of which only the first two clearly state which document is the source (and unedited) document from the parallel pair, and which is the translated document.

Despite being a corpus whose content is mostly very formal, the Hansard corpus does contain examples of code-switching [Carpuat, 2014], both on the section of the corpus composed of transcriptions drawn from meetings from the House of Commons, and on the section of the corpus composed of transcriptions of the committee meetings.

We used our code-switched sentence detection methods detailed in Chapter 2 to identify code-switching. The sizes of the strong and weak anchor wordlists are shown on Table 6.7 and the number of sentences retrieved by the Strong Anchoring, Weak Anchoring and Weak Anchoring + LID methods are shown on Table 6.8.

We used a combination of anchor-word and common-word sentence detection methods and obtained a total of 2,098 sentences, for which we crowd-sourced language tags at the word level. A subset of 811 sentences turned out to be code-switched. From the sentences

	EN	FR
Strong Anchors	32,257	53,031
Weak Anchors	58,693	68,692

Table 6.7: Size of the Strong Anchor and Weak Anchor wordlists for English and French.

Method	# Retrieved Sentences	Unique	CS Sentences	Precision
Common Words	800	764	172	21.5
Weak Anchoring	1,333	1,297	673	50.48
Joint	2,098	X	811	38.65

Table 6.8: Number of sentences retrieved from the Hansard corpus by Common Words and Weak Anchoring methods, along with the percentage of sentences that are code-switched.

retrieved using common words only 21.5% were code-switched, whereas 50.48% of the sentences retrieved using anchor words were code-switched. Note that both methods are very complementary, since only a total of 36 sentences were collected by both methods (1.71% of the total).

Given the 811 code-switched sentences, we aimed to augment the collected corpus to reflect a similar proportion of code-switched sentences to monolingual sentences as the Miami Bangor corpus. We selected 200 code-switched sentences for test purposes and added the rest of 611 sentences for training and validation. We added mono-lingual sentences in English and French from the source documents until we obtained a total of 8,800 sentences in the training set, including the 611 code-switched sentences (7% ratio).

6.7.2 Experiments

Following the methodology we used on the Miami Bangor corpus, we trained Factored Language Models on every subset of cognate-based features. On the left side of Table 6.9 we show the results of training and testing FLMs on Hansard using native features, and on the right subtable we show the results of training and testing FLMs on Hansard using feature values that were computed from English-Spanish cognate pairs.

Model	EN+FR FEATS PPL	EN+ES FEATS PPL
W	82.1	-
W + L	68.9	-
W + LEX	76.72	77.61
W + SEM	76.87	79.73
W + PRON	77.36	77.97
W + LEX + PRON	71.11	71.87
W + LEX + SEM	70.84	71.81
W + PRON + SEM	71.91	72.10
W + LEX + PRON + SEM	70.16	70.98

Table 6.9: Experimental results of training Factored Language Models on the subset of the Hansard corpus. The first column details the features used in each model, the second column shows the perplexity values obtained by each model when trained on cognate-based features from a list of English-French cognates. The third column shows the perplexity values obtained by each model when trained on cognate-based features from a list of English-Spanish cognates.

To assign English-Spanish cognate-features to English or French words we just follow the list of cognate truples. For example, given the word "mathmatiques" in a sentence of the Hansard corpus, there is an entry in the list of cognate truples for (mathmatiques, matemticas, mathematics), such that $f_C L(mathematiques) = f_E N{+}ES(matematicas, mathematics)$.

The Hansard models trained on EN+FR cognate features show similar results to the Bangor models which we analysed in Section 6.6. The model trained on word tri-grams and crowd-sourced language tags (W + L) achieved a 16% relative gain in perplexity. Unlike for the Bangor models, this gain was never matched by any of the cognate-based features. When adding the best performing Lexical (W + LEX), Pronunciation (W + PRON) and Semantic (W + SEM) features, we managed to reduce the perplexity down to 76.72, 76.87 and 77.37 perplexity, for an average relative gain of 6% with respect to the baseline model (W). Finally, when using a combination of cognate-based features we see much better gains

than those we observed on Bangor, achieving as much as 13.7% perplexity gain with the W + LEX + SEM model and 14.54% relative gain with the tri-gram model that includes every cognate-based feature.

When training Hansard models on EN+ES features, we see the same relative gains overall, although in terms of absolute gains, the EN+ES features always trail behind the EN+FR features. This seems to confirm the idea that cognate words, understood as a set of words across languages with the same etymological origin, can improve code-switched language modeling performance across language pairs precisely due to their multilingual nature.

6.8 Conclusions

In this Chapter, we proposed a new set of features extracted from lists of cognate words to improve code-switching detection. This set of features describes the semantic, orthographic and phonetic similarities across pairs of cognate words in English and Spanish. We first showed that there is a very high statistical relationship between these features and code-switching, which signals their potential usefulness for code-switched language modeling. We then showed that FLMs trained on these features achieve similar performance as FLMs trained on manually labeled features like LID and part-of-speech tags separately. The three feature sets (semantic, orthographic and phonetic) do not appear to be very complementary and underperform when compared to the joint use of LID and part-of-speech tags, however they are much simpler and less expensive to obtain.

Furthermore, we showed that cognate features can be used across language pairs. We used EN+ES cognate features on EN+FR Factored Language Models and showed that they help obtain perplexity improvements similar to those obtained by EN+FR features.

Chapter 7

Cross-Lingual Language Modeling Pre-Training for Code-Switching

7.1 Introduction

In the two previous chapters, we addressed the problem of modeling code-switched language by avoiding the use of black-box machine learning models. Instead, we proposed a set of interpretable features based on cognate words for code-switching and trained factored language models, which explicitly represent a hierarchy of features on their back-off graphs. In the last chapter of this thesis, we explore the opposite methodology: to use transfer learning to leverage large amounts of available monolingual data and parallel data to obtain pre-trained cross-lingual language models, and then expand and fine-tune these models on code-switched data to gauge the extent to which monolingual and parallel data can help in a code-switched context.

Intra-sentential code-switching is a relatively sparse phenomenon even in fully multilingual communities. For example the Miami Bangor corpus, one of the few conversational speech corpora with natural code-switching in it, only contains 7% of code-switched utterances. While collecting more code-switching data can be a successful strategy, as we have shown in Chapter 2, we should not ignore the vast amounts of monolingual data potentially available, both in- and out-of-domain. It is of key importance, therefore, to learn to leverage monolingual corpora as much as possible when developing code-switched models.

The main challenge when using monolingual data along with code-switched data to train statistical models is making sure that the monolingual examples do not overshadow the code-switched examples, and the model is able to identify code-switching information and learn from it. Because of the small ratio of intra-sentential code-switching to monolingual data, this is even harder when adding out-of-domain or out-of-genre monolingual corpora to the training recipe. Fortunately, the advent of deep learning has facilitated learning in this kind of setting by following a learning scheme called transfer learning. In transfer learning a (usually deep) model, or part of a model, is first pre-trained using a large collection of out-of-domain data, and then the pre-trained model is used as an unsupervised feature extractor and fine-tuned on in-domain data.

In this Chapter, we propose to use Transformer models to pre-train cross-lingual language models and fine-tune them on code-switched data for the downstream tasks of word-level language tagging and code-switched language modeling. The work presented in this chapter is ongoing, and we are planning to keep doing research on it. The rest of this chapter is organized as follows: Section 7.2 provdes a short background on the topics of recent sequence-to-sequence models and language modeling pre-training. Section 7.3 gives an overview of the data used in this chapter. Section 7.4 shows how our cross-lingual language models are pre-trained. Section 7.5 shows the results of fine-tuning the pre-trained models on code-switched language. Section 7.6 shows the experiments and results of fine-tuning our models on the task of word-level language identification.

7.2 Background

7.2.1 Sequence-to-Sequence Models and Transformers

Sequence-to-sequence (Seq2Seq) models [Sutskever *et al.*, 2014] are machine learning models that take an input in the form of a sequence of elements $\{x_i\}_1^n$ and output another sequence of elements $\{y_j\}_1^m$. Some recent Seq2Seq models like LSTM neural nets have been very successful at modeling long-term dependencies but are expensive to train and are only adept at tasks where the length of both input and output sequences are the same. The encoder-decoder architecture solved this problem by proposing a model with two parts: an

encoder that maps a sequence of elements into a sequence of source hidden states $\{h_s\}_1^n$ and a decoder that transforms the last source hidden state into an output sequence of variable length $\{y_j\}_1^m$. In an encoder-decoder model the encoder can be, for example, an LSTM or bi-LSTM, and the decoder can be another unidirectional LSTM also. The encoder takes an input element from the input sequence at every step along with its own hidden state and outputs a source hidden state. Similarly, the decoder takes the last source hidden state and a start-of-sequence element in the first decoding step and then takes its own hidden state and last prediction for the next steps until it predicts an end-of-sequence element.

Despite the enormous improvements achieved by Seq2Seq models and the Encoder-Decoder architecture, most of these models struggled with long sequences. The attention mechanism [Bahdanau *et al.*, 2015] was designed to model which elements of a sequence are more relevant at each step of the sequential task. Attention takes the sequence of all source hidden state h_s and a target hidden state h_t and computes an alignment vector as long as the sequence of source hidden states. This alignment vector a_t indicates how relevant each source hidden state h_s is to the current target hidden state h_t. A context vector c_t is then computed as a weighted average of the source hidden states and the weights in the attention vector. The context vector c_t and the hidden state h_t are then used as input for the decoder at time step t. The attention mechanism has obtained great improvements in sequence-to-sequence tasks like Machine Translation [Luong *et al.*, 2015], Summarization [Rush *et al.*, 2015] and Speech Recognition [Chorowski *et al.*, 2015].

Transformers [Vaswani *et al.*, 2017] are the latest evolution of encoder-decoder models with attention mechanisms. A Transformer is an encoder-decoder model that process every element in a sequence in relation to each other, instead of one by one. Transformers eliminate the need for convolutional or sequential (LSTM) layers by stacking Transformer modules on top of each other. Each Transformer module is formed by a self-attention layer followed by a fully-connected feed-forward layer, where each one of these layers has a residual connection to its output followed by an addition and normalization layer. The self-attention layer is most commonly a multi-head attention, which is a linear projection of the concatenation of several scaled dot-product attention modules. A scaled dot-product attention module is

defined by the following equation:

$$Attention(Q, K, V) = SoftMax(QK^T/\sqrt{d_k})V \tag{7.1}$$

where Q, K and V are the query (vector representation of one element in the sequence), keys (matrix containing the vector representation of every element in the sequence) and values (matrix representation of every element in the sequence).

Because Transformers do not process elements one by one sequentially, word embeddings are modified before being input into the Transformer layers by a process called positional embedding. This encoding process adds a position-dependent signal to each word embedding. A complete illustration of a Transformer architecture can be found in Figure 1 from the initial Transformer paper [Vaswani *et al.*, 2017].

7.2.2 Language Modeling Pre-training

Word embeddings, like word2vec [Mikolov *et al.*, 2013] or GloVe [Pennington *et al.*, 2014], have been critical to many of the improvements in NLP tasks in the past few years. Despite their critical success, word vectors have important weaknesses: they are shallow (a single vector of weights per word), and furthermore, they are context-free representations (they do not take into account the lexical and semantic context of the current word in the sentence). For embeddings to be able to better capture long-term dependencies and hierarchical relations they need to be more contextualized.

In the past few years, the trend in creating better and more contextualized word embeddings has been to pre-train deep language models on large language corpora, similar to the way Computer Vision has been using deep neural networks trained on ImageNet as feature extractors [Krizhevsky *et al.*, 2012]. These deep language models have been shown to create pre-trained models that work quite well once they are fine-tuned on a smaller datasets for downstream tasks like Machine Translation, Summarization, Sentence Similarity and other NLU tasks.

Below we give a short description of the five pre-trained language models that have made the biggest impact in the past two years:

7.2.2.1 ELMo: Embeddings from Language Models

ELMo [Peters *et al.*, 2018] proposes a deep model of L stacks of bi-LSTM layers. The unsupervised bi-directinal language model (biLM) is trained to minimize the negative log-likelihood in both directions. Once trained, ELMo embeddings are obtained by stacking all the hidden states across layers together and learning a task-specific linear combination of them. Similar to what is observed across the layers of deep models for computer vision, ELMo models show that deep layers better capture semantic information, while the first layers are better suited for syntactic tasks like POS tagging and word sense disambiguation.

7.2.2.2 ULMFit: Universal Language Model Fine-tuning

ULMFit [Howard and Ruder, 2018] proposes a multi-layer bi-LSTM network without attention. This was the first model to introduce the scheme of fine-tuning a pre-trained LM for a downstream task. It follows three steps to perform transfer learning:

- LM pre-training: on large language corpora.
- Target task LM fine-tuning: on target language corpora following two fine-tuning techniques: discriminative tuning (where each layer is applied using a different learning rate) or slanted triangular learning rates (a learning rate scheduling that first increases the learning rate for a short number of epochs and then slowly reduces the learning rate in a long stretch of epochs).
- Target task classification fine-tuning: the pre-trained LM is expanded with feed-forward layers and a softmax layer on top; concat pooling and gradual unfreezing are used to obtain the embedding representations and to avoid overriding all the information gained during pre-training.

7.2.2.3 OpenAI GPT: Generative Pre-Training Transformer

GPT [Radford *et al.*, 2018] innovates with respect to the previous models in several ways: a) it trains an unsupervised language model from large corpora and uses a Transformer decoder as a base model; b) it uses Byte Pair Encoding (BPE) subword units instead of

words; and c) it removes task-specific model architectures for the downstream tasks and instead proposes to fine-tune the Transformer model directly, for example, by adding a linear layer on top of the Transformer for a classification task.

7.2.2.4 BERT: Bidirectional Embedding Representations from Transformers

BERT [Devlin *et al.*, 2019] is similar to GPT with the novelty that the model trained is bidirectional: taking into account both left and right context. BERT achieves this bidirectionality by reformulating the language modeling task. Instead of training a model to predict the next word given the past context of words, BERT proposes the Masked Language Modeling task (MLM). During training, MLM masks 15% of each token in the corpus and: a) keeps the word the same with 0.1 probability; b) substitutes the word for another randomly chosen word with 0.1 probability or c) substitutes the word for a MASK token with 0.8 probability.

7.2.2.5 XLM: Cross-Lingual Language Modeling Pre-training

XLM [Conneau and Lample, 2019] expands the process from BERT by training models on the MLM task and a new cross-lingual task. This new cross-lingual language modeling task, called Translation Language Model (TLM) acts similarly to MLM and consists of concatenating two parallel sentences and randomly masking one element from each, asking the model to effectively guess each of the masked words from the two different languages. TLM encourages the Transformer model to learn alignments between both sentences, by re-setting the positional embeddings across the two sentences.

In this chapter, we explore the effects of using pre-trained language models and Transformers for code-switched language modeling and word-level language tagging. We choose to use the XLM library to perform pre-training and fine-tuning given their state-of-the-art capabilities on cross-lingual tasks, which should serve as a good starting point for code-switched language tasks.

Language	EN	ES	FR
Num. Sentences	43.8M	11.5M	15.8M
Num. Tokens	2.5B	640.9M	765.2M

Table 7.1: Wikipedia distribution of latest articles in English, Spanish and French. The table shows the amount of sentences and tokens.

Language Pair	EN-ES	EN-FR	ES-FR
Num. Sentence Pairs	11.4M	13.2M	11.4M
Num. Total Tokens	702.1M	799.2M	759.3M

Table 7.2: OPUS UN distribution of parallel sentences from UN transcripts in English-Spanish, English-French and Spanish-French. The table shows the amount of parallel sentence pairs and number of tokens.

7.3 Datasets & Pre-Processing

In this chapter we use the following sources of data. For code-switched data we use the Miami Bangor corpus of English and Spanish code-switching, and the subset of collected English and French utterances from the Hansard corpus from Section 6.7. For monolingual data we use the latest Wikipedia dumps in English, Spanish and French from `https://dumps.wikimedia.org/`. Table 7.1 shows the amount of data in each of the data dumps used in this study.

For parallel data, we use the English-French, English-Spanish and Spanish-French parallel documents from the OPUS MultiUN corpus [Eisele and Chen, 2010; Tiedemann, 2012]. Table 7.2 shows the amount of parallel data for each language pair.

The wikipedia, parallel, and code-switched text was cleaned of mark-up text, tokenized, and the accents in Spanish and English were removed. Each language dump from the Wikipedia dataset was split in validation, test and train partitions of 5,000 sentences, 5,000 sentences and the remainder of the dump respectively. The OPUS UN datasets were similarly split into test and validation sets of 5,000 sentence pairs and the rest of the sentence pairs were used for training.

Each model was first pre-trained on a joint dataset of Wikipedia and Parallel UN sentences. In total there were three different training dataset configurations (EN,ES,EN-ES, EN,FR,EN-FR, and EN,ES,FR,EN-ES,EN-FR) and therefore we learnt three different BPE code configurations for English-Spanish, English-French and English-Spanish-French. In each of the three cases the maximum amount of BPE codes was set to 15,000 codes, which means that the vocabulary size is effectively 15K. The BPE tokenization was then applied to the validation and test sets of the Wikipedia and UN corpora and also to the whole Miami Bangor and subset of Hansard corpora.

7.4 Pre-Training Cross-lingual Language Models

We followed the methodology from [Conneau and Lample, 2019] and pre-trained three XLM models EN-ES, EN-FR and EN-ES-FR with the following training data configurations: the EN-ES model was trained on English and Spanish Wikipedia and English-Spanish UN parallel transcripts; the EN-FR model was trained on English and French Wikipedia and English-French UN parallel transcripts; and the EN-ES-FR model was trained on English, French and Spanish Wikipedia and English-Spanish and English-French UN parallel transcripts.

Our pre-trained models were composed of a Transformer encoder with an embedding layer of 512 units and 12 stacked Transformer layers, where each layer has 8 self-attention multi-heads and 512 units. We used sinusoidal positional embeddings, but did not use language embeddings since one of the downstream tasks is language identification. The Transformer layer was pre-trained with a dropout value set to 0.1 on both the fully connected layers and the self-attention layers. All the models here were pre-trained using the MLM objective on monolingual data and the TLM objective on parallel data.

Table 7.3 shows the MLM and TLM perplexity of the pre-trained cross-lingual language models. All our models here are close to achieving the state of the art performance reported by [Conneau and Lample, 2019] by a small margin, but due to computational limitations we were not able to imitate their architecture with fully connected layers and self attention layers with 1,024 units due to the memory limitations in our GPUs.

Model	Validation		Test	
	MLM	TLM	MLM	TLM
EN-ES	22.70	4.20	23.17	4.26
EN-FR	21.84	4.88	21.52	4.99
EN-ES-FR	25.03	2.90	25.16	2.957

Table 7.3: Test perplexity by the pre-trained cross-lingual language models (XLM).

7.5 Fine-Tuning: Language Modeling

In this section, we study whether a pre-trained cross-lingual language model can be fine-tuned on code-switched language successfully. We first observe that the pre-trained language models from Table 7.3 perform very poorly on the subset of code-switched sentences from the Hansard and Miami Bangor corpora. The best performance by the pre-trained models is achieved on Bangor by the EN-ES model (190.57 MLM perplexity) and on Hansard by the EN-FR model (105.85 MLM perplexity). This performance is far from the MLM performance by the same models on their own validation and test sets, which ranged between 21 and 25 perplexity (Table 7.3).

We fine-tune the EN-ES, EN-FR and EN-ES-FR XLM pre-trained models on the Hansard corpus, the Miami Bangor corpus, and the combination of both, respectively. The fine-tuning process is performed until there is no MLM perplexity improvements on their respective validation sets. The fine-tuning does not implicate any change in the architecture of the XLM model: no additional layers are added to the model. Only the top two Transformer layers are fine-tuned and the rest of the layers are kept static.

Table 7.4 presents the MLM perplexity on the Miami Bangor corpus and Hansard corpus test sets. These are the same tests used for Language Modeling experiments in Chapter 6. However, the results reported here cannot be compared to the performance from the Factored Language Models, given that these perplexity values are from the Masked Language Modeling task, and not the standard task.

All three fine-tuned XLM models improve their perplexity numbers by more than 50%, which proves that fine-tuning is both needed and possible to adapt multilingual models to

Model	Before Fine-Tuning		After Fine-Tuning	
	Bangor	Hansard	Bangor	Hansard
EN-ES	190.57	291.80	70.28	109.86
EN-FR	483.76	105.85	120.23	72.65
EN-ES-FR	196.71	110.39	74.32	75.22

Table 7.4: Fine-tuned XLM code-switched language models performance. Performance is reported as MLM perplexity.

code-switched models. Specifically, the EN-ES model achieves the best perplexity result on Miami Bangor (70.28) and the EN-FR model achieves the best perplexity result on Hansard (72.65). The EN-ES-FR model comes close to both results but does not improve on them (74.32 and 75.22) respectively.

When testing the EN-ES and EN-FR model on a different language pair (Hansard and Bangor respectively), we observe very large relative perplexity improvements although the absolute performance (109.86 and 120.23) does not come close to the one achieved by their in-target counterparts (70.28 and 72.65).

Even though fine-tuning XLM models does obtain perplexity improvements over pre-trained models, these improvements do not seem to compare favorably with the performance of the Factored Language Models, which were simpler (in terms of number of parameters) and cheaper to train. We hypothesize that the reason for this is that we only performed fine-tuning on the top two layers, whereas [Peters *et al.*, 2018] showed that syntactic information, which is critical for code-switching is usually represented in the lower layers. For future work, we plan to experiment with more complex types of fine-tuning to obtain better code-switched language modeling performance.

7.6 Fine-Tuning: Word-Level Language Identification

We fine-tune the XLM models on the task of tagging each word with a language identifier. Since language tagging is a relatively simple task, there is no need to add a decoder on top of the pre-trained XLM model. Instead we just add a linear projector layer with softmax

Model	Corpora	Dev	CS Hansard	CS Bangor
Char2Vec	EN-ES	92.12	-	89.42
	EN-FR	88.54	85.43	-
XLM	EN-ES	98.42	82.30	97.11
	EN-FR	93.62	91.88	69.99
	EN-ES-FR	97.79	92.44	97.18

Table 7.5: Fine-tuned XLM language taggers performance. Results are reported as word-level language accuracy.

activations for the classification task. The language tagging task is explained in detail in 2.6.2. In this case though there are three main language tags EN, ES and FR for all three models.

Since we are interested in testing the EN-ES model on Hansard and the EN-FR model on Bangor, we add 1,000 sentences of monolingual French Wikipedia and 1,000 sentences of monolingual Spanish Wikipedia during the fine-tuning of those models so they can recognize French and Spanish words respectively.

Table 7.5 shows the results of fine-tuning the XLM models on their respective fine-tuning sets: Miami Bangor for EN-ES, Hansard for EN-FR and the combination of both for EN-ES-FR. The train, dev and split sets are the same used for language modeling in Chapter 6. We also report the baseline performance by the Char2Vec model from Section 2.6.2 [Jaech *et al.*, 2016b]. Unsurprisingly, the performance of the XLM models is much superior to the char2vec model. From the XLM models, the EN-ES model is able to better fine-tune on the Miami Bangor corpus (98.42% on its validation set) than the EN-FR model on Hansard (93.62%).

Table 7.5 also reports the word-level accuracy on the subset of code-switched sentences from both corpora. The EN-ES model shows a small drop on accuracy on the subset of code-switched sentences (1.33% relative to validation performance) and so does the EN-FR model (1.86% relative drop) on their respective data sets. Both of them underperform severely on the other language pair sets of code-switched sentences: the EN-ES model only achieves 82.30% word-level accuracy on the Hansard subset and the EN-FR model does not

even reach 70% accuracy on the Bangor subset. These results heavily imply that the XLM models, even when trained multilingually on sub-word units that favor cross-lingual word representations, are not able to capture switching across language pairs when one of the languages is not seen during pre-training.

Finally, we observe that the EN-ES-FR model obtains the best performance on the subset of code-switched sentences of both Hansard and Bangor, although these improvements are not significant with respect to the performance from the EN-ES and EN-FR models.

7.7 Conclusions

In this Chapter, we have presented preliminary results on the task of fine-tuning pre-trained multilingual models for code-switching language. We pre-trained three XLM models on EN-ES, EN-FR and EN-ES-FR monolingual Wikipedia data and parallel data from UN transcripts. When fine-tuning these models on the tasks of word-level language tagging we observed that they reached much better performance than our baseline models. When fine-tuning the pre-trained models on the task of language modeling using code-switched data we observed improvements on MLM perplexity, but overall the performance of the fine-tuned models is not desirable.

For future work, we plan to explore different venues to fine-tune the pre-trained XLM models for code-switched language modeling. We hypothesized that the fine-tuning did not work well because we only allowed the top two layers of the Transformer to be re-weighted. We plan to perform more sophisticated fine-tuning, like discriminative tuning and slanted triangular learning rates from [Howard and Ruder, 2018], to fine-tune the whole model.

Part III

Conclusions

Chapter 8

Conclusions

We live in a very multilingual world and as globalization keeps expanding, languages will keep interacting, changing and mixing as a direct consequence of such contact. Similarly, our already linguistically diverse communities will only keep getting more diverse. In a world where almost everyone has access to a cell phone or an internet connection, it is soon going to become imperative that Natural Language Processing and Speech Processing software are able to cope with code-switching.

In the first chapter of this thesis, we identified four main challenges towards adapting or developing code-switching enabled technologies: 1) absence of a large collection of code-switched data; 2) lack of high-quality annotations for models trained on supervised learning algorithms; 3) insufficient understanding of why and when code-switching happens, and useful ways of incorporating that information into our language models; and 4) efficient learning schemes that allow us to use existing monolingual and parallel resources to build code-switching technologies. In this thesis, I have addressed several aspects of each one of these four points. These are the main contributions presented in this thesis:

- **Code-switched data collection scheme:** we introduced a set of code-switched sentence detection methods based on the concept of anchor words and language identification. Anchor words can be easily computed for any language from online resources. We showed that data collected with our anchor methods returned examples with a very high degree of bilingualism and code-switching.

- **A corpus of English-Spanish code-switched tweets:** using our anchor-based collection scheme we retrieved a set of 43,245 tweets in Spanish-English. We obtained word-level language annotations for a subset of 8,285 of them.

- **Part-of-speech tag annotation scheme for English-Spanish:** we started from a crowdsourcing annotation scheme initially developed for English and adapted it to the Universal part-of-speech tagset, then extended it to a code-switching setting. We also created the resources necessary to apply the annotation scheme to Spanish.

- **A collection of part-of-speech tags for the Miami Bangor corpus:** using the annotation scheme, we crowdsourced part-of-speech tags for the whole Miami Bangor corpus. The crowdsourced labels show high agreement with gold standard labels (0.95-0.96) and high average recall across part-of-speech tags (0.87-0.99).

- **Joint part-of-speech and language tagging for code-switching:** we proposed a bi-LSTM model for part-of-speech tagging and another one for joint part-of-speech and language tagging. We showed that our model obtains better tagging accuracy than previous state-of-the-art models. We also showed that the joint model obtained better tagging accuracy.

- **Analysis of the relationship between cognate words and code-switching:** we proved and disproved aspects of the Clyne hypothesis by running statistical analyses on the largest corpus of English-Spanish code-switching that had ever been used for this purposes until now. We confirmed that there is a strong statistical relationship between code-switching and cognate words at the utterance level, when the cognate precedes a switch, but not otherwise.

- **Cognate-based feature set for code-switching:** we proposed a set of cognate-based features and used them for the task of language modeling. We observed that this new set of features can improve language modeling performance with gains as large as the ones provided by gold features like language identifiers or part-of-speech tags. Furthermore, we showed that these features can be used across language pairs and still help obtain better performance.

- **Pre-training cross-lingual language models and fine-tuning for downstream tasks:** we showed preliminary work on using pre-trained XLM models for downstream code-switching tasks. The pre-trained models were trained on large monolingual corpora. We fine-tuned these models for the language modeling on code-switched data and the downstream task of word language tagging.

8.1 Future Work

These are the two main directions that the author of this thesis is planning to pursue to continue work on code-switching:

- **Fine-tuning pre-trained models for downstream code-switching tasks**: our preliminary results on fine-tuning XLM models for code-switched language modeling seems to suggest that more aggressive fine-tuning strategies are needed in order to adapt multilingual language models to code-switching.

- **Automatic Speech Recognition**: the great success seen in the field of speech recognition from new state-of-the-art end-to-end systems opens up exciting possibilities to build code-switch-enabled ASRs. Future work in this direction should tackle cross-lingual training, and include attention mechanisms that focus on the probability of switching languages.

Part IV

Bibliography

Bibliography

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

Heike Adel, Ngoc Thang Vu, Franziska Kraus, Tim Schlippe, Haizhou Li, and Tanja Schultz. Recurrent neural network language modeling for code switching conversational speech. In *Proceedings of ICASSP*, pages 8411–8415. IEEE, 2013.

Heike Adel, Ngoc Thang Vu, and Tanja Schultz. Combination of recurrent neural networks and factored language models for code-switching language modeling. In *Proceedings of ACL*, pages 206–211, 2013.

Basem HA Ahmed and Tien-Ping Tan. Automatic speech recognition of code switching speech using 1-best rescoring. In *Proceedings of IALP*, pages 137–140, 2012.

Mohamed Al-Badrashiny and Mona Diab. The George Washington University system for the code-switching workshop shared task 2016. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 108–111, 2016.

Fahad AlGhamdi, Giovanni Molina, Mona Diab, Thamar Solorio, Abdelati Hawwari, Victor

Soto, and Julia Hirschberg. Part of speech tagging for code switched data. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 98–107. Association for Computational Linguistics, 2016.

Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. Globally normalized transition-based neural networks. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 2442–2452. Association for Computational Linguistics, August 2016.

Peter Auer. *Bilingual conversation*. John Benjamins Publishing, 1984.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of 3rd International Conference on Learning Representations*, San Diego, CA, USA, May 2015.

Pierre Baldi, Søren Brunak, Paolo Frasconi, Giovanni Soda, and Gianluca Pollastri. Exploiting the past and the future in protein secondary structure prediction. *Bioinformatics*, 15(11):937–946, 1999.

Florence Barkin and Alfonso Rivas. On the underlying structure of bilingual sentences. In *Linguistic Society of America, 54th Annual Meeting, Los Angeles, Calif*, 1979.

Utsab Barman, Amitava Das, Joachim Wagner, and Jennifer Foster. Code mixing: A challenge for language identification in the language of social media. In *Proceedings of The First Workshop on Computational Approaches to Code Switching*, pages 13–23, 2014.

Utsab Barman, Joachim Wagner, and Jennifer Foster. Part-of-speech tagging of code-mixed social media content: Pipeline, stacking and joint modelling. In *Proceedings of The Second Workshop on Computational Approaches to Code Switching*, pages 42–51, 2016.

Leslie M Beebe. Social and situational factors affecting the communicative strategy of dialect code-switching. *International Journal of the Sociology of Language*, 1981(32):139–149, 1981.

Hedi M Belazi, Edward J Rubin, and Almeida Jacqueline Toribio. Code switching and X-bar theory: The functional head constraint. *Linguistic inquiry*, pages 221–237, 1994.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

Ann Bies, Justin Mott, Colin Warner, and Seth Kulick. English web treebank LDC2012T13. `https://catalog.ldc.upenn.edu/LDC2012T13`, 2012.

Jeff A Bilmes and Katrin Kirchhoff. Factored language models and generalized parallel backoff. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: companion volume of the Proceedings of HLT-NAACL*, volume 2, pages 4–6. Association for Computational Linguistics, 2003.

Mirjam Broersma and Kees De Bot. Triggered codeswitching: A corpus-based evaluation of the original triggering hypothesis and a new alternative. *Bilingualism: Language and cognition*, 9(1):1–13, 2006.

Chris Callison-Burch and Mark Dredze. Creating speech and language data with amazon's mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 1–12. Association for Computational Linguistics, 2010.

Chris Callison-Burch. Fast, cheap, and creative: evaluating translation quality using amazon's mechanical turk. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, volume 1, pages 286–295. Association for Computational Linguistics, 2009.

Mónica Stella Cárdenas-Claros and Neny Isharyanti. Code-switching and code-mixing in Internet chatting: between 'yes', 'ya', and 'si' – a case study. *The JALT CALL Journal*, 5(3):67–78, 2009.

Marine Carpuat. Mixed language and code-switching in the canadian hansard. In *Proceedings of the first workshop on computational approaches to code switching*, pages 107–115. Association for Computational Linguistics, 2014.

Özlem Çetinoglu. A Turkish-German codeswitching corpus. In *Proceedings of LREC*, pages 4215–4220, 2016.

Khyathi Chandu, Thomas Manzini, Sumeet Singh, and Alan W. Black. Language informed modeling of code-switched text. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, pages 92–97, Melbourne, Australia, 2018. Association for Computational Linguistics.

Tanya L Chartrand and John A Bargh. The chameleon effect: the perception–behavior link and social interaction. *Journal of personality and social psychology*, 76(6):893, 1999.

François Chollet. Keras. `https://github.com/fchollet/keras`, 2015.

Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. Attention-based models for speech recognition. In *Advances in neural information processing systems*, pages 577–585, 2015.

Michael G Clyne. *Transference and triggering: Observations on the language assimilation of postwar German-speaking migrants in Australia.* Martinus Nijhoff, 1967.

Michael G Clyne. Triggering and language processing. *Canadian Journal of Psychology/Revue canadienne de psychologie*, 34(4):400, 1980.

Michael Clyne. Constraints on code switching: How universal are they? *Linguistics*, 25:739–764, 1987.

Michael G Clyne. *Dynamics of language contact: English and immigrant languages.* Cambridge University Press, 2003.

Michael Collins. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the Second conference on Empirical methods in natural language processing*, volume 10, pages 1–8. Association for Computational Linguistics, 2002.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537, 2011.

Alexis Conneau and Guillaume Lample. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems 32*, pages 7057–7067. Curran Associates, Inc., 2019.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. In *Proceedings of Sixth International Conference on Learning Representations (ICLR)*, Vancouver, Canada, April 2018.

Cristian Danescu-Niculescu-Mizil, Michael Gamon, and Susan Dumais. Mark my words!: linguistic style accommodation in social media. In *Proceedings of the 20th international conference on World wide web*, pages 745–754. ACM, 2011.

Brenda Danet and Susan C Herring. *The multilingual Internet: Language, culture, and communication online*. Oxford University Press on Demand, 2007.

Crystal David. *Language and the Internet*. Cambridge, CUP, 2001.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 4171–4186, 2019.

Kevin Donnelly and Margaret Deuchar. The Bangor autoglosser: a multilingual tagger for conversational text. *ITA11, Wrexham, Wales*, pages 17–25, 2011.

Kevin Donnelly and Margaret Deuchar. Using constraint grammar in the Bangor autoglosser to disambiguate multilingual spoken text. In *Constraint Grammar Applications: Proceedings of the NODALIDA 2011 Workshop, Riga, Latvia*, pages 17–25, 2011.

Timothy Dozat, Peng Qi, and Christopher D. Manning. Stanford's graph-based neural dependency parser at the CoNLL 2017 shared task. In *Proceedings of the CoNLL 2017*

Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, pages 20–30, 2017.

Kevin Duh and Katrin Kirchhoff. Automatic learning of language model structure. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 148–154, Geneva, Switzerland, aug 23–aug 27 2004. COLING.

Andreas Eisele and Yu Chen. MultiUN: A multilingual corpus from united nation documents. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May 2010. European Language Resources Association (ELRA).

Heba Elfardy, Mohamed Al-Badrashiny, and Mona Diab. AIDA: identifying code switching in informal arabic text. In *Proceedings of The First Workshop on Computational Approaches to Code Switching*, pages 94–101, 2014.

Tim Finin, Will Murnane, Anand Karandikar, Nicholas Keller, Justin Martineau, and Mark Dredze. Annotating named entities in twitter data with crowdsourcing. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 80–88. Association for Computational Linguistics, 2010.

Juan Carlos Franco and Thamar Solorio. Baby-steps towards building a Spanglish language model. In *Proceedings of International Conference on Intelligent Text Processing and Computational Linguistics*, pages 75–84. Springer, 2007.

Melinda Fricke and Gerrit Jan Kootstra. Primed codeswitching in spontaneous bilingual dialogue. *Journal of Memory and Language*, 91:181–201, 2016.

Yarin Gal and Zoubin Ghahramani. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 1019–1027, 2016.

Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. Building large monolingual dictionaries at the leipzig corpora collection: From 100 to 200 languages. In *Proceedings of LREC*, pages 759–765, 2012.

P Goyal, Manav R Mital, and A Mukerjee. A bilingual parser for Hindi, English and code-switching structures. In *EACL Workshop of Computational Linguistics for South Asian Languages*, pages 15–22. Association for Computational Linguistics, April 2003.

Alex Graves. *Supervised Sequence Labelling*, pages 5–13. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

Klaus Greff, Rupesh Kumar Srivastava, Jan Koutník, Bas R. Steunebrink, and Jürgen Schmidhuber. LSTM: A search space odyssey. *IEEE Trans. Neural Netw. Learning Syst.*, 28(10):2222–2232, 2017.

Julia Hirschberg. Accent and discourse context: Assigning pitch accent in synthetic speech. In *AAAI*, volume 90, pages 952–957, 1990.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard H Hovy. Learning whom to trust with mace. In *Proceedings of the NAACL HLT 2013*, pages 1120–1130. Association for Computational Linguistics, 2013.

Dirk Hovy, Barbara Plank, and Anders Søgaard. Experiments with crowdsourced re-annotation of a POS tagging data set. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 377–382, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, 2018.

Pei-Yun Hsueh, Prem Melville, and Vikas Sindhwani. Data quality from crowdsourcing: a study of annotation selection criteria. In *Proceedings of the NAACL HLT 2009 workshop on active learning for natural language processing*, pages 27–35. Association for Computational Linguistics, 2009.

Liang Huang, Suphan Fayong, and Yang Guo. Structured perceptron with inexact search. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 142–151. Association for Computational Linguistics, 2012.

Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

Aaron Jaech, George Mulcaire, Shobhit Hathi, Mari Ostendorf, and Noah A. Smith. Hierarchical character-word models for language identification. In *Proceedings of The Fourth International Workshop on Natural Language Processing for Social Media*, pages 84–93, Austin, TX, USA, November 2016. Association for Computational Linguistics.

Aaron Jaech, George Mulcaire, Shobhit Hathi, Mari Ostendorf, and Noah A Smith. A neural model for language identification in code-switched tweets. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 60–64, 2016.

Anupam Jamatia, Björn Gambäck, and Amitava Das. Part-of-speech tagging for code-mixed English-Hindi Twitter and Facebook chat messages. In *Proceedings of Recent Advances in Natural Language Processing*, pages 239–248, 2015.

Mukund Jha, Jacob Andreas, Kapil Thadani, Sara Rosenthal, and Kathleen McKeown. Corpus creation for new genres: A crowdsourced approach to pp attachment. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 13–20. Association for Computational Linguistics, 2010.

Aravind K Joshi. Processing of sentences with intra-sentential code-switching. In *Proceedings of the 9th conference on Computational Linguistics*, volume 1, pages 145–150. Academia Praha, 1982.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations (ICLR)*, May 7-9 2015.

Efthymios Kouloumpis, Theresa Wilson, and Johanna Moore. Twitter sentiment analysis:

The good the bad and the omg! In *Fifth International AAAI conference on weblogs and social media*, pages 538–541, 2011.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

Sophia Yat Mei Lee and Zhongqing Wang. Emotion in code-switching texts: Corpus construction and analysis. *Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing*, pages 91–99, 2015.

Rivka Levitan, Agustín Gravano, Laura Willson, Stefan Benus, Julia Hirschberg, and Ani Nenkova. Acoustic-prosodic entrainment and social behavior. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human language technologies*, pages 11–19. Association for Computational Linguistics, 2012.

Ying Li and Pascale Fung. Code-switch language model with inversion constraints for mixed language speech recognition. In *Proceedings of COLING 2012*, pages 1671–1680, 2012.

Ying Li and Pascale Fung. Language modeling with functional head constraint for code switching speech recognition. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, pages 907–916, 2014.

Wang Ling, Chris Dyer, Alan W Black, Isabel Trancoso, Ramón Fermandez, Silvio Amir, Luís Marujo, and Tiago Luís. Finding function in form: Compositional character models for open vocabulary word representation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1520–1530, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

Marcus Liwicki, Alex Graves, Horst Bunke, and Jrgen Schmidhuber. A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks. In *In Proceedings of the 9th International Conference on Document Analysis and Recognition, ICDAR 2007*, 2007.

Marco Lui and Timothy Baldwin. langid.py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 System Demonstrations*, pages 25–30, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

Tetyana Lyudovyk and Valeriy Pylypenko. Code-switching speech recognition for closely related languages. In *Proceedings of 4th International Workshop on Spoken Language Technologies for Under-resourced Languages (SLTU)*, pages 188–193, Saint Petersburg, Russia, May 2014.

Jacob E. Mainzer. Labeling parts of speech using untrained annotators on mechanical turk. Master's thesis, The Ohio State University, 2011.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330, 1993.

Mitchell P Marcus, Beatrice Santorini, Mary Ann Marcinkiewicz, and Ann Taylor. Treebank-3 LDC99T42. `https://catalog.ldc.upenn.edu/ldc99t42`, 1999.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. Universal dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 92–97, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.

Gideon Mendels, Erica Cooper, Victor Soto, Julia Hirschberg, Mark JF Gales, Kate M Knill, Anton Ragni, and Haipeng Wang. Improving speech recognition and keyword search for low resource languages using web data. In *Proceedings of INTERSPEECH*, pages 829–833, 2015.

Gideon Mendels, Erica Cooper, and Julia Hirschberg. Babler-data collection from the web to support speech recognition and keyword search. In *Proceedings of Web as Corpus Workshop (WAC-X) and the EmpiriST Shared Task*, pages 72–81, 2016.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

Lesley Milroy and Matthew Gordon. Style-shifting and code-switching. *Sociolinguistics. Oxford: Blackwell Publishing Ltd*, pages 198–223, 2003.

Ministry of Home Affairs, Government of India. 2011 census data. `http://censusindia.gov.in/2011-Common/CensusData2011.html`, 2011.

Giovanni Molina, Nicolas Rey-Villamizar, Thamar Solorio, Fahad AlGhamdi, Mahmoud Ghoneim, Abdelati Hawwari, and Mona Diab. 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, 2016.

Robert Moore. Fast high-accuracy part-of-speech tagging by independent classifiers. In *COLING*, pages 1165–1176, 2014.

Carol Myers-Scotton and Janice Jake. *A universal model of code-switching and bilingual language processing and production.* Cambridge University Press, 2009.

Carol Myers-Scotton. *Duelling languages: Grammatical structure in codeswitching.* Oxford University Press, 1997.

Ani Nenkova, Agustin Gravano, and Julia Hirschberg. High frequency word entrainment in spoken dialogue. In *Proceedings of the 46th annual meeting of the association for*

computational linguistics on human language technologies: Short papers, pages 169–172. Association for Computational Linguistics, 2008.

Franz Josef Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alexander M Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, et al. A smorgasbord of features for statistical machine translation. In *HLT-NAACL*, pages 161–168, 2004.

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237, 2018.

Slav Petrov, Dipanjan Das, and Ryan McDonald. A universal part-of-speech tagset. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2089–2096, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA).

Mario Piergallini, Rouzbeh Shirvani, Gauri S Gautam, and Mohamed Chouikha. The Howard University system submission for the shared task in language identification in spanish-english codeswitching. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 116–120, 2016.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training, 2018.

David Reitter and Johanna D Moore. Priming of syntactic rules in task-oriented dialogue and spontaneous conversation. In *Proceedings of the Cognitive Science Society*, volume 28, pages 1–6, 2006.

Paul Rodrigues and Sandra Kübler. Part of speech tagging bilingual speech transcripts

with intrasentential model switching. In *AAAI Spring Symposium*, pages 56–63, January 2013.

Filipe Rodrigues, Francisco Pereira, and Bernardete Ribeiro. Sequence labeling with multiple annotators. *Machine Learning*, 95(2):165–181, 2014.

Mike Rosner and Paulseph-John Farrugia. A tagging algorithm for mixed language identification in a noisy domain. In *Proceedings of INTERSPEECH*, pages 190–193, 2007.

Alexander M Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 379–389, 2015.

Hasim Sak, Andrew W Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Proceedings of Interspeech*, pages 338–342, 2014.

Younes Samih, Suraj Maharjan, Mohammed Attia, Laura Kallmeyer, and Thamar Solorio. Multilingual code-switching identification via LSTM recurrent neural networks. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 50–59, Austin, TX, 2016.

Younes Samih. An Arabic-Moroccan darija code-switched corpus. In *Proceedings of LREC*, pages 4170–4175, 2016.

David Sankoff and Shana Poplack. A formal grammar for code-switching. *Research on Language & Social Interaction*, 14(1):3–45, 1981.

Beatrice Santorini. *Part-of-speech tagging guidelines for the Penn Treebank Project (3rd revision)*. LDC, UPenn, 3 edition, 1990. 2nd Printing.

Royal Sequiera, Monojit Choudhury, and Kalika Bali. POS tagging of Hindi-English code mixed text from social media: Some machine learning experiments. In *Proceedings of the 12th International Conference on Natural Language Processing*, pages 237–246, Trivandrum, India, December 2015. NLP Association of India.

Libin Shen, Giorgio Satta, and Aravind Joshi. Guided learning for bidirectional sequence classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 760–767, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Christopher D. Manning. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, 2014.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y Ng. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 254–263. Association for Computational Linguistics, 2008.

Thamar Solorio and Yang Liu. Learning to predict code-switching points. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, pages 973–981, 2008.

Thamar Solorio and Yang Liu. Part-of-speech tagging for English-Spanish code-switched text. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, pages 1051–1060, 2008.

Thamar Solorio, Elizabeth Blair, Suraj Maharjan, Steven Bethard, Mona Diab, Mahmoud Gohneim, Abdelati Hawwari, Fahad AlGhamdi, Julia Hirschberg, Alison Chang, et al. 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, 2014.

Victor Soto and Julia Hirschberg. Crowdsourcing universal part-of-speech tags for code-switching. In *Proceedings of Interspeech*, pages 77–81, Stockholm, Sweden, August 2017.

Victor Soto, Nishmar Cestero, and Julia Hirschberg. The role of cognate words, POS tags, and entrainment in code-switching. In *Proceedings of Interspeech*, pages 1938–1942, Hyderabad, India, September 2018.

Andreas Stolcke, Jing Zheng, Wen Wang, and Victor Abrash. SRILM at sixteen: Update and outlook. In *Proc. IEEE Automatic Speech Recognition and Understanding Workshop*. IEEE SPS, December 2011.

Andreas Stolcke. SRILM – an extensible language modeling toolkit. In *Seventh International Conference on Spoken Language Processing (ICSLP)*, pages 901–904, 2002.

Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. LSTM neural networks for language modeling. In *Proceedings of Interspeech*, pages 194–197, 2012.

Martin Sundermeyer, Tamer Alkhouli, Joern Wuebker, and Hermann Ney. Translation modeling with bidirectional recurrent neural networks. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, pages 14–25, 2014.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc., 2014.

Paul Taylor and Alan W. Black. Assigning phrase breaks from part-of-speech sequences. *Computer Speech and Language*, 12(2):99117, April 1998.

Paul Taylor, Alan W Black, and Richard Caley. The architecture of the festival speech synthesis system. In *Third ESCA Workshop in Speech Synthesis*, pages 147–151. International Speech Communication Association, 1998.

Jörg Tiedemann and Nikola Ljubešić. Efficient discrimination between closely related languages. In *Proceedings of COLING 2012*, pages 2619–2634, Mumbai, India, December 2012. The COLING 2012 Organizing Committee.

Jörg Tiedemann. Parallel data, tools and interfaces in OPUS. In *Proceedings of LREC*, volume 2012, pages 2214–2218, 2012.

US Census Bureau. Annual estimates of the resident population by sex, age, race, and Hispanic origin for the United States: April 1, 2010 to July 1,

2014. https://factfinder.census.gov/bkmk/table/1.0/en/PEP/2014/PEPASR6H?slice=hisp~hisp!year~est72014, 7 2014.

US Census Bureau. American community survey 1-year estimates: S1601 - language spoken at home. https://factfinder.census.gov/bkmk/table/1.0/en/ACS/15_1YR/S1601, 11 2015.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017.

David Vilares, Miguel A Alonso, and Carlos Gómez-Rodrıguez. Sentiment analysis on monolingual, multilingual and code-switching twitter corpora. In *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA)*, pages 2–8, 2015.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164, 2015.

Yogarshi Vyas, Spandana Gella, and Jatin Sharma. POS tagging of English-Hindi code-mixed social media content. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, pages 974–979, 2014.

Peilu Wang, Yao Qian, Frank K Soong, Lei He, and Hai Zhao. A unified tagging solution: Bidirectional LSTM recurrent neural network with word embedding. *arXiv preprint arXiv:1511.00215*, 2015.

Oliver Watts, Junichi Yamagishi, and Simon King. Unsupervised continuous-valued word features for phrase-break prediction without a part-of-speech tagger. In *INTERSPEECH*, pages 2157–2160, 2011.

Genta Indra Winata, Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. Code-switching language modeling using syntax-aware multi-task learning. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, pages 62–67, Melbourne, Australia, 2018. Association for Computational Linguistics.

Kathryn A Woolard. Simultaneity and bivalency as strategies in bilingualism. *Journal of linguistic anthropology*, 8(1):3–29, 1998.

Ellen Woolford. Bilingual code-switching and syntactic theory. *Linguistic inquiry*, 14(3):520–536, 1983.

Kaisheng Yao and Geoffrey Zweig. Sequence-to-sequence neural net models for grapheme-to-phoneme conversion. In *Proceedings of INTERSPEECH*, pages 3330–3334, Dresden, Germany, September 2015.

Heiga Zen, Keiichi Tokuda, and Alan W Black. Statistical parametric speech synthesis. *Speech Communication*, 51(11):1039–1064, 2009.

Part V

Appendices

Appendix A

Disambiguation Task for Specific Tokens

A.1 List of Disambiguation Questions for English Tokens

About:

In the context of the sentence, is 'about' used to mean 'approximately'?

ADV Yes.

ADP No.

All:

In the context of the sentence, 'all'... ?

ADV appears before an adjective or an adverb. For example: 'You got it all wrong', 'He traveled all around the city.'

DET appears before a noun or noun phrase. For example: 'We hang out all day.', 'He had all the right reasons.'

NOUN None of the above. For example: 'I gave my all.'

Around:

In the context of the sentence, could 'around' be replaced with 'approximately'?

ADV Yes. For example: 'He will arrive at around 3PM.'

ADP No. For example: 'He lives around the block.'

As:

In the context of the sentence, does 'as'... ?

ADV ...have a meaning similar to 'so'. For example: 'This one is not AS good.'

ADV ...appears first in a sequence like '...AS soon as...', '...AS tall as...', '...AS long as....'

ADP ...introduces a comparison. For example: 'You are not as tall AS me'

ADP ...specifies a role. For example: 'What is your opinion AS a parent?'

SCONJ None of the above. For example: 'She flew AS I came in', 'As soon AS I told him'.

Back:

In the context of the sentence, is 'back' an adjective, a noun or a verb?

ADJ Adjective. For example: in 'The back door.', 'back' modifies the noun 'door'.)

NOUN Noun. For example: 'My back hurts.'

VERB Verb. For example: 'I need you to back me up on this.'

ADV None of the above. For example: 'Step back from the curb'.

Both:

In the context of the sentence, is 'both' being used with 'and'?

CONJ Yes. For example: 'BOTH girls AND boys can do it'

DET No. For example: 'Both of you are grounded!'

Bottom:

In the context of the sentence, is 'bottom' an adjective, a noun or a verb?

ADJ Adjective ('The bottom drawer')

NOUN Noun ('The bottom of the ocean is still a mistery.')

VERB Verb ('The stocks bottomed out this morning.')

But:

In the context of the sentence, 'but' is more closely translated into Spanish as...?

CONJ 'PERO'. For example: 'You can go but don't tell him' would be translated as 'Puedes ir pero no se lo digas.'

ADP 'EXCEPTO'. For example:'Everybody but me' would be translated as 'Todo el mundo excepto yo.'

ADV 'SOLO' or 'NO ... MAS QUE'. For example: 'She is but a child' would be translated as 'No es mas que una nia.'

Can:

In the context of the sentence, is 'can' a verb that takes the meaning of 'being able to' or 'know'?

AUX Yes. For example: 'I can speak Spanish.'

NOUN No, it refers to a cylindrical container. For example: 'Pass me a can of beer.'

Coming:

In the context of the sentence, is 'coming' being used like 'upcoming'?

ADJ Yes. For example: 'Any plans this coming week?'

VERB No. For example: 'She's coming for the 8pm show.

Dear:

In the context of the sentence, is 'dear' being used like:

NOUN Yes, dear.

ADJ Dear Bob.

INTJ Dear!

Down:

In the context of the sentence, is 'down' being used as ...?

ADJ ...an Adjective. For example: 'Prices are down.', 'They are down 3-0.', 'Are you down to hang out?', 'The system is down.'

VERB ...a Verb. For example: 'He downed two beers already.', 'The storm downed several trees.'

ADP ...being used to refer 'from the higher end to the lower end' or 'from one end to the other'. For example: 'The ball rolled down the hill', 'The bus went down the street'

ADV None of the above. For example: 'The cat jumped down from the table.', 'I went down to Miami for a conference.', 'The computer has been shut down.'

Either:

In the context of the sentence, 'either':

CONJ Means 'o' in Spanish. For example: 'EITHER the girls OR the boys can do it.'

ADV Means 'as well'. For example: 'I don't like her either.'

DET Means 'cualquier' o 'cualquiera' in Spanish and appears before a noun or noun phrase. For example: 'Either option is ok.'

PRON Means 'cualquiera' in Spanish and DOES NOT appear before a noun or noun phrase. For example: 'Either is fine.'

Far:

In the context of the sentence, is 'far' used like:

ADV an Adverb? For example: 'She lives far away'.

ADJ an Adjective? For example: 'The far end of the field'.

Front:

In the context of the sentence, is 'front':

ADJ an Adjective? For example: in 'The front door.', 'front' modifies the noun 'door'.

NOUN a Noun? For example: 'He always sits in the front of the class.'.

VERB a Verb? For example: 'My apartment fronts West 119th Street.'.

Her:

In the context of the sentence, would 'her' be replaced by 'his' or 'him'?

DET HIS. For example: 'Her bag is right there.'

PRON HIM. For example: 'I told her to come asap.'

His:

In the context of the sentence, would 'his' be replaced by 'her' or 'hers'?

DET HER. For example: 'His bag is right there.'

PRON HERS. For example: 'Is that plate his or mine?'

Less:

In the context of the sentence, is 'less' referring to:

ADJ Less of an amount. For example: 'You should eat less food'

ADV Less of an action. For example: 'You should eat less (frequently)'

CONJ The substraction operator. For example: 'Five less than three is two.'

Over:

In the context of the sentence, is 'over'...

CONJ ...being used as a mathematical operator? For example: Eight over two is four.

INTJ ...being used as way to end communication? For example: 'Received? Over!'

ADP ...appearing before a noun, a pronoun or a phrase? For example: 'There is a bridge over THE RIVER.', 'I would choose him over YOU.', 'She is finally over LOSING HER JOB'?

ADV None of the above. ('Let me think that over.', 'The building just fell over.', 'Come over and play!')

May:

In the context of the sentence, is 'may' a noun?

PROPN Yes. For example: 'I was born in May.'?

AUX No. For example: 'May I use the toilet, please?'

More:

In the context of the sentence, is 'more':

ADJ Referring to more of an amount or object. For example: 'You should eat more food', 'It grows to five feet or more', or 'more of the same'.

ADV Replaceable by an adverb (such as 'almost'), referring to more of an action (e.g. 'You should run more') or modifies an adverb (e.g. 'more carefully')

Much:

In the context of the sentence, is 'much' and adjective, adverb or pronoun?

ADJ Adjective. For example: 'That is too much food.'.

ADV Adverb. For example: 'That is much better!'

PRON Pronoun. For example: 'Much has been said about the financial crisis.'

Near:

In the context of the sentence, is 'near' a preposition, adjective or adverb?

ADP Preposition. For example: 'We were near the station.'

ADJ Adjective. For example: 'The near side of the moon.'

ADV Adverb. For example: 'Her record is near perfect', 'They had gotten quite near.'

Neither:

In the context of the sentence, is 'neither' being used with 'nor'?

CONJ Yes. For example: 'Neither the girls nor the boys can do it.'

DET No. For example: 'Neither option is great.'

Next:

In the context of the sentence, 'next'

ADJ Is an adjective modifying or describing a noun. For example: 'The next train.'

ADV Is an adverb. For example: 'They live next to me.'

No:

In the context of the sentence, is 'no':

SPA In Spanish.

ENG In English.

In the context of the sentence, in which 'no' is in Spanish:

INTJ Being used as opposite of yes. For example: 'NO!', 'NO?', 'NO, yo no lo hice.'

ADV It is used for verb negation. For example: 'Yo NO lo HICE.'

In the context of the sentence, in which 'no' is in English:

INTJ Being used as opposite of yes. For example: 'No!', 'No?', 'No, I don't want to talk to them.'

DET Appears before a noun or phrase. For example: 'There is NO answer yet.', 'NO spoiled child will tell me what to do.', 'Last time I asked, they had NO idea.'

ADV None of the above. For example: 'This is no longer an issue.'

One:

In the context of the sentence, can 'one':

PRON Be replaced by 'he', 'she' or 'it'? For example: 'One shouldn't smoke in public parks.'

NOUN Be replaced by 'person', 'thing', 'object'....? It might be preceded by an article like 'the'. For example: 'The one who cares', 'The person who cares.'

NUM None of the above. For example: 'One of the reasons', 'One dollar.'

Only:

In the context of the sentence, can 'only' be replaced by 'sole'?

ADJ Yes. For example: 'The only solution', 'The sole solution.'

ADV No. For example:''I only brought bread and milk.'

Other:

In the context of the sentence, 'other'...

ADV Means 'apart from' in the sentence 'other than...'. For example: 'Other than that, I'm fine.'

ADJ Means 'second' or 'different'. For example: 'I get paid every other week.'

PRON Means 'other person', 'other people', or other subject in general. For example: 'The other is bad.'

DET None of the above, always goes before a noun. For example: 'Other people are coming later.'

Please:

In the context of the sentence, is 'please' used to make a polite request?

INTJ Yes. Example: 'Please, pass me the bread.'

VERB No. Example: 'You can do what you please.'

Plus:

In the context of the sentence, is 'plus'...

CONJ ...being used as mathematical operator? For example: 'Two plus two is four.'

NOUN ...being used as a synonym of 'asset' or positive quantity? For example: 'He is a real asset to the team'

ADJ None of the above. For example: 'A battery has a plus pole and a minus pole.'

Side:

In the context of the sentence, is 'side'...

ADJ ...modifying a noun? For example: 'the side door.'

VERB ...acting like a verb? For example: 'I need you to side with me.'

NOUN None of the above. For example: 'A square has four sides.'

So:

In the context of the sentence, 'so'...

SCONJ Is used like 'so that' or 'therefore'? (eg. 'I opened the door so (that) he could leave', 'He ate too much cake, so he fell ill.')

INTJ Is used after a pause for thought to introduce a new topic, question or story (eg. 'So, who's coming with me?) or short for 'so what?' (eg. 'So???')

ADV None of the above. (eg. 'It was so hot', 'so far as', 'so long as', 'so much as', 'Just say so!')

That:

In the context of the sentence, could 'that' be replaced by:

PRON 'WHICH' and does not precede a noun phrase as in 'The car that can't start.'

PRON 'IT' as in 'That is crazy.'

DET 'THE' and precedes a noun phrase as in 'I want that car.'

SCONJ None of the above. Select this option ONLY if the first three options are definitely wrong. In this case, 'that' can be removed from the sentence without any consequence. For example: 'I was surprised that he didn't come.', 'Have you heard that she got fired?'

This:

In the context of the sentence, could 'this' be replaced by:

PRON 'it' as in 'This is crazy.'

DET 'the' as in 'I want this car.'

Those:

In the context of the sentence, could 'those' be replaced by:

PRON 'they' as in 'They are crazy.'

DET 'the' as in 'I want those cars.'

These:

In the context of the sentence, could 'these' be replaced by:

PRON 'they' as in 'These are crazy.'

DET 'the' as in 'I want these cars.'

Then:

In the context of the sentence, is 'then' being used like 'former'?

ADJ Yes. For example: 'The then president traveled to Finland.'

ADV No. For example: 'What's the plan then.?

There:

In the context of the sentence, does 'there' refer to a location and can be replaced by an adverb?

ADV Yes. For example: 'I want to go there.'

PRON No. For example: 'There was a loud noise.'

Times:

In the context of the sentence, 'times' translation in Spanish would be...

CONJ POR. For example: 'Seven times five is thirty-five.' would be translated as 'Siete por cinco es treinta y cinco.'

NOUN TIEMPOS. For example: 'Modern times are so very different from the past.' would be translated as 'Estos tiempos modernos son tan distintos a los pasados.'

VERB MEDIR o CRONOMETRAR. For example: 'Every day he times how long it takes his ride home.' would be translated as 'Cada día cronometra lo que tarda en llegar a casa.'

To:

In the context of the sentence, is 'to' a particle for the infinitive form of a verb?

PART Yes. For example: 'to be', 'to have', 'to move.'

ADP No, it's a preposition. For example: 'He is moving to Arizona next week.'

Top:

In the context of the sentence, is 'top'...

ADJ ...modifying a noun? For example: 'top drawer'

NOUN ...a noun? For example: 'The kite got caught at the top of a tree.'

VERB ...a verb? For example: 'Top my ice cream with chocolate sauce.'

ADV ...taking the meaning of 'first'? For example: 'She came top in her French exam.'

Up:

Is 'up'...

ADJ ...an adjective? For example: 'Time is up!', 'I am up for a tree.', 'It's 1AM and I am still up.'

VERB ...a verb? For example: 'If we up the volume, maybe we'll be able to hear.

NOUN ...a noun? For example: 'Up is the correct way to go.', 'There are always ups and downs.'

ADP ...used to refer to 'towards the top of, towards a point of reference, or further along'? For example: 'The cat went up the tree.', 'They took a boat up the river.', 'Go up the street.'

ADV ...modifying a verb? For example: 'Look it up in the dictionary.', 'Tear up the contract.', 'Cheer up man!', 'Drink up, the pub is closing.', 'Put up your weapons.'

Very:

In the context of the sentence, is 'very' being used like 'mere', 'sheer', or 'real'?

ADJ Yes. For example: 'The very thought.'

ADV No. For example: 'I am very grateful for the present.'

Vice:

In the context of the sentence, is 'vice' being in the same context as 'vice president' or 'vice principal'?

ADJ Yes.

NOUN No. For example: 'The drugs and vice department is underfunded.'

Well:

In the context of the sentence, is 'well':

NOUN Being used as a noun ('The well was full of water.')

ADJ Being used as the opposite of sick ('He is feeling well.')

INTJ Being used to start a sentence: to acknowledge a statement or situation ('Well, I thought it was good.'), as an exclamation of surprise ('Well, well, well, look who's here!), used to fill gaps 'Well...we went for a picnic.', or to express reluctance ('It was a bit...well...loud.')

ADV None of the above. Examples: 'He does his job well', 'A well done steak', 'The author is well known.'

What:

In the context of the sentence, does 'what' appear immediately before a noun (not a pronoun) and any adjectives it may have?

DET Yes. For example: 'What KIND do you want?', 'Tell me what BOOK to buy.'

PRON No. For example: 'What is your problem?, 'What was that about?', 'You know what I mean?'

Whatever:

In the context of the sentence, does 'whatever' appear immediately before a noun and any adjectives it may have?

DET Yes. For example: 'Whatever EVENTS happen, we will be alright.', 'Sell whatever BOOKS you own.'

PRON No. For example: 'Whatever happens, we will be all right.'

When:

In the context of the sentence, can 'when' be substituted by 'at what time', 'what time', 'how soon', 'in what circumstances' or 'on which'?

ADV YES. For example: 'when did you last see him?', 'since when have you been interested?', 'when can I see you?', 'when would such a rule be justifiable?', 'Saturday is the day when I get my hair done.'

SCONJ NO. For example: 'I loved math when I was in school', 'He had just drifted off to sleep when the phone rang', 'Why bother to paint it when you can photograph it with the same effect?'

Will:

In the context of the sentence, is 'will' a noun, a verb, or a modal verb?

NOUN Noun. For example: 'He has no will.'

VERB Verb meaning 'wish something to happen' or 'bequath something to someone'. For example: 'Tom willed him to leave.'

AUX Modal verb for future tense. For example: 'Sarah will visit her relatives.'

Worth:

In the context of the sentence, does 'worth' appear before a value or quantity? (e.g. 'worth ten dollars', 'worth a lot', 'worth your attention', 'worth the effort', etc.)

ADP Yes. For example: 'He's not worth the pain.'

NOUN No. For example: 'His net worth is really impressive.'

Yes:

In the context of the sentence, 'yes'...

INTJ Is used to give an affirmative response. For example: 'Yes, I will do it.', 'Yes?'

NOUN Is an affirmative answer or decision, especially in voting. For example: 'He voted yes in the referendum', 'Is that a yes?', 'They are still counting the yeses and the noes.'

Yet:

In the context of the sentence, could 'yet' be replaced by 'but'?

CONJ Yes. For example: 'I like this, yet I wouldn't eat it again.'

ADV No. For example: 'I'm not there yet.'

A.2 List of Disambiguation Questions for Spanish Tokens

Algo:

In the context of the sentence, is the word 'algo'...

PRON Used to refer to an unknown identity ('Hay algo para mi?') or a small or unknown quantity ('Apostemos algo.')

ADV Used to refer to 'un poco, no del todo' ('Está algo sucio.'), 'con poca intensidad' ('Llovía algo cuando salí') or a short period of time ('Durmió algo.')

Bien:

In the context of the sentence, the word 'bien':

NOUN Means 'opposite of evil'. For example: 'El bien y el mal.'

NOUN Means 'property' or 'resources'. For example: 'Los bienes familiares.'

ADV Means 'well', 'nice' or 'fine'. For example: 'Hiciste bien.'

ADV Means 'very'. For example: 'Está bien sucio.'

Bueno:

In the context of the sentence, word 'bueno' used to start a conversation or to refer to the speaker's discontent or partial agreement?

INTJ Yes. For example: 'Bueno? Quien llama?', 'Bueno, lo que me faltaba!', 'Bueno, mira, haz lo que quieras.'

ADJ No. For example: 'El salmón estaba muy bueno.'

Como:

In the context of the sentence, choose the option that is most correct about the word 'como':

VERB It is a form of the verb 'comer'. For example: 'Como fruta todos los días'.

ADP It can be replaced by 'en calidad de', 'en concepto de' or 'a modo de' without changing the meaning and structure of the sentence.

ADV It is equivalent to 'más o menos', 'aproximadamente' (eg. 'Debe hacer como tres aos que no nos veíamos.'), 'tan pronto como' ('Como recibí su carta, me alegre mucho.'), 'según' ('Como me decía mi padre, de los politicos no te creas nada.'), 'de la forma que' ('Te lo contaré todo como ha ocurrido.').

SCONJ It is used to indicate a comparison of equality. For example: 'Julia es tan alta como su madre.', 'Invitaron a la reunion tanto a Carla como a Pilar.', 'No hay como ponerse a trabajar para terminar pronto.'

SCONJ It links a condition to a consequence in causal and conditional sentences. For example: 'Como no estudié, suspendí.', 'Como no estudies, suspenderás'.

ADV Appears followed by 'si' or 'que'. For example: 'Se comportó como si estuviera solo.', 'Hizo como que no me vio.')

Cómo:

In the context of the sentence, the word 'cómo'...

NOUN Appears right after an article, as 'el'. For example: 'El problema no era el cómo sino el donde.'

INTJ Is used to refer to anger. For example: 'Cómo?! Yo jamas haría eso.'

ADV None of the above. For example: 'Cómo te encuentras?', 'Cómo te gustan los huevos?'

Cuando:

In the context of the sentence, can the word 'cuando' be replaced by...

SCONJ ...'en caso de que', 'a pesar de que', 'puesto que' without changing the meaning or structure of the sentence.

ADP ...'en el tiempo de' or 'el tiempo de' without changing the meaning or structure of the sentence.

ADV None of the above.

Donde:

In the context of the sentence, can the word 'donde' be replaced by 'cerca de', 'en casa de', 'el lugar de', 'la casa de', 'al lugar de' or 'a casa de' without changing the meaning or structure of the sentence?

ADP Yes. For example: 'Fuimos donde Antonio' becomes 'Fuimos a casa de Antonio'. 'Merodeaba por donde Antonio' becomes 'Merodeaba por la casa de Antonio'. 'El banco está donde la fuente' becomes 'El banco está cerca de la fuente'.

ADV No. For example: 'Está donde lo dejaste', 'La tienda donde te llevo está cerca', 'Esa es la calle donde nací'.

Entonces:

In the context of the sentence, the word 'entonces'...

SCONJ Can be replaced by 'por lo tanto'? (Example: 'Lo dice el periódico, entonces no puede ser mentira.')

NOUN Appears after 'aquel' and is used to refer to the past? (example: 'En aquel entonces....')

ADV None of the others. (Example: 'Se casará con él y entonces se irán a vivir a Francia.', 'La juventud de entonces era más responsable.', 'Si llegó ayer, entonces tendríamos que haberlo visto ya.')

Esa:

In the context of the sentence, can 'esa' be replaced by 'la' without changing the structure of the sentence?

DET Yes ('Esa silla está rota' becomes 'La silla está rota')

PRON No ('Esa no me gusta tanto' becomes 'La no me gusta tanto')

Esas:

In the context of the sentence, can 'esas' be replaced by 'las' without changing the structure of the sentence?

DET Yes ('Esas sillas están rotas' becomes 'Las sillas están rotas.')

PRON No ('Esas no me gustan tanto' becomes 'Las no me gustan tanto.')

Esos:

In the context of the sentence, can 'esos' be replaced by 'los' without changing the structure of the sentence?

DET Yes ('Esos folios están rotos' becomes 'Los folios están rotos.')

PRON No ('Esos no me gustan tanto' becomes 'Los no me gustan tanto.')

Ese:

In the context of the sentence, which of the following is true for 'ese'?

DET It precedes a noun and any adjectives it may have. For example: 'Ese CHICO es muy guapo', where chico is the noun.

PRON It is the subject or object of the sentence, and DOES NOT precede a noun phrase, for example: 'Ese es mi favorito.' or 'Me compras ese?')

Esta:

In the context of the sentence, can 'esta' be replaced by 'la' without changing the structure of the sentence?

DET Yes ('Esta silla está rota' becomes 'La silla está rota.')

PRON No ('Esta no me gusta tanto' becomes 'La no me gusta tanto.')

Estas:

In the context of the sentence, can 'estas' be replaced by 'las' without changing the structure of the sentence?

DET Yes ('Estas sillas están rotas' becomes 'Las sillas están rotas.')

PRON No ('Estas no me gustan tanto' becomes 'Las no me gustan tanto.')

Estos:

In the context of the sentence, can 'estos' be replaced by 'los' without changing the structure of the sentence?

DET Yes ('Estos folios están rotos' becomes 'Los folios están rotos.')

PRON No ('Estos no me gustan tanto' becomes 'Los no me gustan tanto.')

Este:

In the context of the sentence, which of the following is true for 'este'?

NOUN It refers to a cardinal point (norte, sur, este, oeste).

PRON It can be replaced by 'El chico' without changing the sentence's structure ('Este me gusta' becomes 'El chico me gusta.')

DET It cannot be replaced by 'El chico' without changing the sentence's structure ('Este computador es caro' becomes 'El chico computador es caro.')

La:

In the context of the sentence, would 'la' be translated in English as 'her' or 'the'?

DET THE ('La nia está corriendo' becomes 'The girl is running.')

PRON HER ('La dije que parase' becomes 'I told her to stop.')

Las:

In the context of the sentence, would 'las' be translated in English as 'them' or 'the'?

DET THE ('Las tarjetas se cayeron al suelo' becomes 'The cards spilled all over the ground.')

PRON THEM ('Las tomé ayer a las 5.' becomes 'I took them yesterday at 5.')

Los:

In the context of the sentence, would 'los' be translated in English as 'them' or 'the'?

DET THE ('Los síntomas empezaron inmediatamente.' becomes 'The symptons started immediately.')

PRON THEM ('Los vi yendo a coger el metro.' becomes 'I saw them going to the subway.')

Lo:

In the context of the sentence, what is the best translation of 'lo' in English?

PRON IT ('Lo vi' becomes 'I saw it.')

DET THE ('Lo mejor de todo...' becomes 'The best part...')

Menos:

In the context of the sentence, is 'menos':

NOUN ...used simply to refer to the math symbol of subtraction, and it appears after 'un'. For example 'Hay un menos delante del paréntesis'.

CONJ ...used as a math operator? For example 'Three minus two.'

ADP ...used to indicate an exception and 'menos' can be replaced by 'salvo' or 'excepto'?

ADJ ...used to mean less quantity of a noun. For example: 'Tiene menos interés'. 'Trajo cuatro tornillos menos.'

ADV None of the above.

Mucho:

In the context of the sentence, 'mucho'...

DET Can be replaced by 'un' without changing the sentence structure?

PRON Can be replaced by 'esto' without changing the sentence structure?

ADV Can be replaced by 'con intensidad', 'con frecuencia', 'demasiado tiempo', 'en gran cantidad'? Or it is followed by 'más' as in 'Eso es mucho más bonito'.

ADJ None of the above.

Nada:

In the context of the sentence, 'nada'...

VERB ...means 'to swim'.

PRON ...can be substituted for 'ninguna cosa', 'cualquier cosa', 'ninguna cantidad' or 'poco tiempo'.

ADV ...modifies an adjective or an adverb, e.g. 'Los ejercicios no eran nada fáciles'. 'Ese tren no va nada despacio.'

NOUN None of the above.

Otro:

In the context of the sentence, what would be the best translation for 'otro' in English?

DET 'Other' or 'Another'. For example: 'Otro chico me dijo lo mismo.' would translate as 'Another boy told me the same.'

PRON 'Another one'. For example: 'Otro no diría lo mismo' would translate as 'Another one would not say the same.'

Otros:

In the context of the sentence, what would be the best translation for 'otros' in English?

DET 'Other' or 'Another'. For example: 'Otros chicos me dijeron lo mismo.' would translate as 'Other kids told me the same.'

PRON 'Other people'. For example: 'Otros no dirían lo mismo' would translate as 'Other people would not say the same.'

Otra:

In the context of the sentence, what would be the best translation for 'otra' in English?

DET 'Other' or 'Another'. For example: 'Otra chica me dijo lo mismo.' would translate as 'Another girl told me the same.'

PRON 'Another one'. For example: 'Otra no diría lo mismo' would translate as 'Another one would not say the same.'

Otros:

In the context of the sentence, what would be the best translation for 'otros' in English?

DET 'Other' or 'Another'. For example: 'Otras chicas me dijeron lo mismo.' would translate as 'Other kids told me the same.'

PRON 'Other people'. For example: 'Otras no dirían lo mismo' would translate as 'Other people would not say the same.'

Para:

In the context of the sentence, is the word 'para' a form of the verb 'parar' (to stop)?

VERB Yes.

ADP No.

Que:

In the context of the sentence, 'que':

PRON Appears after 'uno', 'una', 'la', 'el', 'las', 'los' or 'lo'. For example: the sentence 'La que está en la película....', 'Lo que yo quiero', 'Busco una que me haga feliz.'

PRON What 'que' refers to can be replaced by 'el que', 'la que', 'los que', 'las que' or 'lo que'. For example: the sentence 'La estrella que está en la película....' can be modified by substituting the antecedent 'La estrella que' by 'La que', obtaining 'La que está en la película...'. Same with 'La mujer con que yo hablé' to 'Con la que yo hablé' and 'La casa que yo quiero' to 'La que yo quiero...'

SCONJ None of the above. For example: 'Mira que te lo dije.'

Qué:

In the context of the sentence, 'qué'...

DET Appears before a noun. For example: 'A qué AMIGO hay que llamar?', 'Qué VINO te gusta?', 'Me pregunto qué EDAD tendría su hijo.'

ADV It appears before an adjective or an adverb in an question context. For example: 'Qué DIFíCIL es!', 'Qué DESPACIO va!'

PRON None of the others. For example: 'Qué te gusta más?', 'Qué dices que es tan difícil? '

Sí:

In the context of the sentence, 'sí'...

INTJ ...means 'Yes'. Examples: 'Sí, traelo.', 'Te dijo que sí.'

PRON ...has the meaning of itself, himself, herself. For example: 'Se lo aplicó a sí mismo.'

NOUN ...means 'permission' or 'approval'. For example: 'Ya tengo el sí de mi padre.'

Toda:

Select the first correct option following the given order. In the context of the sentence, 'toda':

DET Appears before a noun or a noun phrase that matches in gender and number ('Toda la vida he estado esperando este momento.')

ADJ Can be substituted by 'entera' ('Se la comió toda.')

PRON None of the above. ('Toda es carísima.')

Todas:

Select the first correct option following the given order. In the context of the sentence, 'todas':

DET Appears before a noun or a noun phrase that matches in gender and number ('Todas las veces que me llamaste estaba fuera!')

ADJ Can be subsituted by 'enteras' ('Se las comió todas.')

PRON None of the above (‘Todas son más rápidas que yo.’)

Todos:

Select the first correct option following the given order. In the context of the sentence, ‘todos’:

DET Appears before a noun or a noun phrase that matches in gender and number (‘Todos los días me acuerdo de ti.’)

ADJ Can be substituted by ‘enteros’ (‘Se los comió todos’)

PRON None of the above. (‘Todos son más rápidos que yo.’)

Todo:

Select the first correct option following the given order. In the context of the sentence ‘todo’:

DET Appears before a noun or a noun phrase that matches in gender and number. For example: ‘Todo el tiempo me lo pasé pensando en ti.’

ADJ Can be substitued by ‘entero’. For example: ‘Se lo comió todo.’

PRON None of the above. For example: ‘Todo es carísimo.’

Una:

In the context of the sentence, ‘una’:

DET Appears before a noun that matches in gender and number. For example: ‘Una nia’, ‘unos nios’...

PRON Is the subject or the object of the verb. For example: ‘Una me dijo que me fuese cuanto antes’, ‘Díselo a una solo.’

NUM Refers to the number one. For example: ‘Son la una y media.’

Uno:

In the context of the sentence, 'uno':

NUM Means the number one. For example: 'Uno, dos, tres....', 'Solo tengo uno.'

PRON Is the subject or object of the sentence. For example: 'Uno me dijo que me fuese cuanto antes', 'Díselo a uno solo.'

Unos:

In the context of the sentence, 'unos':

DET Appears before a noun that matches it in gender and number. For example: 'Unos perros', 'unos nios'...

PRON Is the subject or object of the sentence. For example: 'Unos dijeron de quedar mas tarde.', 'Les di comida a unos y bebida a otros.'

NOUN Is the plural of the number one. For example: 'Los unos de cada mes va al médico.'

Hay:

In the context of the sentence, 'hay' is used as an auxiliary verb as in 'hay que + verb':

AUX Yes. For example: 'Hay que hacerlo cuanto antes.', 'Hay que encontrar una solución.'

VERB No. For example: 'Hay dos opciones.'

Ni:

In the context of the sentence, 'ni':

CONJ Would be translated as 'neither' 'nor' or 'or' as in 'Ni Juan, ni Pedro ni Felipe te darán la razón.' (here 'ni' can be translated as 'Neither Juan, nor Pedro, nor

Felipe will admit you're right.'), or 'No descansa ni de día ni de noche.' (here the best translation would be 'He doesn't rest at day or at night.')

ADV It's more closely translated as 'even' or 'not even'. For example: 'No descansé ni un minuto' – 'I didn't rest even for a minute.'

Más:

In the context of the sentence, 'más':

CONJ ...being used as mathematical operator? For example: 'Dos más dos son cuatro.'

NOUN ...means symbol of the sum operation? For example: 'En esta suma falta el más.'

ADV ...denotes superiority and appears before adjectives and adverbs. For example: 'Ve más rápido', 'l está más gordo.'

ADJ ...denotes more quantity of something and appears before nouns. For example: 'Pon más pan en la mesa.'

PRON None of the above. In general means more of a quantity or quality but it does not appear before a noun, adjective or adverb. For example: 'He comprado más.'

Any:

In the context of the sentence, 'any':

DET Means 'cualquier', 'cualquiera', 'algún', 'alguna', 'ningún' or 'ninguna' in Spanish, and is ALWAYS followed by a noun. For example: 'Do you have any bread?', 'I'll watch any film', 'I'll take any leftovers.', 'I didn't watch any film.'

ADV Means 'at all'. For example:''He wasn't any good at soccer.'

PRON None of the above. For example: 'Have you met any of my friends?', 'Any will do.'

Anything:

In the context of the sentence, 'anything' means 'nada'.

ADV Yes. For example: 'She is not anything like him.'

PRON No. For example: 'Anything will do.', 'I'll do anything to prove it.'

Away:

In the context of the sentence, 'away' is used as 'estar fuera' or 'estar de viaje'.

ADJ Yes. For example: 'My father is away in Chicago.'

ADV No. For example: 'He walked away after seeing the price.', 'It's two miles away.'

Enough:

In the context of the sentence, 'enough' would be best translated as:

ADV 'Lo suficiente' or 'Lo suficientemente'. For example: 'He worked enough to pay for his college bills' or 'He is fast enough to arrive here on time.'

INTJ 'Basta!'. For example: 'Enough! I don't want to listen to you anymore!'

ADJ 'Suficiente' or 'bastante' AND appears before a noun. For example: 'Do we have enough money?'

PRON 'Suficiente' or 'bastante' AND DOES NOT appear before a noun. For example: 'Do we have enough?'

Even:

In the context of the sentence, 'even' would be best translated as:

ADV 'Aún', 'hasta' or 'ni siquiera'. For example: 'I feel even worse than you.', 'Even a child could do it.'

VERB 'Nivelar', 'aplanar' or 'allanar' algo. For example: 'They used a roller to even the lawn.'

ADJ None of the above. For example: 'The first half of the match was fairly even', 'Any number doubled is even.'

Here:

In the context of the sentence, 'here':

ADV Is best translated as 'aquí', 'ahora'. For example: 'He's still here with us', 'What we need to do here is this.'

INTJ Is used to attract someone's attention. For example: 'Here, let me hold it.'

Inside:

In the context of the sentence, 'inside':

ADV Means 'indoors'. For example: 'I stay inside when it rains.'

ADP Means 'in the interior of'. For example: 'He stayed inside the plane', 'He felt it deep inside himself.'

NOUN Means 'interior'. For example: 'The inside of the house is beautiful.'

ADJ Means 'positioned on the inside'. For example: 'Those jeans have an inside pocket.'

Like:

In the context of the sentence, 'like':

VERB Corresponds to the verb 'to like'. For example: 'I like hamburgers.'

ADV Is used to quote someone, usually preceded by the verb to be. For example: 'And he was like, I don't know what to say!'

ADV Is used as filler and can be deleted from the sentence without changing its meaning. For example: 'And I am there LIKE totally lost LIKE waiting for someone to help me out.'

NEXT None of the above.

In that case, maybe 'like':

ADP Can be substituted for 'similar to'. For example: 'He used to have a car like mine.'

SCONJ Can be substitued for 'as if', 'in the same way that'. For example: 'I felt like I'd been kicked by a camel', 'People that change partners like they change clothes.'

NEXT None of the above.

In that case, maybe 'like'

ADP Can be substituted for 'for example' or 'such as'. For example: 'There are lots of birds, like ducks and gulls.'

ADJ Can be substituted for 'similar'. For example: 'My partner and I have like minds.'

NOUN None of the above. For example: 'Tell me your likes and dislikes.', 'It was something the likes of which I had never seen before.'

Not:

In the context of the sentence, 'not':

PART Is used for verb negation. For example: 'I did not go to Paris.', 'I am not going to study for tomorrow's final.'

ADV None of the above. For example: 'Not a single attempt was made to fix it.'

Nothing:

In the context of the sentence, 'nothing':

ADV Means 'not at all'. For example: 'He looks nothing like his father.'

NOUN Means 'a person or thing of no importance'. For example: 'He is nothing to me now.'

PRON None of the above. For example: 'Theres nothing you can do.'

Down:

In the context of the sentence, 'on' means 'encendido', 'en pie' or 'abierto' in Spanish?

ADJ Yes. For example: 'The computer is on.', 'The party is still on', 'The festival is on all week long.'

NEXT No. For example: 'The apple is on the table,' 'Put it on!'

Does an object immediately follow the word 'on'?

ADP Yes, there is an object right after 'on'. For example: 'I put it on THE TABLE.' – the object 'the table' immediately follows 'on'.

NEXT No, the object is not right after 'on', or there is no object. For example: in 'The table he put it on was green.' the object is not right after 'on'; in 'What is going on?' 'on' has no object.

Is the object of the preposition 'on' located earlier in the sentence? For example, in 'The table he put it on was green.', the preposition 'on' refers to the object 'the table' earlier in the sentence.

ADP Yes.

NEXT No.

Can you place an adverb describing how something is done (such as 'calmly', 'quietly', 'easily', 'quickly', etc.) between 'on' and its associated verb? For example, introducing the adverb 'steadily' in the sentence 'She burbled (steadily) on.'

ADV Yes.

ADP No.

Off:

In the context of the sentence, 'off' means 'apagado', 'distinto', 'raro' or 'erroneo'.

ADJ Yes. For example: 'The computer is off.', 'Something feels off.', 'Calculations were off by a hundred.'

NEXT No. For example: 'That's off limits.', 'Switch it off now!'

Does the object of the preposition 'off' immediately follow the word 'off'?

ADP Yes, there is an object right after the preposition. For example: 'He took off the shirt' – the object 'the shirt' immediately follows the preposition 'off'.

NEXT No, the object is not right after the prepostion, or there is no object. For example: 'When did he log off?' – 'off' has no object.

Is the object of the preposition 'off' located earlier in the sentence? For example, in 'The shirt he took off was yellow.', the preposition 'off' refers to the object 'the shirt' earlier in the sentence.

ADP Yes.

NEXT No.

Can you place an adverb describing how something is done (such as 'calmly', 'quietly', 'easily', 'quickly', etc.) between 'off' and its associated verb? For example, introducing the adverb 'quickly' in the sentence 'The man ran (quickly) off'

ADV Yes.

ADP No.

Once:

In the context of the sentence, 'once' can be substituted by 'as soon as' or 'when':

SCONJ Yes. For example: 'We'll get a move on once we find the keys!'

ADV No. For example: 'They deliver once a week.', 'She was once the best opera singer alive.'

Out:

In the context of the sentence, 'out':

ADJ Means 'not at home', 'revealed' or 'made public'. For example: 'He's out since this morning.', 'The secret was soon out.'

VERB Is a verb. For example: 'There is not reason to out a closeted politician.'

NOUN Is a noun. For example: 'They gave me an out.'

NEXT None of the above.

Does the object of the preposition 'out' immediately follow the word 'out'

ADP Yes, there is an object right after the preposition. For example: 'I walked out the building.' – the object 'the building' immediately follows the preposition 'out'.

NEXT No, the object is not right after the prepostion, or there is no object. For example: 'Why don't you get out?' – 'out' has no object.

Is the object of the preposition 'out' located earlier in the sentence? For example, in 'The fire the firemen put out was reported by a neighbor.', the preposition 'out' refers to the object 'the fire' earlier in the sentence.

ADV Yes.

NEXT No.

Can you place an adverb describing how something is done (such as 'calmly', 'quietly', 'easily', 'quickly', etc.) between 'out' and its associated verb? For example, introducing the adverb 'quickly' in the sentence 'I walked (quickly) out'

ADV Yes.

ADP No.

Outside:

In the context of the sentence, 'outside':

ADV Means 'outdoors'. For example: 'I slept outside last night.', 'The dog is barking outside.'

ADJ Means 'exterior' (noun). For example: 'The outside lights are turned off.'

NOUN Means 'external side'. For example: 'The outside of the house needs to be painted.'

ADP None of the above. For example: 'There was a boy outside the door.'

Something:

In the context of the sentence, 'something' means 'somewhat' or 'to a degree':

ADV Yes. For example: 'My back hurts something terrible', 'The baby looks something like his father', 'There was something close to a million dollars.'

PRON No. For example: 'Something that I like...', 'He whispered something that I could not hear.'

Somewhere:

In the context of the sentence, 'somewhere' would be best translated in Spanish as:

ADV 'En alguna parte'. Notice the inclusion of the word 'EN'. For example: 'I've seen you somewhere before' would be best translated as 'Te he visto antes en alguna parte.'

PRON 'Algún sitio.' For example: 'In search of somewhere to live.' would be best translated as 'En busca de algún sitio para vivir.'

Though:

In the context of the sentence, 'though' would be best translated in Spanish as:

ADV 'Sin embargo'. For example: 'I was hunting for work. Jobs were scarce though' would be translated as 'Estaba buscando trabajo. Sin embargo había poco.'

SCONJ 'Aunque'. For example: 'Though they were whispering, I could hear them' would be translated as 'Aunque estaban susurrando, les podía oír.'

Appendix B

Question Tree for Part-of-Speech Tagging Disambiguation

B.1 Question Tree for Spanish Tokens

Node **PRE-START**: In the context of the sentence, is the word 'token'...?

PROPN A proper noun or part of a proper noun. Proper nouns can be names for people ('Juan', 'Jessica'), places ('Francia', 'Nueva York', 'Everest', 'Hudson'), objects ('Páginas Amarillas') brands or companies ('Naciones Unidas', 'Apple', 'Google', 'Coke'), days of the week ('Lunes', 'Martes'), months of the year ('Enero', 'Febrero',...)

INTJ A single-word used as an exclamation that expresses an emotional reaction ('Sí!', 'Qué?!', 'Mierda!', 'Wow', 'Gracias!') and may include a combination of sounds not found in the language (eg. 'mmhm', 'huh', 'psst', etc)

start None of the above.

Node **START**: In this context, 'token' is a(n):

NOUN Noun, because it names a thing ('mesa'), animal ('perro'), places ('tienda'), events ('verano') or ideas ('amor').

ADJ Adjective, because it says something about the quality ('la mesa AZUL'), quantity ('MS mesas') or the kind of the noun or pronoun it refers to.

verb-inf Verb, because it is used to demonstrate an action or a state of being.

ADV Adverb, because it describes the how, where, when, or the degree at which something is done. It modifies a verb (e.g. 'ven rápidamente'), adjective (e.g. 'completamente quieto'), clause (e.g. 'Sorprendentemente, sí qui lo hizo.'), or another adverb (e.g. 'muy bien').

Node **VERB-INF**: Does the word 'token' end in -ar, -er or -ir?

aux-start Doesn't end in -ar, -er or -ir. For example: 'estoy', 'eres', 'venimos.'

verb-noun Ends in -ar, -er or -ir. For example: 'estar', 'ser', 'venir.'

Node **VERB-NOUN Disambiguation**: In the context of this sentence, can the verb 'token' be preceded by an article like 'el'?

NOUN Yes. As in: 'El deber me llama.', 'El querer ir no es suficiente.'

aux-start No.

Node AUX-START **Auxiliary Verb Detection** The verb 'token'...?

VERB ...appears isolated from another verb. For example: 'VENGO en son de paz.'

periph ...appears alongside another verb, separated by a word particle like 'de', 'a', 'que', etc. For example: 'HE de DECIR que no me gusta la idea.', 'VINIERON a APAGAR las luces.', 'TENGO que DECIR algo importante.'

aux-final ...appears directly attached to another verb. For example: 'HE VISTO de todo.', 'ESTOY VINIENDO tan deprisa como puedo.'

Node **PERIPHRASIS** (Periphrasis Detection): Does the verb 'token' appear before the preposition or conjunction?

AUX It appears before. For example: 'HE de decir que no me gusta la idea.', 'VINIERON a apagar las luces.', 'TENGO que decir algo importante.'

VERB It appears afterwards. For example: 'He de DECIR que no me gusta la idea.', 'Vinieron a APAGAR las luces.', 'Tengo que DECIR algo importante.'

Node AUX-FINAL **Auxiliary Verb Disambiguation**: Is the verb 'token'...

VERB ...the second of the two verbs as in 'Ya he ESTADO ahí.', 'Sabes si está LLOVIENDO?'

AUX ...the first of the two verbs as in 'Ya HE estado ahí.', 'Sabes si EST lloviendo?'

B.2 Question Tree for English Tokens

Node **PRESTART:**
In the context of the sentence, is the word 'token'...?

PROPN A proper noun or part of a proper noun. Proper nouns can be names for people ('John', 'Jessica'), places ('France', 'New York', 'Everest', 'Hudson'), objects ('Yellow Pages') brands or companies ('United Nations', 'Apple', 'Google', 'Coke'), days of the week ('Monday', 'Tuesday'), months of the year ('January', 'February',...)

INTJ A single-word used as an exclamation that expresses acknowledgement or an emotional reaction ('Yes!!!', 'What?!', 'F*ck!', 'Wow', 'Please') and may include a combination of sounds not found in the language (eg. 'mmhm', 'huh', 'psst', etc)

start None of the above.

Node **START:**

In this context, 'token' is a(n):

NOUN Noun, because it names a thing ('table'), animal ('dog'), places ('shop'), events ('summer') or ideas ('love').

ADJ Adjective, because it says something about the quality ('the BLUE table'), quantity ('MORE tables') or the kind of the noun or pronoun it refers to.

verb-ing Verb, because it is used to demonstrate an action or a state of being.

ADV Adverb, because it tells the how, where, when, or the degree at which something is done. It modifies a verb (e.g. 'come QUICKLY', 'go HOME'), adjective (e.g. 'COMPLETELY lifeless'), clause (e.g. 'SURPRISINGLY, he did it.'), or another adverb (e.g. 'VERY nicely').

Node **VERB-ING :**

Does the verb 'token' end in -ing? If it does, is this -ing a suffix (e.g. 'walk-ing', 'travel-ing') and not only a verb like 'bring' or 'sing'?

aux-start Doesn't end in -ing.

verb-start Ends in -ing but it's not a suffix like in 'bring' or 'sing'.

verb-noun-adj Ends in -ing and it's a suffix like in 'walk-ing' and 'travel-ing'.

Node **VERB-NOUN-ADJ:**

Could the verb 'token' actually be a Noun or Adjective?

verb-noun It could be a Noun. For example, 'running' and 'reading' can be verbs or nouns depending on the context.

verb-adj It could be an Adjective. For example: 'stunning' and 'crushing' can be verbs or adjectives depending on the context.

VERB No, it's definitely a verb.

Node **VERB-NOUN:**

In the context of this sentence, can the word ''token'...?

VERB be modified by an adverb and cannot be pluralized?

NOUN be pluralized? For example: 'reading' and 'readings.'

NOUN be modified by an adjective like 'good' or 'first'? For example: 'first reading.'

NOUN be preceded by one or more nouns? For example: 'road running.'

Node **VERB-ADJ:**

In the context of this sentence, can the word 'token' either be preceded by a degree adverb, such as 'very' or 'extremely', or can it take the prefix 'un-' and have the opposite meaning?

ADJ YES.

VERB NO.

Node **ADJ-START:**

Could 'token' be a noun or a verb?

ADJ-NOUN It could be a noun. For example: 'fun' can be a noun, as in 'That was a lot of fun', or an adjective, as in 'That was a fun trip!'.

ADJ-VERB It could be a verb. For example: 'surprised' can be a verb, as in 'He surprised me', or an adjective, as in 'I am very surprised'.

ADJ No, it's definitely an adjective.

Node **ADJ-NOUN:**

In the context of the sentence, the word 'token'...

PROPN Is a proper noun that serves the role of an adjective. For example: 'Chinese' in 'I bought Chinese food.'

ADJ CAN be modified by the adverbs 'very' or 'really'. For example: 'A fun trip.' to 'A very fun trip.'

NOUN CANNOT be modified by the adverbs 'very' or 'really'. For example: 'A dark brilliant.' to 'A dark very brilliant.'

ADJ-VERB:

In the sentence, can 'token'...

ADJ be modified by the adverbs 'really' or 'very'. For example: 'I am surprised' to 'I am very surprised.'

ADJ reference a state as opposed to an event. For example: 'At that time, I was married.'

VERB a reference to an event or action. For example: 'I was married on a Sunday.'

AUX START:

Is 'token' a form of the verbs 'to be', 'to have', 'to do' and 'to get'?

VERB It is a form of 'to be', 'to have', 'to do' or 'to get' and it appears isolated from another verb. For example: 'I was happy,' 'I have so much to study,' 'I do it all the time,' 'I got it!'

aux-final It is a form of to be, to have, to do or to get and it appears alongside another verbal form, acting either as auxiliary or main verb. For example: 'I WAS told

to come.', 'I HAVE gone there many times!', 'I HAVE DONE it already.', 'DO you think so?', 'DID you DO it?', 'I DIDn't DO it.', 'DOn't push it!', 'I GOT to go.'

verb-start No.

AUX-FINAL :

Does the form 'token'...

VERB ...act as the main verb of the compound verb as in 'I have BEEN there.', 'I have HAD that feeling before', 'I was DONE when you arrived.', 'I've GOT so much to do.'

AUX ...act as an auxiliary verb to the main verb as in 'I WAS told to come.', 'I HAVE loved you since the first day.', 'DO you think so?', 'DOn't push it!', 'I GOT to go.'

Node **VERB-START :**

In the context of the sentence, 'token' is in some form of the past tense.

participle Yes.

VERB No.

Node **PARTICIPLE:**

If you replaced 'token' with a form of 'to see', 'to give', or 'to know', would that form be: (ignore the change in meaning)

VERB Saw, gave or knew.

ADJ-VERB Seen, given or known.

Appendix C

List of Automatically Tagged Words

C.1 List of Automatically Tagged Words in English

ADJ: due, for, many, most, non, such.

ADP: despite, during, outta, per, regarding.

ADV: ahead, anyhow, anymore, aside, eventually, ever, everytime. forwards, how, however, later, meanwhile, rather, sooner, therefore, whence, where, whereby, wherein, whereupon, why.

AUX: ca (from can't), could, 'd, 'll, must, ought, shall, should, would.

CONJ: and, nor, or.

DET: another, each, every, its, my, our, some, the, their, your.

NOUN: data, maximum, minimum, people, plenty.

NUM: gazillion, sixteen.

PART: n't, '.

PRON: anybody, anyone, everyone, he, herself, hers, him, himself, I, it, itself, mine, myself, ones, ourselves, ours, she, somebody, someone, theirs, them, themselves, they, us, we, whom, who, whose, you, yourself, yourselves, yours.

SCONJ: although, cuz, whereas, whether.

C.2 List of Automatically Tagged Words in Spanish

ADJ: buenos, ciertas, cierto, distintas, juntos, misma, mismas, mismito, mismo, mismos, primer, quinto, segunda, sexto, tercera, tercer, tercero, varia.

ADP: a, a+ (from al), con, de, durante, en, entre, hasta, in, pa, per, por, según, sin.

ADV: abajo, adónde, ahorita, alla, allí, alrededor, apenas, arriba, así, aun, casi, dentro, después, detrás, siquiera, sólo, todavía, ya.

CONJ: e, o, pero, sino, u, y.

DET: aquella, cualquier, cuanta, cuánta, +el (from 'del' and 'al'), el, mi, mis, mucha, su, sus, tanta, tantas, tantos, tu, tus, unas, un.

INTJ: aló.

NOUN: contras, día, ele, gente, ochos, repente, súper, través, vez.

NUM: cero, ciento, cientos, cuatrocientos, doscientos, ochocientos, quinientos, seiscientos, trescientas, trescientos.

PRON: alguien, bastantes, cuáles, cuantos, ella, ellas, ello, ellos, él, ésa, ésas, ése, ésta, éstas, éste, esto, +la, +las, +le, le, +les, les, +lo, +los, +me, nosotros, nosotras, +nos, +os, os, quién, +se, se, +te, tí, tú, vosotras, vosotros, yo.

SCONJ: aunque, porque, pues, si.

VERB: creo, dice, hacer, ir, sabes, sé.

Appendix D

List of Manually Tagged Words

D.1 List of Manually Tagged Words in English

- above ADP ADV
- across: ADP ADV
- after: ADP ADV SCONJ
- against: ADP ADV
- alike: ADJ ADV
- along: ADP ADV
- anytime: ADV INTJ
- anyway: ADV INTJ
- anyways: ADV INTJ
- anywhere: ADV PRON
- away: ADV INTJ ADJ
- before: ADP ADV SCONJ
- behind: ADP ADV NOUN
- below: ADP ADV
- besides: ADP ADV
- between: ADP ADV
- beyond: ADP ADV NOUN
- by: ADP ADV
- considering: ADP SCONJ ADV
- else: ADJ ADV
- enough: INTJ ADJ ADV PRON
- everyday: ADJ ADV NOUN
- except: ADP SCONJ
- in: ADP ADV

- including: ADP VERB
- inside: NOUN ADP ADV ADJ
- nobody: PRON NOUN
- none: PRON ADV
- once: ADV SCONJ
- opposite: ADP ADV NOUN ADJ
- otherwise: ADV ADJ
- outside: NOUN ADJ ADV ADP
- p.m.: PART
- self: ADJ NOUN PRON
- since: ADP ADV SCONJ
- somewhere: ADV PRON
- than: ADP SCONJ
- though: SCONJ ADV
- through: ADP ADV ADJ
- till: ADP SCONJ
- under: ADP ADV
- underneath: ADV ADP ADJ NOUN
- unlike: ADJ ADP
- until: ADP SCONJ
- upon: ADP ADV
- upside: NOUN ADP
- whenever: ADV SCONJ
- while: NOUN SCONJ
- without: ADP ADV

D.2 List of Manually Tagged Words in Spanish

- alguna: DET PRON
- algunas: DET PRON
- alguno: ADJ DET PRON
- algunos: DET PRON
- alto: ADJ ADV
- antes: ADV ADJ
- aparte: ADV ADJ
- aquel: PRON DET
- bajo: ADP ADJ NOUN ADV
- bastante: PRON DET ADV ADJ
- cerca: ADV NOUN
- contra: ADP NOUN
- cual: PRON ADV
- cuál: PRON DET

- cualquiera: ADJ PRON NOUN
- cuantas: DET PRON
- cuántas: PRON DET
- cuanto: DET ADV PRON
- cuánto: PRON DET ADV
- cuántos: PRON DET
- demás: ADJ PRON
- demasiado: ADJ ADV PRON
- demasiados: ADJ PRON
- inclusive: ADV ADJ
- incluso: ADV ADJ
- junto: ADV ADJ
- mía: PRON ADJ
- mías: PRON ADJ
- millones: NUM NOUN
- millón: NUM NOUN
- mío: PRON ADJ
- míos: PRON ADJ
- muchas: DET PRON
- muchísimo: ADJ ADV
- muchos: DET PRON
- nuestra: DET ADJ PRON
- nuestro: DET ADJ PRON
- nuestros: DET ADJ PRON
- poca: DET PRON
- pocas: DET PRON
- poco: DET ADV NOUN PRON
- pocos: DET PRON
- primera: ADJ NOUN
- primeras: ADJ NOUN
- primero: ADJ ADV NOUN
- primeros: ADJ NOUN
- solo: ADJ ADV
- tal: ADJ/DET
- tanto: DET ADV NOUN PRON
- tuya: ADJ PRON
- tuyas: ADJ PRON
- tuyo: ADJ PRON
- tuyos: ADJ PRON
- varias: DET PRON
- varios: DET PRON