LinCE: A Centralized Benchmark for Linguistic Code-switching Evaluation

Gustavo Aguilar, Sudipta Kar, Thamar Solorio
University of Houston
Department of Computer Science
{gaguilaralas, skar3, tsolorio}@uh.edu

Abstract

Recent trends in NLP research have raised an interest in linguistic code-switching (CS); modern approaches have been proposed to solve a wide range of NLP tasks on multiple language pairs. Unfortunately, these proposed methods are hardly generalizable to different code-switched languages. In addition, it is unclear whether a model architecture is applicable for a different task while still being compatible with the code-switching setting. This is mainly because of the lack of a centralized benchmark and the sparse corpora that researchers employ based on their specific needs and interests. To facilitate research in this direction, we propose a centralized benchmark for **Lin**guistic **C**ode-switching **E**valuation (**LinCE**) that combines ten corpora covering four different code-switched language pairs (i.e., Spanish-English, Nepali-English, Hindi-English, and Modern Standard Arabic-Egyptian Arabic) and four tasks (i.e., language identification, named entity recognition, part-of-speech tagging, and sentiment analysis). As part of the benchmark centralization effort, we provide an online platform at `ritual.uh.edu/lince`, where researchers can submit their results while comparing with others in real-time. In addition, we provide the scores of different popular models, including LSTM, ELMo, and multilingual BERT so that the NLP community can compare against state-of-the-art systems. LinCE is a continuous effort, and we will expand it with more low-resource languages and tasks.

Keywords: code-switching, code-mixing, multilingualism, benchmark

1. Introduction

Linguistic code-switching[1] (CS) is the multilingual phenomenon that happens when speakers alternate languages within the same sentence or utterance. During the last decade, the CS phenomenon has attracted more research interest from the NLP community. Many researchers have proposed novel methods to handle code-switched data, showing improvements on core NLP tasks such as language identification (LID), named entity recognition (NER), and part-of-speech (POS) tagging. However, many of these approaches are usually evaluated on a few language pairs and a specific domain, and it is not clear whether these models are exclusive to such scenarios or they can generalize to other tasks, domains, and language pairs.

Moreover, research in code-switching currently has a slow process of comparison in which researchers have to replicate previous methods to report scores on different datasets. Furthermore, choosing the best-published model for benchmarking purposes is not an easy task either. These problems exist mainly because 1) there is no official benchmark for general code-switching evaluation that allows direct comparisons across multiple tasks, and 2) methods are usually not comprehensively evaluated across datasets with different language pairs.

To overcome these problems, we propose a centralized **Lin**guistic **C**ode-switching **E**valuation (**LinCE**) benchmark. We have consolidated a benchmark from preexisting corpora considering the following aspects: 1) multiple language pairs from high- and low-resource languages with a reasonable range of code-mixing indexes (CMI) (Gambäck and Das, 2014), 2) typologically-diverse languages[2], 3) a variety of NLP tasks including core tasks and downstream

Language Pair	LID	POS	NER	SA
Spanish-English	✓	✓	✓	✓
Hindi-English	✓	✓	✓	-
Nepali-English	✓	-	-	-
MS Arabic-Egyptian Arabic	✓	-	✓	-

Table 1: Overview of the LinCE language pairs and tasks.

applications, and 4) different code-switching domains from social media platforms. LinCE is comprised of four LID datasets, two POS tagging datasets, three NER datasets, and one sentiment analysis (SA) dataset, providing a total of ten datasets (see Table 1). Furthermore, an important contribution of LinCE is the new stratification process to provide fair and, in some cases, official splits for the tasks at hand. This required a careful inspection of the original datasets from which we list five major issues (see Section 4.5) and propose new splits for nine out of the ten datasets.

In addition to the LinCE benchmark, we also provide strong baselines using popular models such as LSTMs (Hochreiter and Schmidhuber, 1997), ELMo (Peters et al., 2018), and multilingual BERT (Devlin et al., 2018). In our analysis, we evaluate the results of the best model and describe the outstanding challenges in this benchmark. Moreover, LinCE is publicly available at `ritual.uh.edu/lince`, and we anticipate this benchmark to continue to grow and include new tasks and language pairs as they become available. We hope that LinCE motivates future work and accelerates the progress on NLP for code-switched languages.

2. Related Work

Linguistic code-switching has been studied in the context of many NLP tasks (Sitaram et al., 2019), including language identification (Solorio et al., 2014; Bali et al., 2014), part-

---

[1] We use code-switching and code-mixing equivalently.

[2] We also consider the geolocation of such languages to account for places across the world.

of-speech tagging (Soto and Hirschberg, 2018; Soto and Hirschberg, 2017; Molina et al., 2016; Das, 2016; Solorio and Liu, 2008), named entity recognition (Aguilar et al., 2018), parsing (Partanen et al., 2018), sentiment analysis (Vilares et al., 2015), and question answering (Raghavi et al., 2015; Chandu et al., 2018). Many code-switching datasets have been made available through the shared-task series FIRE (Sequiera et al., 2015b; Choudhury et al., 2014; Roy et al., 2013) and CALCS (Solorio et al., 2014; Molina et al., 2016; Aguilar et al., 2018), which have focused mostly on core NLP tasks. Additionally, other researchers have provided datasets for dialect recognition (Hamed et al., 2018), humor detection (Khandelwal et al., 2018), sub-word code-switching detection (Mager et al., 2019), among others. Despite the availability and recent growth of datasets, it is still unclear how to compare models across language pairs, domains, and general language processing tasks.

In the case of language identification (LID) at the token level, researchers have evaluated approaches such as conditional random fields (CRF) with hand-crafted features (Al-Badrashiny and Diab, 2016), LSTM models with word and character embeddings (Mave et al., 2018; Samih et al., 2016), code-mixed word embeddings (Pratapa et al., 2018), and transfer learning (Aguilar and Solorio, 2019). While most of these approaches reach over 90% of accuracy regardless of the language pairs, it is hard to determine which model is the best overall and what the trade-offs are by using one instead of the others. Likewise, for part-of-speech (POS) tagging, the community has explored tools that heavily rely on monolingual hand-crafted linguistic information and morphological features (AlGhamdi et al., 2019), traditional ML techniques (e.g., SVM) with heuristics that exploit monolingual resources (Solorio and Liu, 2008), combined monolingual taggers including CRF and Random Forest (Jamatia et al., 2015), and jointly modeling POS tagging with LID using recurrent neural networks (Soto and Hirschberg, 2018). Although such approaches are effective on their datasets at hand, they are language-specific and not easy to compare across each other.

A slightly different trend has been marked in named entity recognition (NER). Although the main problem in NER has been the lack of datasets, it is until recently that researchers have provided a few corpora on Hindi-English (Singh et al., 2018b), Spanish-English and Modern Standard Arabic-Egyptian Arabic (Aguilar et al., 2018). The participants of the 2018 CALCS competition proposed models based on standard neural NER architectures (e.g., character CNN, followed by a word-based LSTM, and CRF) (Geetha et al., 2018), including variations with attention (Wang et al., 2018b) and multi-task learning (Trivedi et al., 2018). Additionally, most of the participants exploited publicly available resources such as gazetteers as well as monolingual and multilingual embeddings (Winata et al., 2018). While the CALCS competition provided datasets on Spanish-English and Modern Standard Arabic-Egyptian Arabic simultaneously, the participants were allowed to provide predictions on one or both competitions. This flexibility left the question open regarding which model was overall the best across language pairs.

Sentiment analysis (SA) on code-switched data has not been explored extensively either. Vilares et al. (2015) provided a Spanish-English polarity annotations for a small section of the 2014 CALCS LID corpus. Their focus was to compare different *bag-of-words* features using L2-regularized logistic regression models often employed in monolingual SA. Concurrent to our work, SemEval-2020 is hosting the first competition for SA in code-switched data, *Task 9: Sentiment Analysis for Code-Mixed Social Media Text* (Patwa et al., 2020), which covers Spanish-English and Hindi-English with tweets annotated with both sentiment and language identification labels. We adopt the Spanish-English corpus as part of the LinCE benchmark.

Although there has been progress in code-switching overall, CS is still lacking advancements in many NLP tasks. Additionally, CS tends to advance guided by language-specific challenges, usually providing sparse technologies that may not necessarily be effective for other language pairs. By gathering different language pairs and tasks into a single benchmark, we expect LinCE to strive for consolidated and steady progress in code-switching research.

3. Linguistic Challenges

Although code-switching can happen in more than two languages, this benchmark focuses on language pairs only. The frequent alternations between two languages is precisely what makes the automated processing of code-switching data difficult. We quantify such complexity using the CMI index proposed by Gambäck and Das (2014) as shown in Table 2. The higher the CMI index, the more alternations the dataset contains, and hence, the more complex the code-switching behavior is. In addition to the alternation of languages, we briefly describe other linguistic challenges that each specific language pair poses to current NLP systems:

- **Spanish-English (SPA-ENG)**. While English is a Germanic language, a significant number of words from its current vocabulary have been borrowed from Latin and French since the Middle Ages (Tristram, 1999). This particular set of words tends to overlap with words from Spanish, a Latin-based language. This overlap increases ambiguity and directly affects systems that rely on character-based approaches, for example, in the case of language identification. Code-switching also appears within the words, often inflecting words by conjugating English verbs using Spanish grammatical rules. This behavior is known as Spanglish (Rothman and Rell, 2007), and it particularly affects non-contextualized word embeddings as it increases the out-of-vocabulary (OOV) rate.

- **Hindi-English (HIN-ENG)**. One of the most challenging aspects of this language pair is the lack of a standardized transliteration system. Speakers transliterate Hindi employing mostly ad-hoc phonological rules to use the English alphabet when writing. Using the same roman alphabet makes code-switching more convenient but the lack of an official standard for transliteration makes it difficult to process with existing resources exclusively available for Hindi with

Task	Corpus	Languages	All Posts	All CMI	CS Posts	CS CMI	Lang1	Lang2	All Tokens
LID	Molina et al. (2016)	SPA-ENG	32,651	8.29	12,380	21.86	129,065	170,793	390,953
	Solorio et al. (2014)	NEP-ENG	13,011	19.85	10,029	25.75	59,037	78,360	188,784
	Mave et al. (2018)	HIN-ENG	7,421	10.14	3,317	22.68	84,752	29,958	146,722
	Molina et al. (2016)	MSA-EA	11,243	2.82	1,326	23.89	140,057	40,759	227,354
POS	Singh et al. (2018b)	HIN-ENG	1,489	20.28	1,077	28.04	12,589	9,882	33,010
	Soto and Hirschberg (2017)	SPA-ENG	42,911	24.19	41,856	24.81	178,135	92,517	333,069
NER	Aguilar et al. (2018)	SPA-ENG	67,223	5.49	17,466	21.16	163,824	402,923	808,663
	Singh et al. (2018a)	HIN-ENG	2,079	19.99	1,644	25.28	13,860	11,391	35,374
	Aguilar et al. (2018)	MSA-EA	12,335	–	–	–	–	–	248,478
SA	Patwa et al. (2020)	SPA-ENG	18,789	20.70	18,196	21.37	65,968	144,533	286,810

Table 2: The CMI scores and the number of tokens across corpora. **All Posts** describes the number of posts in the corpora and **All CMI** is the corresponding CMI scores for such samples. Similarly, **CS Posts** denotes the number of code-switched posts (excluding monolingual posts) and **CS CMI** is the corresponding CMI scores for such samples. We also show the number of tokens that belong to the language pairs (**Lang1**, **Lang2**) as well as the overall number of tokens (**All Tokens**), which includes other LID labels beyond the language pairs. English is the **Lang1** class for English-paired languages; for MSA-EA, Modern Standard Arabic is the **Lang1** class. We omit the CMI information for the MSA-EA NER corpus because the corpus does not come with language identification labels.

the Devanagari script. Furthermore, although Hindi loosely follows the subject-object-verb (SOV) structure, its flexible word order poses an additional challenge to NLP systems.

- **Nepali-English (NEP-ENG).** Similar to HIN-ENG, Nepali is transliterated using the English alphabet when code-switched with English. This behavior makes Nepali speakers to write driven by arbitrary phonological rules that allow the romanization of Nepali using the English alphabet, which excludes the few monolingual resources available for Nepali. Also, Nepali is a subject-object-verb (SOV) language while English is subject-verb-object (SVO). This grammatical difference intuitively encourages more code-switching points since it is proven that, when code-switching occurs, the languages involved still preserve their grammatical structure (Solorio and Liu, 2008), which forces more fine-grained alternations to obey the SOV and SVO structures. In practice, we see a large code-switching rate for Nepali-English captured by the averaged CMI index in Table 2, being one of the largest scores while having a corpus of middle size.

- **Modern Standard Arabic-Egyptian Arabic (MSA-EA).** Arabic is well known for its diglossia (Ferguson, 1959); which combines a number of Arabic dialects with Modern Standard Arabic within the same community. This combination of dialects enables a large occurrence of linguistic code-switching. One of the main challenges with this language pair is that there is a significantly large word overlap while the word meanings can vary depending on the language. Even more, Arabic is a morphologically rich language and it allows multiple word orders, which increases the semantic complexity for NLP systems. Additionally similar to Spanglish, code-switching can occur at the morpheme level, where speakers often add morphological inflections to nouns.

4. Tasks

LinCE is built upon four tasks and four language pairs to provide a total of eleven datasets. In Sections 4.1 to 4.4, we discuss the datasets used for every task. Then, in Section 4.5, we describe and justify the modifications to nine out of the ten datasets in order to establish official splits that can be adopted for this benchmark. Lastly, in Section 4.6, we explain the evaluation criteria to rank the leaderboard in the LinCE platform.

4.1. Language Identification (LID)

Handling code-switched data requires to identify the languages involved. The task of language identification (LID) is one of the first steps that validates whether a system can handle code-switched data or not. Correctly classifying the language associated to text units (e.g., words or sub-word tokens) enables to process code-switched text in higher-level applications where general language understanding takes place. LinCE uses preexisting datasets for the language identification task. Specifically, in this version of LinCE, we focus on the language pairs Spanish-English, Hindi-English, Nepali-English, and Modern Standard Arabic-Egyptian Arabic. We briefly explain each corpus below, and for some of them, we propose new splits as explained in the stratification section (Section 4.5). Figure 1 shows the final distribution of the labels across the LID corpora used in LinCE. Also, these datasets follow the CALCS LID label scheme, which is `lang1`, `lang2`, `mixed` (partially in both languages), `ambiguous` (either one or the other language), `fw` (a language different than `lang1` and `lang2`), `ne` (named entities), `other`, and `unk` (unrecognizable words). More details about the LID label scheme are in Appendix A.1.

- **SPA-ENG.** We use the Spanish-English corpus from the 2016 CALCS workshop (Molina et al., 2016). This corpus uses Twitter data and it contains 32,651 posts that are comprised of 390,953 tokens. We provide new splits for this corpus because the original splits do not
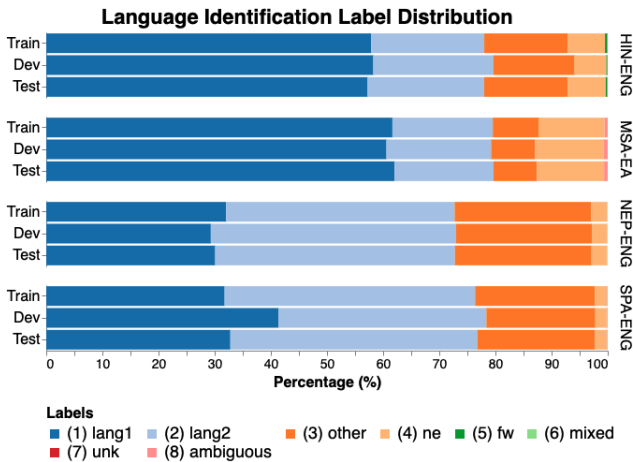
Figure 1: LID label distribution used in LinCE. While HIN-ENG and SPA-ENG have very few tokens for `unk` and `fw` (<1%), MSA-EA and NEP-ENG do not have occurrences of such labels. Also, with the exception of MSA-EA, all the partitions are proposed for LinCE as described in Section 4.5. The label scheme is described in Appendix A.1.

have a similar label distribution and the label `fw` does not appear in the development set.

- **HIN-ENG**. We use the Hindi-English corpus released by Mave et al. (2018). This corpus uses Twitter and Facebook data, which have been partly collected and partly re-used from the ICON 2016 competition (Sequiera et al., 2015a). The corpus contains a total 7,421 posts comprised of 146,722 tokens. Also, we proceed with the stratification process on this corpus because the length of the posts were not considered while doing the splits; Twitter has a character length limit in its post, whereas Facebook posts do not have such restriction resulting in significantly longer text. Moreover, the labels `ambiguous` and `unk` do not appear in the development set.

- **NEP-ENG**. The Nepali-English corpus comes from the 2014 CALCS workshop (Solorio et al., 2014). This corpus was collected from Twitter and it contains 13,011 posts and 188,784 tokens. We perform a stratification process to provide standard splits for this corpus since the organizers only provided train and test, and the test set does not include any occurrence of the `ambiguous` class.

- **MSA-EA**. We use the Modern Standard Arabic-Egyptian Arabic corpus from the 2016 CALCS workshop (Molina et al., 2016). This corpus contains Twitter data and it is comprised of 11,243 tweets with 227,354 tokens. Note that there is no occurrence of the labels `fw` and `unk` in the entire corpus. We propose new partitions due to the variation across distributions for both the LID labels as well as sentence lengths.

4.2. Parts-of-Speech (POS) Tagging

Part-of-speech (POS) tagging is an important linguistic component that enables more sophisticated syntactic anal-
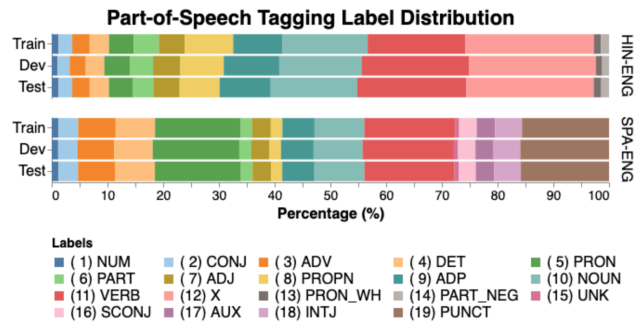

Figure 2: POS label distribution used in LinCE. The partitions for both datasets are proposed for LinCE as described in Section 4.5. Note that the labels UNK, SCONJ, AUX, INTJ, and PUNCT only appear in the SPA-ENG corpus, whereas PRON_WH is unique for HIN-ENG. The labels are described in Appendix A.2

ysis such as constituency and dependency parsing. Code-switched data is not exempted of such analysis. In fact, previous studies have shown that syntax is preserved and compliant with the syntactic rules of the individual languages when code-switching occurs (Solorio and Liu, 2008). In this benchmark, we consider the language pairs Hindi-English and Spanish-English:

- **HIN-ENG**. Singh et al. (2018b) provides 1,489 tweets (33,010 tokens) annotated with POS tags and three language IDs (`hi` for Hindi, `en` for English, and `rest` for any other token). The POS tags are annotated using the universal POS tagset proposed by Petrov et al. (2012) with the addition of two labels: PART_NEG and PRON_WH. The corpus does not provide training, development, and test splits due to the small number of samples. However, for the purposes of the benchmark, we propose standard splits using the stratification criteria discussed in Section 4.5.

- **SPA-ENG**. We use the Miami Bangor corpus with the annotations provided by Soto and Hirschberg (2017). The Bangor corpus is composed of bilingual conversations from four speakers with a total of 42,911 utterances and 333,069 tokens. The corpus contains POS tags from the universal POS tagset and LID labels. The LID labels are `eng` for English, `spa` for Spanish, `eng&spa` for mixed or ambiguous words, and UNK for everything else. Additionally, we proceed with the stratification process to provide the official training, development, and testing sets for this benchmark since the original sets were split by speakers.

4.3. Named Entity Recognition (NER)

Named entity recognition (NER) is another important core NLP task that enables higher-level applications such as question-answering, semantic role labeling, and information extraction. LinCE covers NER for three languages pairs: Spanish-English, Modern Standard Arabic-Egyptian Arabic, and Hindi-English.
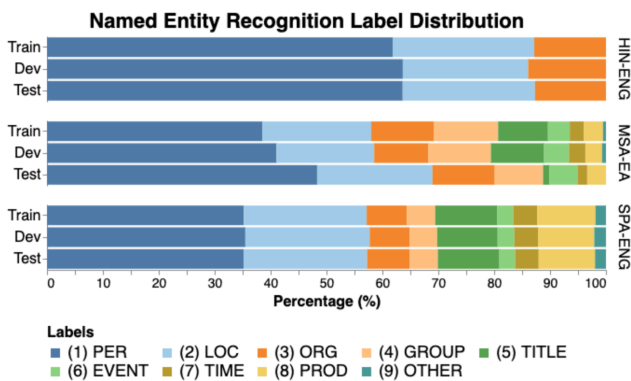
Figure 3: NER label distribution used in LinCE. All the datasets have the BIO scheme, but we only show the entity types for simplicity. Note that HIN-ENG only contains PER, LOC, and ORG. Also, with the exception of MSA-EA, all the other partitions are proposed for LinCE as described in Section 4.5. The labels are described in Appendix A.3.

- **SPA-ENG**. This corpus was introduced in the 2018 CALCS competition for NER (Aguilar et al., 2018), and it contains a total of 67,223 tweets with 808,663 tokens. The labels are organization, person, location, group, product, title, event, time, and other. Along with the NER labels, we have added the LID categories for every token, which follows the CALCS LID scheme. Moreover, we propose new splits for this corpus since the distribution of the NER labels across the splits is not consistent to the one from the full corpus. Additionally, the original development set is significantly small compared to the other splits, only accounting for 832 tweets, and the LID labels were not taken into consideration for the splitting process (e.g., the label fw does not appear in the development set). We provide new splits following the stratified process described in Section 4.5

- **MSA-EA**. This corpus was also introduced in the 2018 CALCS competition for NER, following the same entity label scheme as in the SPA-ENG corpus. The corpus uses the tweets from the 2016 CALCS LID dataset to form the training and development sets. While the LID labels are available for the training and development splits, the test set was annotated only using the NER labels. Thus, this is the only corpus for which we do not consider the language identification analysis. The corpus contains 12,335 posts and 248,452 tokens. We adopt the splits provided by the organizers during the 2018 CALCS competition.

- **HIN-ENG**. This corpus is proposed by Singh et al. (2018a), and it is composed of 2,079 tweets with 35,374 tokens. The dataset has been annotated with both NER and LID labels. The entity labels are person, location, and organization, while the LID labels are eng (English), hin (Hindi), and rest (any other token). This dataset is small, and for that reason, the authors opted to do 5-fold cross validation instead of partitioning the dataset. Nevertheless,
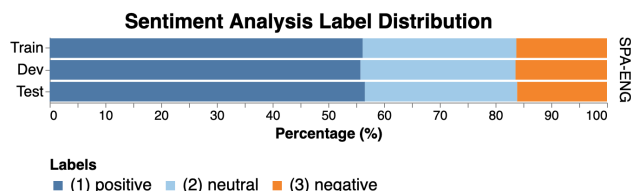

Figure 4: Label distribution of the sentiment analysis corpus used in LinCE. Note that this distribution differs from the origial dataset.

for the sake of the benchmark, we split the data using our stratification process that fairly splits the dataset accounting for LID and NER label distributions, as well as the distribution of the tweet lengths.

4.4. Sentiment Analysis (SA)

We choose sentiment analysis as our fourth benchmark task to incorporate a high-level NLP application in contrast to the previous core NLP tasks. We use the Spanish-English corpus provided in the SentiMix competition (Patwa et al., 2020). The organizers reduce the monolingual posts, increasing the number code-switched instances. Table 2 shows that this language pair is the second highest scores on the "All CMI" column for the sentiment analysis task. The task requires to predict one of the sentiments positive, negative, or neutral for every post. Additionally, this corpus is annotated with LID labels at the token level, following the CALCS LID scheme, and it contains 18,789 tweets comprised of 286,810 tokens. We propose new partitions for this dataset to correct the label distribution from the original splits.

4.5. Stratification

For nine out of ten datasets,[3] we propose new splits that in our view lead to a more appropriate evaluation (see Table 3 for a high-level distribution). We provide new splits for datasets where we found at least one of the following issues:

1. At least one of the splits does not have one or more classes. That is, one or more classes from are not evaluated at all in the development or test set.

2. The distribution of the label set for a given task is substantially different across splits or against the full corpus distribution (i.e., when merging all the splits into a single set).

3. The length of the sentences do not follow a similar distribution across splits or against the full corpus. This is a relevant criteria to consider since length is positively correlated with context, and less or more context can make a huge different for tasks such as NER.

---

[3]We did not partition the NER MSA-EA dataset because it does not have LID labels, which is essential to keep the code-switching behavior balanced accross splits in our stratification process.

Tasks	Corpus Authors	Languages	Training			Development			Test		
			CMI	Posts	Tokens	CMI	Posts	Tokens	CMI	Posts	Tokens
LID	Molina et al. (2016)	SPA-ENG	8.491	21,030	253,221	7.062	3,332	40,391	8.264	8,289	97,341
	Solorio et al. (2014)	NEP-ENG	20.322	8,451	122,952	17.079	1,332	19,273	19.754	3,228	46,559
	Mave et al. (2018)	HIN-ENG	10.222	4,823	95,224	10.122	744	15,446	9.930	1,854	36,052
	Molina et al. (2016)	MSA-EA	2.567	8,464	171,872	3.185	1,116	21,978	3.849	1,663	33,504
POS	Singh et al. (2018b)	HIN-ENG	21.449	1,030	22,993	15.293	160	3,476	18.910	299	6,541
	Soto and Hirschberg (2017)	SPA-ENG	24.191	27,893	217,068	24.040	4,298	33,345	24.282	10,720	82,656
NER	Aguilar et al. (2018)	SPA-ENG	5.567	33,611	404,428	4.398	10,085	122,656	5.867	23,527	281,579
	Singh et al. (2018a)	HIN-ENG	20.117	1,243	21,065	19.913	314	5,364	19.733	522	8,945
	Aguilar et al. (2018)	MSA-EA	–	10,103	204,296	–	1,122	22,742	–	1,110	21,414
SA	Patwa et al. (2020)	SPA-ENG	20.643	12,194	186,602	21.553	1,859	28,202	20.528	4,736	72,006

Table 3: Final data distribution of the LinCE benchmark. Note that the proposed distribution follows the stratification process described in Section 4.5, which generates partitions that differ from the original datasets.

Task	Dataset	Reason	KL-divergence	
			before	after
LID	SPA-ENG	1, 2	0.10586	0.00528
	HIN-ENG	1, 2, 3	4.64265	0.00064
	NEP-ENG	1, 3, 5	0.00552	0.00059
	MSA-EA		0.17737	0.00026
POS	SPA-ENG	4, 5	0.00140	0.00005
	HIN-ENG	5	–	0.00133
NER	SPA-ENG	1, 2, 4	0.00239	0.00001
	HIN-ENG	5	–	0.00007
SA	SPA-ENG	2, 4	0.09579	0.00002

Table 4: The table shows the datasets for which we propose new splits. The column *Reason* provides the reason number according to the aspects listed in Section 4.5. For the KL-divergence columns, we provide the average scores for the original (*before*) and the proposed (*after*) splits. The lower the score, the more similar the splits are to the full corpus distribution.

4. The NER, POS, and SA datasets contain LID labels that were not considered during the time of the stratification process, potentially affecting the balance of code-switching occurrences across the splits.

5. There is no official split for the training, development and test sets to provide the scope of fair comparison.

For the datasets where we find at least one of these issues (see Table 4), we proceed to stratify based on the language identification labels, if available, the task-specific labels (i.e., for tasks other than LID), and the lengths of the sentences. Note that providing splits that consider these three factors jointly in the stratification process is not trivial. In fact, in the case of sequence labeling tasks, we may have multiple non-unique labels per sentence, which constraints the ability to draw a distribution similar to the full corpora (e.g., adding a sentence impacts the distribution of different labels occurring in the same sentence).

To provide splits considering these three criteria, we follow the iterative stratification process proposed by Sechidis et al. (2011). This process targets multi-label data, which is a different scenario for the document and sequence labeling classification datasets used in LinCE. To adapt the sequence labels to the multi-label scenario, we treat a post as a single sample that is associated to a group of labels. In the case of tasks other than LID, we gather the LID labels with the task-specific labels (i.e., NER, POS, or SA labels) into a single group of unique labels. We also incorporate sentence lengths to the label set of a sample by choosing one of three length categories: small ($\leq$ 10 tokens), medium (>10 and $\leq$ 20 tokens), or large (>20 tokens). For instance, the SPA-ENG NER sample

$$\text{``LREC}_{ne}^{event} \; será_{lang2} \; hosted \; in \; Marseille_{ne}^{location}\text{''}$$
English: *"LREC will be hosted in Marseille"*

has the set of unique labels {lang1, lang2, ne, event, location, O, small}, where the first three labels are for LID, the following three labels are for NER, and the last one represents the sentence length (note that the order and the repetitions of the labels do not matter). Once we have the set of labels associated to a single sample (e.g., a set of LID, POS, and length labels), we can follow the iterative stratification processed used for multi-label classification on our corpus. We have found that this procedure works well in practice; we measure the KL-divergence of the label distributions from each of the splits against the distribution of the full corpus before and after the stratification, and we found that the proposed splits have less divergence (see Table 4). While KL-divergence is not often employed to corroborate the distributions of a stratified corpus, we use the divergence score to quantify whether the distribution of the full corpus has been preserved in the proposed splits, and whether the new splits are better distributions than the original splits. The final numbers of sentences and tokens per partition are listed in Table 3.

4.6. Evaluation

LinCE adopts an evaluation model similar to SemEval, Kaggle, and GLUE (Wang et al., 2018a). A platform will be hosted at `ritual.uh.edu/lince` where participants will be able to upload their predictions for the test data on each task. The platform will score the submissions and publish the results in a public leaderboard for each task. The leaderboard is ranked by the average of the task scores.

Model	Avg	LID (Accuracy)				POS (Accuracy)		NER (Micro F1)			SA (Accuracy)
		SPA-ENG	HIN-ENG	NEP-ENG	MSA-EA	SPA-ENG	HIN-ENG	SPA-ENG	HIN-ENG	MSA-EA	SPA-ENG
BiLSTM	73.20	94.16	92.34	93.29	74.26	94.80	81.84	44.92	48.36	62.64	45.39
ELMo	78.64	98.12	96.21	96.19	80.54	96.30	88.42	53.80	65.83	61.00	49.97
ML-BERT	82.93	98.53	96.44	96.57	84.14	97.00	89.28	63.56	75.96	67.61	60.20

Table 5: Baseline results on the test set of the LinCE benchmark.

5. Baseline Models

We propose LinCE to motivate a single model architecture that has good generalization capability across all the proposed tasks. To this end, we experiment with the following model architectures that demonstrated superior performance in a wide range of NLP tasks in recent years.

- **Bidirectional LSTM**. Our simplest baseline is based on LSTM cells (Hochreiter and Schmidhuber, 1997), which we operate towards both directions of the texts. For the sequence labeling tasks (i.e., LID, POS, and NER), we concatenate the forward and backward hidden representations for each of the input token and use a linear layer to predict the most probable label for each token. For sentiment analysis, we perform a max-pooling operation over all the hidden representations of the tokens and use that to predict the most probable sentiment class. For all of the tasks, we represent each token by a randomly initialized word embedding vector that is tuned during the training process.

- **ELMo**. Combining character-level convolution and word-level sequence modeling with LSTM, ELMo (Peters et al., 2018) have shown improvement in various NLP tasks acting as a pre-trained language model. We fine-tune the publicly available pre-trained ELMo models on the proposed tasks by using its high-level word representations to perform sequence labeling (i.e., LID, POS, and NER). We use the mean of the token representations to predict sentiment labels for the SA task.

- **ML-BERT** Transformer (Vaswani et al., 2017) based pre-trained language model like BERT (Devlin et al., 2018) have shown impressive generalized performance in a wide range of natural language understanding tasks. The strength of such model comes from the large amount of parameters tuned on a huge amount of training data from diverse domains. We develop our third baseline system with the pre-trained BERT model trained on multilingual data from 104 languages. We add a task-specific prediction layer over BERT and fine-tune the whole model.

5.1. Implementation and Training

We implement the models using PyTorch[4] deep learning library. We also use AllenNLP (Gardner et al., 2018) and HuggingFace's Transformers (Wolf et al., 2019) to fine-tune pre-trained ELMo and BERT based models, respectively. While evaluating a single model architecture across

[4]pytorch.org

different tasks, we keep the whole model architecture exactly the same except the prediction layer. We train the BiLSTM and ELMo models using SGD ($\eta = 0.1$ for BiLSTM and $\eta = 0.01$ for ELMo, $\beta = 0.9$). For fine-tuning BERT, we use the AdamW (Loshchilov and Hutter, 2019) ($\eta = 5e^{-5}, \epsilon = 1e^{-8}$). We use a batch size of 32 for training every model. We train each model for a maximum of 50 epochs and stop if the validation performance does not improve for 10 epochs. All of our experimental choices were tuned by observing the performance on the validation sets.

6. Results and Analysis

We report our results on the test sets for the LinCE tasks in Table 5 and discuss the details here. Across almost all the tasks we observe superior performance of the pre-trained language models compared to the simple BiLSTM model. Among the pre-trained language models, ML-BERT demonstrates superior performance in each task for all the available language pairs. ELMo's performance is very close to ML-BERT in most of the LID and POS tasks ($\approx$1-4%), but the performance gap is bigger for NER ($\approx$8-10%) and SA ($\approx$18%). The average performance gap between ELMo and BERT is $\approx$6%. We suspect that such improvement for ML-BERT against ELMo is powered by its larger parameter set (110M vs 13.6M) and the amount of data used for pre-training (Book Corpus (Zhu et al., 2015) and Wikipedia for BERT and Billion Words Benchmark (Chelba et al., 2013) for ELMo). It is also noteworthy to mention that the training data for BERT is document-level data, whereas the Billion Words Benchmark is a sentence-level corpus. Among the four tasks, NER and SA seem harder compared to LID and POS. It shows that the involvement of semantic understanding in code-switched texts makes tasks harder compared to syntactic analysis.

We observe that the LID task is harder for MSA-EA compared to the other language pairs. A possible reason for this is the large overlap between these two languages, which also affected the annotation process (Molina et al., 2016).

7. Conclusion

In this paper, we motivate the need for a centralized platform to perform evaluation of technology for code-switching data across multiple tasks and language pairs. To this end, we introduce the **Lin**guistic **C**ode-switching **E**valuation (**LinCE**) benchmark using ten publicly available datasets. In addition, we review such datasets and found important issues that undermine the evaluation process (e.g., labels not appearing in the test set, or substantially different distributions among splits, etc.). Then, we propose new splits using a new stratification technique with up to three criteria (e.g., LID labels, task-specific labels, and sentence lengths). We show the distribution of the full

corpus is preserved in the proposed splits used in LinCE, which is not always the case in the original partitions. Additionally, we provide results with strong baselines using state-of-the-art models on monolingual datasets, including BERT and ELMo. Finally, we expect that LinCE will be well-received by the NLP community, and we will keep the platform evolving with the incorporation of more tasks and language pairs in the near future.

8. Acknowledgements

9. Bibliographical References

Aguilar, G. and Solorio, T. (2019). From English to Code-Switching: Transfer Learning with Strong Morphological Clues. *arXiv preprint arXiv:1909.05158*.

Aguilar, G., AlGhamdi, F., Soto, V., Diab, M., Hirschberg, J., and Solorio, T. (2018). Named Entity Recognition on Code-Switched Data: Overview of the CALCS 2018 Shared Task. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, pages 138–147, Melbourne, Australia, July. Association for Computational Linguistics.

Al-Badrashiny, M. and Diab, M. (2016). LILI: A Simple Language Independent Approach for Language Identification. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1211–1219, Osaka, Japan, December. The COLING 2016 Organizing Committee.

AlGhamdi, F., Molina, G., Diab, M., Solorio, T., Hawwari, A., Soto, V., and Hirschberg, J. (2019). Part-of-Speech Tagging for Code-Switched Data. *arXiv preprint arXiv:1909.13006*.

Bali, K., Sharma, J., Choudhury, M., and Vyas, Y. (2014). "I am borrowing ya mixing ?" An Analysis of English-Hindi Code Mixing in Facebook. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 116–126, Doha, Qatar, October. Association for Computational Linguistics.

Chandu, K., Loginova, E., Gupta, V., Genabith, J. v., Neumann, G., Chinnakotla, M., Nyberg, E., and Black, A. W. (2018). Code-Mixed Question Answering Challenge: Crowd-sourcing Data and Techniques. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, pages 29–38, Melbourne, Australia, July. Association for Computational Linguistics.

Chelba, C., Mikolov, T., Schuster, M., Ge, Q., Brants, T., Koehn, P., and Robinson, T. (2013). One Billion Word Benchmark for Measuring Progress in Statistical Language Modeling. Technical report, Google.

Choudhury, M., Chittaranjan, G., Gupta, P., and Das, A. (2014). Overview of FIRE 2014 Track on Transliterated Search. *Proceedings of FIRE*, pages 68–89.

Das, A. (2016). Tool contest on POS tagging for code-mixed Indian social media (Facebook, Twitter, and Whatsapp) text. retrieved 05-10-2019.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*.

Ferguson, C. A. (1959). Diglossia. *word*, 15(2):325–340.

Gambäck, B. and Das, A. (2014). On Measuring the Complexity of Code-mixing. In *Proceedings of the 11th International Conference on Natural Language Processing, Goa, India*, pages 1–7. Citeseer.

Gardner, M., Grus, J., Neumann, M., Tafjord, O., Dasigi, P., Liu, N., Peters, M., Schmitz, M., and Zettlemoyer, L. (2018). AllenNLP: A Deep Semantic Natural Language Processing Platform. *arXiv preprint arXiv:1803.07640*.

Geetha, P., Chandu, K., and Black, A. W. (2018). Tackling Code-Switched NER: Participation of CMU. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, pages 126–131, Melbourne, Australia, July. Association for Computational Linguistics.

Hamed, I., Elmahdy, M., and Abdennadher, S. (2018). Collection and Analysis of Code-switch Egyptian Arabic-English Speech Corpus. In Nicoletta Calzolari (Conference chair), et al., editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 7-12, 2018. European Language Resources Association (ELRA).

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural computation*, 9(8):1735–1780.

Jamatia, A., Gambäck, B., and Das, A. (2015). Part-of-speech tagging for code-mixed English-Hindi twitter and Facebook chat messages. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 239–248, Hissar, Bulgaria, September. INCOMA Ltd. Shoumen, BULGARIA.

Khandelwal, A., Swami, S., Akhtar, S. S., and Shrivastava, M. (2018). Humor detection in English-Hindi code-mixed social media content : Corpus and baseline system. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May. European Language Resources Association (ELRA).

Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Mager, M., Çetinoğlu, Ö., and Kann, K. (2019). Subword-level language identification for intra-word code-switching. *Proceedings of the 2019 Conference of the North*.

Mave, D., Maharjan, S., and Solorio, T. (2018). Language identification and analysis of code-switched social media text. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, pages 51–61, Melbourne, Australia, July. Association for Computational Linguistics.

Molina, G., AlGhamdi, F., Ghoneim, M., Hawwari, A.,

Rey-Villamizar, N., Diab, M., and Solorio, T. (2016). Overview for the Second Shared Task on Language Identification in Code-Switched Data. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 40–49, Austin, Texas, November. Association for Computational Linguistics.

Partanen, N., Lim, K., Rießler, M., and Poibeau, T. (2018). Dependency parsing of code-switching data with cross-lingual feature representations. In *Proceedings of the Fourth International Workshop on Computational Linguistics of Uralic Languages*, pages 1–17, Helsinki, Finland, January. Association for Computational Linguistics.

Patwa, P., Aguilar, G., Kar, S., Pandey, S., PYKL, S., Garrette, D., Gambäck, B., Chakraborty, T., Solorio, T., and Das, A. (2020). SemEval-2020 Sentimix Task 9: Overview of SENTIment Analysis of Code-MIXed Tweets. In *Proceedings of the 14th International Workshop on Semantic Evaluation (SemEval-2020)*, Barcelona, Spain, September. Association for Computational Linguistics.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June. Association for Computational Linguistics.

Petrov, S., Das, D., and McDonald, R. (2012). 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. European Language Resources Association (ELRA).

Pratapa, A., Choudhury, M., and Sitaram, S. (2018). Word embeddings for code-mixed language processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3067–3072, Brussels, Belgium, October-November. Association for Computational Linguistics.

Raghavi, K. C., Chinnakotla, M. K., and Shrivastava, M. (2015). Answer ka type kya he?: Learning to classify questions in code-mixed language. In *Proceedings of the 24th International Conference on World Wide Web*, pages 853–858. ACM.

Rothman, J. and Rell, A. B. (2007). A linguistic analysis of spanglish: Relating language to identity. *Linguistics and the Human Sciences*, 1(3):515–536.

Roy, R. S., Choudhury, M., Majumder, P., and Agarwal, K. (2013). Overview of the FIRE 2013 Track on Transliterated Search. In *Post-Proceedings of the 4th and 5th Workshops of the Forum for Information Retrieval Evaluation*, page 4. ACM.

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

Sechidis, K., Tsoumakas, G., and Vlahavas, I. (2011). On the stratification of Multi-label Data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 145–158. Springer.

Sequiera, R., Choudhury, M., and Bali, K. (2015a). POS Tagging of Hindi-English Code Mixed Text from Social Media: Some Machine Learning Experiments. In *2015 Proceedings of International Conference on NLP*. NLPAI, December.

Sequiera, R., Choudhury, M., Gupta, P., Rosso, P., Kumar, S., Banerjee, S., Naskar, S. K., Bandyopadhyay, S., Chittaranjan, G., Das, A., et al. (2015b). Overview of FIRE-2015 Shared Task on Mixed Script Information Retrieval. In *FIRE Workshops*, volume 1587, pages 19–25.

Singh, K., Sen, I., and Kumaraguru, P. (2018a). Language identification and named entity recognition in Hinglish code mixed tweets. In *Proceedings of ACL 2018, Student Research Workshop*, pages 52–58, Melbourne, Australia, July. Association for Computational Linguistics.

Singh, K., Sen, I., and Kumaraguru, P. (2018b). A twitter corpus for Hindi-English code mixed POS tagging. In *Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media*, pages 12–17, Melbourne, Australia, July. Association for Computational Linguistics.

Sitaram, S., Chandu, K. R., Rallabandi, S. K., and Black, A. W. (2019). A survey of code-switched speech and language processing. *arXiv preprint arXiv:1904.00784*.

Solorio, T. and Liu, Y. (2008). Part-of-speech tagging for english-spanish code-switched text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 1051–1060, Stroudsburg, PA, USA. Association for Computational Linguistics.

Solorio, T., Blair, E., Maharjan, S., Bethard, S., Diab, M., Ghoneim, M., Hawwari, A., AlGhamdi, F., Hirschberg, J., Chang, A., and Fung, P. (2014). Overview for the first shared task on language identification in code-switched data. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 62–72, Doha, Qatar, October. Association for Computational Linguistics.

Soto, V. and Hirschberg, J. (2017). Crowdsourcing universal part-of-speech tags for code-switching. *Interspeech 2017*, Aug.

Soto, V. and Hirschberg, J. (2018). Joint part-of-speech and language ID tagging for code-switched data. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, pages 1–10, Melbourne, Australia, July. Association for Computational Linguistics.

Tristram, H. L. C. (1999). *How Celtic is Standard English?* Nauka, St Petersburg, Russia.

Trivedi, S., Rangwani, H., and Kumar Singh, A. (2018). IIT (BHU) submission for the ACL shared task on named entity recognition on code-switched data. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, pages 148–153,

Melbourne, Australia, July. Association for Computational Linguistics.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In I. Guyon, et al., editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Vilares, D., Alonso, M. A., and Gómez-Rodríguez, C. (2015). 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*, pages 2–8. Association for Computational Linguistics.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. (2018a). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium, November. Association for Computational Linguistics.

Wang, C., Cho, K., and Kiela, D. (2018b). Code-Switched Named Entity Recognition with Embedding Attention. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, pages 154–158, Melbourne, Australia, July. Association for Computational Linguistics.

Winata, G. I., Wu, C.-S., Madotto, A., and Fung, P. (2018). Bilingual character representation for efficiently addressing out-of-vocabulary words in code-switching named entity recognition. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, pages 110–114, Melbourne, Australia, July. Association for Computational Linguistics.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., and Brew, J. (2019). Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *The IEEE International Conference on Computer Vision (ICCV)*, December.

Appendix for "LinCE: A Centralized Benchmark for Linguistic Code-switching Evaluation"

A. Label Schemes

A.1. LID Label Scheme

We use the CALCS label scheme for language identification, which contains the following labels:

- `lang1` represents one of the code-switched languages. For English-paired lanaguages, this label is used for English tokens; for MSA-EA, this label is used for Modern Standard Arabic.

- `lang2` represents the other code-switched language. For LinCE, this label could be Spanish, Hindi, Nepali, or Egyptian Arabic, depending on the dataset.

- `mixed` represents a word that is partly in `lang1` and partly in `lang2`.

- `ambiguous` is used for words whether it is unclear if they belong to one or the other language.

- `fw` means foreign word, and it is used for words that are in a language different than `lang1` or `lang2`.

- `ne` represents the named entity tokens.

- `unk` means unknown, and it is used for tokens whose language is not recognized.

- `other` captures symbols, punctuation, and emoticons.

A.2. POS Label Scheme

We use the universal part-of-speech (UPOS) tagset proposed by Petrov et al. (2012) with the addition of `PRON_WH` proposed by Singh et al. (2018b) and `UNK` proposed by Soto and Hirschberg (2017).

- `ADJ` is used for adjectives.

- `ADP` is used for prepositions and postpositions.

- `ADV` is used for adverbs.

- `AUX` is used for auxiliaries.

- `CONJ` is used for coordinating conjunctions. This is represented by 'CCONJ' in the univeral POS tagset.

- `DET` is used for determiners and articles.

- `INTJ` is used for interjections.

- `NOUN` is used for nouns.

- `NUM` is used for numerals.

- `PART` is used for particles.

- `PRON` is used for pronouns.

- `PROPN` is used for proper nouns.

- `PUNCT` is used for punctuation marks.

- `SCONJ` is used for subordinating conjunctions.

- `VERB` is used for verbs.

- `X` for all other categories such as abbreviations or foreign words.

- `PRON_WH` is used for interrogative pronouns (like where, why, etc.). This extension is employed by Singh et al. (2018b).

- `UNK` is used when it is not possible to determine the syntactic category. This extension is employed by Soto and Hirschberg (2017).

A.3. NER Label Scheme

We use the CALCS label scheme for named entity recognition (Aguilar et al., 2018). These labels use the BIO scheme and contain the following entity types:

- `person` for proper nouns or nicknames.

- `organization` for institutions, companies, organizations, or corporations. Not to confuse with products when they have the same name as an organization.

- `location` for physical places that people can visit and that have a unique name. Addresses, facilities, and touristic places are examples of this.

- `group` for sports teams, music bands, duets, etc. Not to confuse with organization.

- `product` for articles that have been manufactured or refine for sale, like devices, medicine, food, well-defined services.

- `title` for title of movies, books, TV shows, songs, etc. Titles can be sentences and they usually refer to media, which can be considered a fine-grained version of product.

- `event` for situations or scenarios that gather people for a specific purpose such as concerts, competitions, conferences, award events, etc. Events do not consider holidays.

- `time` for months, days of the week, seasons, holidays and dates that happen periodically, which are not events (e.g., Christmas). It excludes hours, minutes, and seconds.

- `other` for any other named entity that does not fit in the previous categories.