Speaker Information Can Guide Models to Better Inductive Biases: A Case Study On Predicting Code-Switching

Anonymous ACL submission

Abstract

Natural language processing (NLP) models trained on people-generated data can be unreliable because, without any constraints, they can learn from spurious correlations or propagate dangerous biases about personal identities. We hypothesize that enriching models with speaker information in a controlled, educated way can guide them to pick up on relevant inductive biases. For the speaker-driven task of predicting code-switching points in English–Spanish bilingual dialogues, we show that adding sociolinguistically-grounded speaker features as prepended prompts significantly helps to improve accuracy. We find that by adding influential phrases to the input, speaker-informed models learn useful and explainable linguistic information. To our knowledge, we are the first to incorporate speaker characteristics in the code-switching setup, and more generally, take a step towards developing transparent models that control for biases in person-centric tasks.

1 Introduction

Imbalanced datasets, flawed annotation schemes, and even model architectures themselves can all cause neural models to encode and propagate biases (Sun et al., 2019; Field et al., 2021). These biases can be social, linguistic, or a mix of both, resulting in models that are brittle and offensive in the presence of racial or gender attributes (Kiritchenko and Mohammad, 2018; Nozza et al., 2021), unsuitable for processing mixed-language text or dialect variations (Sap et al., 2019; Kumar et al., 2021; Winata et al., 2021), or ones that can miscommunicate intents in translation setups. Contextualizing models in social factors is important for preventing these issues and building more socially intelligent and culturally sensitive NLP technologies (Hovy and Yang, 2021).

We hypothesize that grounding models in speaker information can help them learn more useful inductive biases, thereby improving performance on person-oriented classification tasks. We test this hypothesis on the task of code-switch (language change) prediction in a multilingual dialogue, which is inherently linguistically *and* socially driven (Li, 2013). Prior approaches to predicting code-switching consider only shallow linguistic context (Doğruöz et al., 2021). As we show in our experiments, using a standard Transformer-based classifier (Conneau et al., 2020) trained with only linguistic context results in sub-optimal and unstable models.

We then ground the models in relevant social factors, such as age, native language, and language-mixing preference of the interlocutors, via text-based speaker descriptions or *prompts* (cf. Zhong et al., 2021; Wei et al., 2021). We find that prepending speaker prompts to dialogue contexts improves performance significantly, and leads to more stable generalizations. Our prompts are different from the embedding-based "personas" of Li et al. (2016) and the synthesized descriptions from Persona-Chat (Zhang et al., 2018), capturing fine-grained theoretically grounded social and linguistic properties of speakers, as opposed to simple likes or dislikes.

To analyze the inductive biases that the models learn, we use *SelfExplain* (Rajagopal et al., 2021)—an interpretable text classification model highlighting key phrases in the input text. We propose a new method for aggregating the interpretations produced by SelfExplain, which helps us to explain model predictions and align them with sociolinguistic literature.

We motivate our study of predicting code-switching in §2, and describe the task and interpretable neural text classification models in §3. After outlining important ethical considerations in §4, we detail our experiments (§5) and results (§6), and provide an analysis of speaker-aware model generalizations that are grounded in prior psycholinguistic research on code-switching (§7).

2 Motivation

Our overarching goal is to develop a general and theoretically-informed methodology to ground neural models in a social context, because a wide array of person-centric classification tasks, such as sentiment prediction or hate speech detection, can fail without proper social contextualization (Sap et al., 2019; Kiritchenko and Mohammad, 2018; Hovy and Yang, 2021). We choose a speaker-driven task that is ethically safer to experiment with (see a detailed discussion in §4): predicting code-switching in human–human dialogues.

Code-switching is the alternation between languages within and between utterances.[1] It is a language- and speaker-driven phenomenon, reflecting speaker identities and relationships between them, in addition to their linguistic backgrounds, preferences and topical constraints (Beatty-Martínez et al., 2020). Prior sociolinguistic work established the importance of speaker context for code-switching, and existing multilingual models—trained with only monolingual linguistic context—are not speaker-grounded nor well-suited for dealing with mixed-language data, leaving gaps which we begin to address.

Figure 1 provides a key motivating example of how global speaker features of two bilingual conversational participants influence their local speech production. *Blue*, whose native language is Spanish, begins speaking in Spanish, while *Green* responds in English. Following *Green*'s clarification question about the actor *The Rock*, *Green* begins in English, but will accommodate *Blue* (Ahn et al., 2020; Beatty-Martínez et al., 2020) to reply with *el actor* (Spanish), motivating the need for social context when processing mixed-language data.

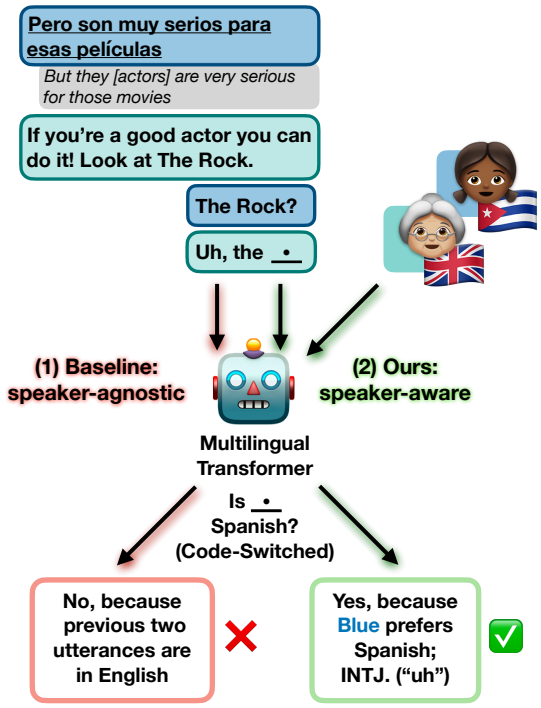

Figure 1: We use a Transformer-based model to predict language switches in dialogues and identify phrase-level features guiding predictions. Here, both speakers are bilingual, but Blue's native language is Spanish and Green's native language is English. They have unique social factors (such as age). The dialogue structure reflects speaker identities and relationships: Green will switch to Spanish with *el actor*, accommodating Blue's language preference. Using only dialogue context, the baseline (1) fails to pick up on this, while our speaker-aware model (2) successfully predicts a code-switch and identifies useful linguistic cues.

3 Methodology

In this section we introduce the task of predicting code-switching points and describe the base model for it, with a self-explainable architecture as its backbone. We then describe how we incorporate speaker-grounding prompts into the model.

3.1 Task Definition

Let $d_i = [w_1, w_2, \ldots, w_u]$ be an utterance (string of tokens) in the full dialogue $\mathcal{D}$. Given a context window of size h, a model processes a local dialogue context: $[d_{i-h}, \ldots, d_{i-1}, d'_i]$, where $d'_i := [w_1, w_2, \ldots, w_b], b \in \{1, 2, \ldots, u\}$. In other words, we take the prefix of the current utterance d_i up to an index b. Each word w_j in the dialogue has a language tag l_j associated with it. For the given dialogue context D up to boundary-word w_b, a model must predict whether the language of the next word after w_b will be *code-switched* (1), or the same (0). In our setup, a code-switch occurs between two consecutive words w_b, w_{b+1} if the language of w_b is in English and the language of w_{b+1} is in Spanish (or vice versa). In particular, a word with an ambiguous language, such as the proper noun *Maria*, cannot be a switch point; only words with unambiguous language tags are switched.

Speaker-Aware Grounding Each utterance in the dialogue context has a speaker associated with it. Let the set of all speakers in the dialogue context be $\mathcal{S} = \{s_1, s_2, s_3, \ldots, s_M\}$. We define a speaker-aware prompt $\mathcal{P} = \{p_1, p_2, p_3, \ldots, p_K\}$

[1] See Appendix A.1 for a detailed example of code-switched dialogue.

Prompt	Speaker Description Example
List	**ASH** is first speaker, older, female, from Spanish speaking country, between English and Spanish prefers both, rarely switches languages. **JAC** is second speaker, older, male, from Spanish speaking country, between English and Spanish prefers both, never switches languages.
Sentence	**ASH** is a middle-aged woman from a Spanish speaking country. Between English and Spanish she prefers both, and she rarely switches languages. **ASH** speaks first. **JAC** is a middle-aged man from a Spanish speaking country. Between English and Spanish he prefers both, and he never switches languages. **JAC** speaks second.
Partner	**ASH, JAC** are all middle-aged from a Spanish speaking country. Between English and Spanish they prefer both. **ASH** is a woman and rarely switches languages. **JAC** is a man and never switches languages. **ASH** speaks first.

Table 1: Examples of prompts for two speakers ID'd **ASH** and **JAC**, structured in the three different formats: List, Sentence, and Partner. We prepend these prompts to dialogue context D to train our speaker-grounded models. All prompts cover attribute set $\mathcal{A}$ consisting of age, gender, country of origin, language preference, code-switching preference, and speaker order in the global dialogue context. Sentence and List prompts are similar in that they describe speakers separately; Sentence prompts are more prose-like. Partner prompts first highlight similarities between speakers, capturing speaker entrainment features, before describing unique features of each speaker.

as a concatenation of K strings p_i, each describing an attribute of a speaker in the dialogue. Together, $\mathcal{P}$ describes the unique attributes of all M speakers in the dialogue context.

Our proposed speaker-guided models take as input $\mathcal{P} \cdot \mathcal{D} = [p_1, \ldots, p_K, d_{i-w}, \ldots, d_i']$, the concatenation of prompts and dialogue context. We encode the inputs with a multilingual Transformer-based architecture (Devlin et al., 2018; Conneau et al., 2020) before using a linear layer to predict the presence or absence of a code-switch.

3.2 Generating Speaker Prompts

We incorporate global information about each speaker in a dialogue using different prompt styles, generating a prompt $\mathcal{P}$ for a given dialogue context D. In theory, these prompts have the potential to change the model's priors by contextualizing dialogue with speaker information and should be more useful for predicting upcoming language switches. We consider two aspects when designing prompts.

Content The prompt describes all speakers $\mathcal{S}$ in the dialogue using a set of speaker attributes $\mathcal{A} = \{a_1, a_2, \ldots, a_T\}$. To create a description P_m for speaker $s_m \in \mathcal{S}$, we combine phrases $p_{s_{m_1}}, p_{s_{m_2}}, \ldots, p_{s_{m_T}}$, such that each phrase corresponds to exactly one attribute. As Table 1 indicates, we use speaker IDs to tie a speaker to her description, and all prompts cover the full set of attributes, $\mathcal{A}$, for all speakers in D.

Form We consider three prompt forms: *List*, *Sentence*, and *Partner*. The prompt form determines both the resulting structure of prompt string $\mathcal{P}$ and the way we combine local attribute phrases p_j to generate a speaker description P_i. Table 1 provides concrete examples of List, Sentence, and Partner prompts for a pair of speakers.

List & Sentence List and Sentence prompts do not explicitly relate speakers to each other: the final prompt $\mathcal{P} = \{P_1, \ldots, P_m, \ldots, P_M\}$ concatenates individual speaker prompts P_i. List forms combine all attributes in a speaker description P_m with commas, while Sentence forms are more prose-like.

Partner According to prior work (Bawa et al., 2020; Ahn et al., 2020; Myslín and Levy, 2015), speaker *entrainment* or accommodation influences code-switching behavior. Thus, we created Partner prompts to explicitly highlight relationships between speakers. We hypothesize these are more useful than the List and Sentence forms, from which the model must implicitly learn speaker relationships. Partner prompts include an initial P_i containing attribute qualities shared by all speakers:

$$P_i := \left\{ p_{a_j} | a_j = v_k, \forall s \in \mathcal{S} \right\},$$

where $a_j \in \mathcal{A}$ and v_k is a value taken on by attribute a_j. As an example, all speakers may prefer Spanish, so P_i will contain an attribute string p_i capturing this. The final partner prompt is $\mathcal{P}_{partner} = \{P_i, P_1, P_2, \ldots, P_M\}$, where speaker-specific descriptions $P_1, P_2, \ldots, P_M$, highlight unique values of each speaker.

We prepend prompts $\mathcal{P}$ to dialogue context $\mathcal{D}$ using [EOS] tokens for separation.

3.3 Interpretable Text Classification

Our proposed setup takes as input the dialogue context and a prepended speaker prompt. To explain predictions of the baseline and our speaker-aware setups, we use SelfExplain (Rajagopal et al., 2021), a framework for interpreting text-based deep learning classifiers using phrases from the input. SelfExplain incorporates a Locally Interpretable Layer (LIL) and a Globally Interpretable Layer (GIL). GIL retrieves the top-k relevant phrases in the training set for the given instance, while LIL ranks local phrases within the input according to their influence on the final prediction. Using a local classification layer, LIL quantifies the effects that subtracting a local phrase representation from the full sentence have on the resulting prediction. We exclusively use LIL to highlight phrases in the speaker prompts and dialogues to identify both social factors and linguistic context influential to models; through post-hoc analysis, we can reveal whether these features can be corroborated with prior literature or indicate a model's reliance on spurious confounds. Figure 4 illustrates our full proposed model with two classification heads: one for prediction and one for interpretation. §7.1 describes our method for scoring phrases according to their influence on the final prediction. These phrase scores are essential for our analysis.

4 Ethical Considerations

Data Privacy In line with prior behavioral studies, our work illustrates that sociolinguistic cues are essential for predicting code-switching points. Deployable speaker-informed models must protect the identity and privacy of users through techniques such as federated machine learning: deploying local models to end-users without sending any user information back to the cloud (Konečný et al., 2016). Local models and data should be encrypted to prevent breaches and tampering with algorithms, as well as possible reconstruction of training data (Hitaj et al., 2017; Carlini et al., 2019; Zhang et al., 2020). A deployed system should only collect and access information if the user agrees to it. All conversational participants voluntarily shared the information we use to develop our models.

Moreover, this research is important to conduct because there is good reason to believe that human users react positively to appropriately adaptive technologies (Branigan et al., 2010). Specifically, initial experiments indicate that users rate dialogue systems that incorporate code-switching higher than ones that do not (or that do it less naturally) (Ahn et al., 2020; Bawa et al., 2020). A classifier, such as the one we explore in this work, can be very useful for developing such a naturalistic dialogue system. Different regions and cultures have varying opinions of code-switching. It is important to understand these before building an application for a new language pair (Doğruöz et al., 2021).

5 Experimental Setup

5.1 Dialogue Data

Our task requires a dataset which not only has natural, mixed-language dialogue, but includes also information about its speakers. We use the Bangor Miami (Deuchar et al., 2014) dataset (BM) containing 56 transcribed dialogues in mixed English and Spanish. Moreover, language IDs are provided for every token. The dataset includes a questionnaire of self-reported information about each conversational participant; this includes macro-social features such as age, gender, and country of origin, as well as language preferences and speaker-provided linguistic ability. We identify each country according to the primary language (English, Spanish, or neither) spoken in the country and bin age features into four comparative groups ranging from youngest to oldest. An order feature indicates which speaker spoke first, second, etc. in the global dialogue context. These six features define our attribute set $\mathcal{A}$.

5.2 Code-switching Dataset Creation

For each dialogue in BM, we extract all existing code-switch points; for a given switched word, we retain all left-most context in its containing utterance and vary the number of prior utterances that are included as context between 1, 2, 3, and 5. To generate negative examples, we select monolingual utterances by sampling from a binomial distribution with $p = 0.75$. For each retained utterance, we randomly choose three potential switch points (extracting leftmost context in the same way), resulting in a dataset that is approximately 25% switched.

Creating Splits Most speakers participate in only one of the 56 dialogues in the corpus. To help ensure the model sees new dialogue context in

training and testing time, we split the train, validation, and test splits by conversation in a 60:20:20 ratio. For each dialogue, we compute the multilinguality index (M-Index) (Barnett et al., 2000), a measure between 0 and 1 indicating the mixedness in the text: 0 is monolingual text, while 1 is a code-switch at every word. We stratify the conversations by the M-Index and code-switching labels to enforce a more balanced distribution of monolingual to mixed-language conversations.

We down-sample monolingual examples to balance training and validation splits and report results on unbalanced validation and test sets. Table 2 shows the proportions of code-switched examples. Our final balanced training and validation splits have about 14,000 and 3,000 examples, while the unbalanced validation and test sets have approximately 7,000 and 9,000 examples, respectively.

Marking Dialogue Turns The baseline setup does not incorporate speaker cues. Instead we use [EOT] and [EOU] tokens at the end of each utterance to signify end-of-turn and end-of-utterance, respectively. Given two consecutive utterances, an [EOT] signifies a change in speakers, while [EOU] indicates no change. In the speaker-informed setup, unique speaker IDs distinguish utterances from each speaker, and we prepend informative *prompts* characterizing the conversational participant(s). Prompts include user-reported metadata of personal preferences and characteristics. We use three prompt templates, as detailed in Section 3.

5.3 Training Details

We use XLM-RoBERTa (XLMR) (Conneau et al., 2020) to encode the text and jointly fine-tune XLMR on the code-switch prediction task. As a baseline, we use an XLMR model without prompt inputs $\mathcal{P}$. Our speaker-prompted models, SP-XLMR, are trained by prepending speaker prompts to the dialogue context. The small size of our dataset results in higher variability in performance; thus, we train 10 models, each on a different seed, per prompt type, resulting in 30 speaker-prompted models and 10 baseline models. We refer to speaker-prompted models as SP-XLMR and to the non-speaker baseline as simply XLMR. All models are trained using AdamW optimizers with a weight decay of $1e^{-3}$ for a maximum of 10 epochs. SP-XLMR models are trained with a learning rate of $5e^{-5}$ and XLMR models use a learning rate of $1e^{-5}$. To refer to a particular speaker-prompted model, we use a combination of prompt form and context size, for example, LIST-5.

We report accuracy, F1, precision, and recall on the unbalanced validation and test sets.

Set	Validation	Test
Balanced	0.500	0.500
Unbalanced	0.250	0.252

Table 2: Proportion of code-switched examples in the balanced and unbalanced validation and test splits.

6 Evaluation

Speaker prompts significantly improve code-switch prediction. Table 3 includes average accuracy and F1 of XLMR, LIST, SENTENCE, and PARTNER models, across all context windows and seeds on the unbalanced validation and test sets. Each value is an average of 40 models. Adding prompt features boosts accuracy upwards of 5-8% and F1 by 4-5 points compared to XLMR; XLMR does not even surpass the majority baseline in accuracy. Based on validation set results, partner features are most helpful, confirming our sociolinguistically-driven hypothesis (see Section 3.2) Moreover, the standard deviation of XLMR accuracy is more than twice as large (3.66 on validation and 2.95 on test) as that of any speaker-prompted model, suggesting that explicit speaker information guides models to pick up on relevant inductive biases.

We see similar trends, regarding accuracy, F1, and standard deviation, in Table 4, which includes results for SP-XLMR and XLMR across the different context windows; each SP-XLMR and XLMR value is an average of 30 and 10 models, respectively. Larger context windows are helpful for both model types. Tables 6 and 7 include precision and recall scores for each prompt type and context window; in general, speaker-prompted models have upwards of 10 points higher precision than baseline XLMR, indicating that speaker information helps to identify valid switch points.

7 Explaining Performance Gaps

Compared to baseline models, speaker models not only attain higher accuracy and F1 scores, but they also have a much smaller standard deviation in scores. For these experiments, we seek to explain our findings using the important phrases identified by LIL. Within a speaker prompt $\mathcal{P}$, each speaker characteristic maps to its own phrase (i.e., *from an*

	Validation		Test	
Model	**Acc. (%)**	**F1**	**Acc. (%)**	**F1**
Majority	75.0	–	74.8	–
Minority	25.0	29.4	25.2	29.6
XLMR	70.3 ±3.66	57.3	72.0 ±2.95	59.1
List	77.6 ±1.68	61.5	**79.7** ±1.13	**63.2**
Sentence	78.1 ±1.60	61.8	79.5 ±1.31	63.0
Partner	**78.3** ±1.58	**62.1**	79.4 ±1.50	62.2

Table 3: Average accuracy and F1 scores of prompt models and XLM-R on validation and test sets. There are N=40 models for all setups. Majority and Minority baselines are included for comparison. **Bold** scores indicate the best performance on the split. All results are significant ($p < 0.0001$) by Mann-Whitney U Tests.

English-speaking country); in the dialogue, we extract 5-gram phrases using a sliding window. In the sections below, we detail our approach to scoring phrase influence and explain our analyses of key dialogue and speaker features.

7.1 Computing Phrase Relevance

Our goals are to (a) identify phrases in the input whose removal will change the resulting model prediction and (b) identify phrases which contribute high confidence to the resulting model prediction. Let F be the full textual input consisting of sole dialogue context or dialogue context prepended with prompts, while Z_F is the softmax output from our classifier. Let j be the index of the class predicted from the full input. LIL inputs Z_F along with a series of masks each corresponding to a local phrase in either the dialogue or the speaker prompt. Let *nt* be a local phrase, such that nt is either a speaker phrase p_i or an n-gram in an utterance $d_i \in D$. Using LIL, we quantify the effect of removing the representation of phrase *nt* from the representation of F by comparing the activation differences of Z_{nt} and Z_f at index j, and we analyze the resulting sign and magnitude to address goals (a) and (b), respectively:

$$C := \begin{cases} 1 & \text{argmax } Z_{nt} = j \\ -1 & \text{argmax } Z_{nt} \neq j \end{cases} \quad (1)$$

$$r(nt) = C\,|z_{nt_j} - z_{F_j}| \quad (2)$$

where z_{nt_j} and z_{F_j} are the softmax scores of the phrase-ablated sentence and the full sentence, respectively, at index j, and $r(nt)$ is the relevance score of nt. As Equations 1 and 2 indicate, we analyze a local phrase's score as follows:

1. **Sign** A positive sign ($C = 1$) indicates that the representation without *nt* does not change the resulting prediction. A negative score indicates a more influential phrase because its ablation results in a different prediction ($C = -1$).
2. **Magnitude** Magnitude corresponds to the weight of the contribution of a particular phrase. If the activation difference is high in magnitude, then *nt* strongly influences the resulting prediction. Magnitudes near 0 indicate a non-influential phrase.

Our scoring approach differs slightly from the original implementation (see Appendix A.2).

7.2 Analyzing Dialogue Phrases

Given a context size, the dialogue phrase masks are identical for SP-XLMR and XLMR; thus, we directly compare which phrases are most informative in the presence and absence of speaker features. We consider only phrases which are influential enough to change a given model's prediction after their representations are subtracted from the full-sentence representations (phrases with a negative score).

Setting context size to 5, we identify examples from the validation set for which the majority of SP-XLMR models (out of 30) predicted correctly and the majority of XLMR models (out of 10) predicted incorrectly. Nearly 95% of such examples are not switched, indicating that added speaker information helps improve model precision. We sample a portion of these instances for our analysis.

For a given validation set example and model setup, we track all influential phrases and count the number of models for which each phrase is influential. To account for phrase interactions, we track the agreement on co-occurring pairs and trios of important phrases. We compare only top-10 influential phrases. We hypothesize speaker models (1) exhibit more phrase agreements compared to baseline models and (2) use more helpful and relevant linguistic features for code-switch prediction.

Most speaker models agree on similar groups of phrases. Figure 2 indicates that the majority of speaker-prompted models (out of 30) tend to agree on the top-10 important phrase groupings, especially across single and pairwise groupings. The speaker models likely pick up on similar inductive biases, as revealed through the higher feature agreement among these models. Only around 38-40% of baseline models tend to agree on which phrases are

	Validation				Test			
	SP-XLMR		XLM-R		SP-XLMR		XLM-R	
Ctx	Acc. (%)	F1	Acc. (%)	F1	Acc. (%)	F1	Acc. (%)	F1
1	76.9 ±1.96	60.5	66.4 ±2.84	54.0	78.8 ±1.54	60.7	69.5 ±2.75	56.5
2	77.9 ±1.10	61.5	70.3 ±3.27	57.2	79.6 ±1.13	62.9	71.8 ±2.23	58.7
3	78.6 ±1.17	62.2	71.4 ±1.92	58.2	80.0 ±0.96	**63.6**	72.4 ±2.31	59.8
5	78.7 ±1.56	**63.1**	73.1 ±2.74	59.8	79.7 ±1.34	**63.9**	74.2 ±2.39	61.2

Table 4: Average Accuracy and F1 of prompt models and baseline XLM-R on validation and test sets, for N=30 SP-XLMR models and N=10 XLMR models. All results are significant ($p < 0.0001$) by Mann Whitney U Tests.

most important, potentially explaining the higher standard deviation in results among the baseline models compared to the speaker models.

Speaker models make better use of language information On monolingual (negative) examples, both speaker-prompted and baseline models tend to look at a majority of monolingual phrases in the same languages (English or Spanish), and these phrases are mainly located in the first quarter of tokens preceding the switch point. However, speaker models successfully predict many of these negative examples correctly, unlike baselines. In many cases, the speaker models have additional access to global speaker properties of the current speaker – for example, *never switches languages* – and this may also influence them to make the correct prediction given prior linguistic context. Even when baseline models have strong evidence for predicting no code-switch (i.e., ranking only monolingual phrases as important), they tend to misuse this history and randomly predict code-switches.

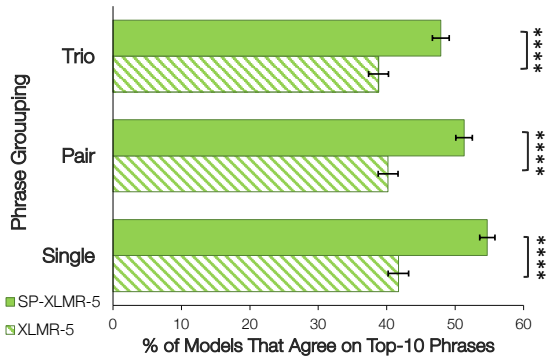

Figure 2: Each bar indicates the average percent and standard deviation of XLMR (dashed green, N=10) and SP-XLMR (green, N=30) models that agree on the top-10 phrases. We consider single phrases, as well as pairs and trios of phrases. There is significantly less agreement ($p < 0.0001$) among XLMR models as compared to SP-XLMR, potentially accounting for the higher standard deviation in XLMR models' scores.

On code-switched examples, speaker models continue to favor phrases that are nearest to the switchpoint, while baseline models are sensitive to phrases in early and late dialogue context. Using phrases closer to switch points may give better structural context from which to predict a switch. In several cases, speaker models correctly predict an English-to-Spanish switch and rank prior Spanish phrases as influential, while baseline models highly rank English phrases and predict no switch. We see a similar pattern in Spanish-to-English switches. Speaker information may help models learn, linguistically, what it means to code-switch.

7.3 Analyzing Speaker Phrases

Linguistic preference features are most influential across model setups. For all speaker-prompted models, speakers' language preferences are the most influential on the resulting predictions. Country of origin information is helpful, too, but may be misleading: speakers may immigrate from a Spanish country but grow up speaking English; in such cases, the language information likely helps disambiguate any confusions. Following these linguistic features are relational features (speaker order) in the dialogue, and less often, age features, especially in partner models. Gender is almost never influential. Our findings confirm prior work in sociolinguistics, which states that individual macro-social features rarely influence resulting dialogues; instead, cultural identities and social relations to larger groups are more important (Eckert, 2012; Ochs, 1992). Macro-social attributes may be influential in partner models because these explicitly access relationships between speakers; the "participant constellation" influences how speaker express themselves and modulate social distance (Giles and Baker, 2008; Myslín and Levy, 2015).

Ablating Features All speaker prompts contain the 6 defined attributes $\mathcal{A}$ (Section 5.1). Using the best-performing setups on the validation

set, namely Partner and Sentence models with 5 prior utterances for context, we identify influential speaker attributes using a leave-one-out-approach to mask out each attribute $a_i \in \mathcal{A}$. For each attribute, we train 10 ablated models and evaluate on the validation set. Note that this is different from the phrase ablations using LIL because we finetune the XLM-R encoder during the training process; in this setup, the ablated feature information is never backpropagated to update the encoder weights.

The results of these experiments (see Appendix A.3) give some evidence that language preference, mixing, and age information have statistically significant effects on the performance of Partner-5 models, but this does not hold for the Sentence-5 models. We have strong evidence to believe that these speaker attributes have more complex underlying relationships and leave the exploration of these multi-feature interactions for future work.

8 Related Work

Our use of prompts[2] is similar to Zhong et al. (2021) and Wei et al. (2021), who rely on prompts to put models in different states for different tasks.

Speaker Personas Open-domain dialogue agents which act according to a persona are more natural and engaging than the non-personalized baselines (Li et al., 2016); these personas can be short, superficial descriptions generated through crowdsourcing (Zhang et al., 2018), gathered from Reddit (Mazaré et al., 2018), or self-learned (inferred) from dialogue context (Madotto et al., 2019; Cheng et al., 2019). These works, however, primarily evaluate dialogue *content* and only in one language (English) instead of analyzing how speaker properties influence the downstream dialogue structure.

Addressing Model Bias Prior works for mitigating social biases feature adversarial learning (Pryzant et al., 2018; Elazar and Goldberg, 2018), counterfactual data augmentation (Zmigrod et al., 2019; Kaushik et al., 2020) or dataset balancing (Zhao et al., 2017), and more recently, using an interpretability-driven approach to uncover and controllably demote hidden biases (Han and Tsvetkov, 2021). Techniques for adapting to linguistic variants and mixed-language data include adversarial learning to pick up on key linguistic cues (Kumar et al., 2021), augmenting datasets with synthetic text (Winata et al., 2019) or examples of variants that models underperform on (Chopra et al., 2021), discriminative learning (Gonen and Goldberg, 2018), and transfer learning with morphological cues (Aguilar and Solorio, 2019).

Codeswitch Prediction The first work in codeswitch prediction (Solorio and Liu, 2008) uses Naive Bayes (NB) on lexical and syntactic features of shallow word context before switch boundaries. They run experiments on a small, self-collected dataset of English-Spanish conversations. Another NB approach predicts switch points on Turkish-Dutch social media data (Papalexakis et al., 2014), additionally using multi-word expressions and emoticons in their experiments. (Piergallini et al., 2016) extends the techniques of the prior two works to Swahili-English codeswitched data. A fine-grained logistic regression analysis (Myslín and Levy, 2015) goes beyond lexical information to incorporate psycholinguistic properties, such as word accessibility, and includes a binary feature to mark whether or not younger speakers are present.

9 Conclusion

We presented a methodology that interprets and directly compares sociolinguistic generalizations made by a neural text classifier. To the best of our knowledge, this is the first work that incorporates sociolinguistically-grounded social factors for predicting code-switch points. We demonstrated that our speaker-aware models can better leverage mixed-language linguistic cues, compared to a text-only baseline: specifically, we showed performance gains of up to 7% in accuracy and 5 points in F1 scores on an imbalanced code-switching dataset.

In the future, we plan to explore whether such speaker prompting can help models learn better inductive biases in *other* person-centered tasks, e.g., coreference resolution (especially for datasets explicitly testing gender biases) or sentiment analysis. Using techniques such as data augmentation, we will explicitly guide models away from biases learned during the training phase. Moreover, we will move from static to dynamic personas that are reflective of local dialogue context. Speaker-grounded models must be carefully engineered to protect user privacy, using proxies for personal information and keeping private information away from shared resources. With ethical considerations in mind, our work advances the state-of-the-art in building more adaptable and person-aware NLP technologies.

[2] Our prompts are data-dependent and fixed, and thus rather unrelated to the prompt tuning literature (Liu et al., 2021).

References

Gustavo Aguilar and Thamar Solorio. 2019. From english to code-switching: Transfer learning with strong morphological clues. *CoRR*, abs/1909.05158.

Emily Ahn, Cecilia Jimenez, Yulia Tsvetkov, and Alan W Black. 2020. What code-switching strategies are effective in dialogue systems? In *Proceedings of the Society for Computation in Linguistics (SCiL) 2020*.

Ruthanna Barnett, Eva Codó, Eva Eppler, Montse Forcadell, Penelope Gardner-Chloros, Roeland Van Hout, Melissa Moyer, Maria Carme Torras, Maria Teresa Turell, Mark Sebba, et al. 2000. The lides coding manual: A document for preparing and analyzing language interaction data version 1.1–july 1999. *International Journal of Bilingualism*, 4(2):131–271.

Anshul Bawa, Pranav Khadpe, Pratik Joshi, Kalika Bali, and Monojit Choudhury. 2020. Do multilingual users prefer chat-bots that code-mix? let's nudge and find out! *Proc. ACM Hum.-Comput. Interact.*, 4(CSCW1).

Anne L. Beatty-Martínez, Christian A. Navarro-Torres, and Paola E. Dussias. 2020. Codeswitching: A bilingual toolkit for opportunistic speech planning. *Frontiers in Psychology*, 11:1699.

Holly P Branigan, Martin J Pickering, Jamie Pearson, and Janet F McLean. 2010. Linguistic alignment between people and computers. *Journal of Pragmatics*, 42(9):2355–2368.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pages 267–284.

Hao Cheng, Hao Fang, and Mari Ostendorf. 2019. A dynamic speaker model for conversational interactions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2772–2785, Minneapolis, Minnesota. Association for Computational Linguistics.

Parul Chopra, Sai Krishna Rallabandi, Alan W Black, and Khyathi Raghavi Chandu. 2021. Switch point biased self-training: Re-purposing pretrained models for code-switching.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

M Deuchar, P Davies, J R Herring, M Parafita Couto, and D Carter. 2014. Building bilingual corpora.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

A. Seza Doğruöz, Sunayana Sitaram, Barbara E. Bullock, and Almeida Jacqueline Toribio. 2021. A survey of code-switching: Linguistic and social perspectives for language technologies. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1654–1666, Online. Association for Computational Linguistics.

Penelope Eckert. 2012. Three waves of variation study: The emergence of meaning in the study of sociolinguistic variation. *Annual review of Anthropology*, 41:87–100.

Yanai Elazar and Yoav Goldberg. 2018. Adversarial removal of demographic attributes from text data. In *Proc. EMNLP*.

Anjalie Field, Su Lin Blodgett, Zeerak Waseem, and Yulia Tsvetkov. 2021. A survey of race, racism, and anti-racism in NLP. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1905–1925, Online. Association for Computational Linguistics.

Howard Giles and Susan C Baker. 2008. Communication accommodation theory. *The international encyclopedia of communication*.

Hila Gonen and Yoav Goldberg. 2018. Language modeling for code-switching: Evaluation, integration of monolingual data, and discriminative training. *CoRR*, abs/1810.11895.

Xiaochuang Han and Yulia Tsvetkov. 2021. Influence tuning: Demoting spurious correlations via instance attribution and instance-driven updates.

Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. 2017. Deep models under the gan: information leakage from collaborative deep learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 603–618.

Dirk Hovy and Diyi Yang. 2021. The importance of modeling social factors of language: Theory and practice. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 588–602.

Divyansh Kaushik, Eduard Hovy, and Zachary Chase Lipton. 2020. Learning the difference that makes a difference with counterfactually-augmented data. In *Proc. ICLR*.

Svetlana Kiritchenko and Saif M. Mohammad. 2018. Examining gender and race bias in two hundred sentiment analysis systems.

Jakub Konečný, H. Brendan McMahan, Daniel Ramage, and Peter Richtárik. 2016. Federated optimization: Distributed machine learning for on-device intelligence. *CoRR*, abs/1610.02527.

Sachin Kumar, Shuly Wintner, Noah A. Smith, and Yulia Tsvetkov. 2021. Topics to avoid: Demoting latent confounds in text classification.

Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. 2016. A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 994–1003, Berlin, Germany. Association for Computational Linguistics.

Wei Li. 2013. Codeswitching. In *The Oxford Handbook of Chinese Linguistics*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Andrea Madotto, Zhaojiang Lin, Chien-Sheng Wu, and Pascale Fung. 2019. Personalizing dialogue agents via meta-learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5459, Florence, Italy. Association for Computational Linguistics.

Pierre-Emmanuel Mazaré, Samuel Humeau, Martin Raison, and Antoine Bordes. 2018. Training millions of personalized dialogue agents. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2775–2779, Brussels, Belgium. Association for Computational Linguistics.

Mark Myslín and Roger Levy. 2015. Codeswitching and predictability of meaning in discourse. *Language*, 91.

Debora Nozza, Federico Bianchi, and Dirk Hovy. 2021. Honest: Measuring hurtful sentence completion in language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2398–2406.

Eleanor Ochs. 1992. 14 indexing gender. *Rethinking context: Language as an interactive phenomenon*, 11(11):335.

Evangelos Papalexakis, Dong Nguyen, and A Seza Doğruöz. 2014. Predicting code-switching in multilingual communication for immigrant communities. In *First Workshop on Computational Approaches to Code Switching (EMNLP 2014)*, pages 42–50. Association for Computational Linguistics (ACL).

Mario Piergallini, Rouzbeh A. Shirvani, Gauri Shankar Gautam, and Mohamed F. Chouikha. 2016. Word-level language identification and predicting codeswitching points in swahili-english language data. In *CodeSwitch@EMNLP*.

Reid Pryzant, Kelly Shen, Dan Jurafsky, and Stefan Wagner. 2018. Deconfounded lexicon induction for interpretable social science. In *Proc. NAACL-HLT*.

Dheeraj Rajagopal, Vidhisha Balachandran, Eduard Hovy, and Yulia Tsvetkov. 2021. Selfexplain: A self-explaining architecture for neural text classifiers.

Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A. Smith. 2019. The risk of racial bias in hate speech detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1668–1678, Florence, Italy. Association for Computational Linguistics.

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

Tony Sun, Andrew Gaut, Shirlyn Tang, Yuxin Huang, Mai ElSherief, Jieyu Zhao, Diba Mirza, Elizabeth Belding, Kai-Wei Chang, and William Yang Wang. 2019. Mitigating gender bias in natural language processing: Literature review. In *Proc. of ACL*, pages 1630–1640.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Genta Indra Winata, Samuel Cahyawijaya, Zihan Liu, Zhaojiang Lin, Andrea Madotto, and Pascale Fung. 2021. Are multilingual models effective in code-switching? *CoRR*, abs/2103.13309.

Genta Indra Winata, Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2019. Code-switched language models using neural based synthetic data from parallel sentences. *CoRR*, abs/1909.08582.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics.

Yuheng Zhang, Ruoxi Jia, Hengzhi Pei, Wenxiao Wang, Bo Li, and Dawn Song. 2020. The secret revealer: Generative model-inversion attacks against deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 253–261.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2017. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.

Ruiqi Zhong, Kristy Lee, Zheng Zhang, and Dan Klein. 2021. Adapting language models for zero-shot learning by meta-tuning on dataset and prompt collections. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2856–2878.

Ran Zmigrod, Sabrina J. Mielke, Hanna Wallach, and Ryan Cotterell. 2019. Counterfactual data augmentation for mitigating gender stereotypes in languages with rich morphology. In *Proc. ACL*.

A Appendix

A.1 Code-Switching Example

Figure 3 provides a key motivating example of how global speaker features of two conversational participants, ID'd RIC and SEB, influence their local speech production. RIC was raised in the United States and knows Spanish, while SEB is from a Spanish-speaking country and has a strong grasp of English. For most of the dialogue, RIC speaks English, unless he is specifically accommodating to seb, as we see in the very first example utterance. RIC demonstrates more intrasentential (within-utterance) switches, often switching back to English, which corresponds to his preference for English (Beatty-Martínez et al., 2020). SEB accommodates to RIC by responding in English with *Yeah, she knows about it?*, but, similarly to RIC, relies on Spanish to express vocabulary or phrases that are more complex for him (i.e., *foreseeing the future*).

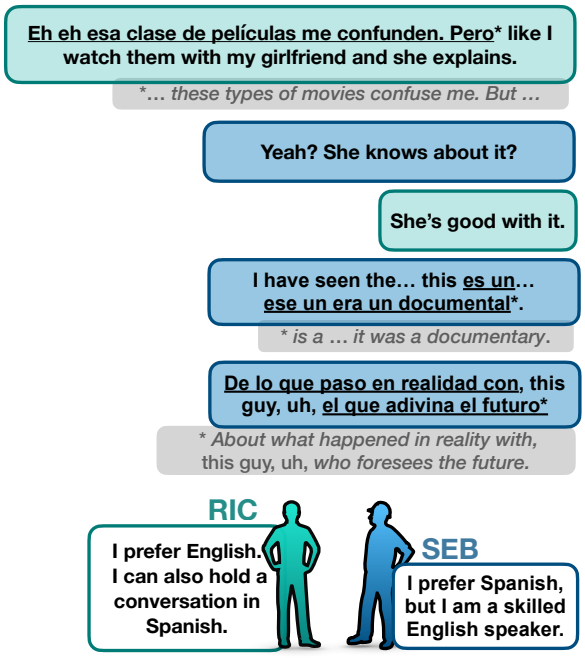

Figure 3: A dialogue between two speakers, whose IDs are RIC and SEB. RIC and SEB typically switch to to the languages they prefer: English and Spanish, respectively. RIC and SEB also mix languages to accommodate each other, demonstrating the need for speaker awareness in code-switched language processing.

A.2 Scoring Phrase Relevance

Note that the original implementation scored phrases using the raw softmax difference to reflect the contribution of each phrase. To further distinguish the predictive power of various local phrases, we give meaning to the sign of a score. This disambiguates edge cases such as the following: (1) the softmax score of the predicted class $z_F = 0.6$ and the phrase-ablated sentence yields a softmax score, $z_A = 0.4$ (2) or, we have the scenario that $z_{F'} = 0.9$ and $z_{A'} = 0.7$; in both scenarios, subtraction yields $z_{F'} - z_{A'} = 0.2$, thus, the purported relevance of the ablated phrases is the same. However, we would like to distinguish that in case 1), the ablated phrase causes an overall change in prediction, unlike case 2), and thus, the case 1) phrase would be more relevant than the case 2) phrase.

A.3 Speaker Ablation Results

	Partner-5		Sentence-5	
Feature	**Acc. (%)**	**F1**	**Acc. (%)**	**F1**
Full	78.9 ± 1.23	62.9	78.2 ± 1.86	63.2
Language	*76.9 ±1.82	62.7	77.7 ±1.84	62.9
Mixing	*77.8 ±1.34	63.2	78.9 ±1.72	63.6
Country	79.0 ±1.54	63.5	78.4 ±1.60	63.2
Order	78.9 ±1.75	63.1	78.4 ±1.30	63.2
Gender	78.0 ±1.90	63.0	77.6 ±2.12	62.7
Age	*77.5 ±1.09	62.6	79.1 ±1.80	63.4

Table 5: Average accuracy and F1 scores of speaker-ablated Partner-5 and Sentence-5 models on the validation set. N=10 for both setups. Full (non-ablated) models are included for comparison. Starred results are significant ($p < 0.05$) by Mann-Whitney U Tests.

Model Type	Context	Acc. (%)	F1	Recall	Precision
Majority	-	75.0	–	–	–
List	1	75.9 ± 1.390	60.1	72.3	51.5
List	2	77.4 ± 0.932	60.9	70.4	53.8
List	3	78.3 ± 1.191	62.0	71.0	55.3
List	5	78.9 ± 1.418	63.1	72.1	56.4
Partner	1	77.4 ± 2.301	61.1	70.6	54.1
Partner	2	78.3 ± 0.966	61.8	70.4	55.2
Partner	3	78.7 ± 0.982	62.6	71.2	55.9
Partner	5	78.8 ± 1.228	62.9	71.7	56.2
Sentence	1	77.5 ± 1.667	60.5	68.9	54.2
Sentence	2	77.9 ± 1.210	61.6	71.0	54.6
Sentence	3	78.2 ± 1.255	62.1	69.5	56.3
Sentence	5	78.2 ± 1.863	63.2	74.5	55.1
XLMR	1	66.4 ± 2.836	54.0	78.9	41.3
XLMR	2	70.3 ± 3.269	57.2	79.0	45.2
XLMR	3	71.4 ± 1.916	58.2	79.6	46.0
XLMR	5	73.1 ± 2.739	59.8	79.9	48.0

Table 6: Performance of all models on the validation set (25.0% code-switched). Each value is an average of N=10 models.

Model Type	Context	Acc. (%)	F1	Recall	Precision
Majority	-	74.8	–	–	–
List	1	78.9 ± 1.247	61.4	66.7	57.2
List	2	79.7 ± 1.063	63.0	68.5	58.4
List	3	80.0 ± 0.769	64.0	70.4	58.8
List	5	80.2 ± 0.919	64.3	70.9	59.0
Partner	1	78.5 ± 1.966	59.8	63.5	57.0
Partner	2	79.4 ± 1.120	62.4	67.8	58.1
Partner	3	80.0 ± 0.927	63.5	69.0	58.9
Partner	5	79.5 ± 1.268	62.9	69.2	58.1
Sentence	1	79.0 ± 1.260	60.9	65.9	57.5
Sentence	2	79.6 ± 1.095	63.4	70.4	57.9
Sentence	3	80.0 ± 1.021	63.4	69.4	58.6
Sentence	5	79.4 ± 1.602	64.4	74.2	57.3
XLMR	1	69.5 ± 2.755	56.5	78.7	44.3
XLMR	2	71.8 ± 2.231	58.7	79.7	46.7
XLMR	3	72.4 ± 2.312	59.8	81.3	47.4
XLMR	5	74.2 ± 2.394	61.2	80.9	49.5

Table 7: Performance of all models on the test set (25.2% code-switched). Each value is an average of N=10 models.

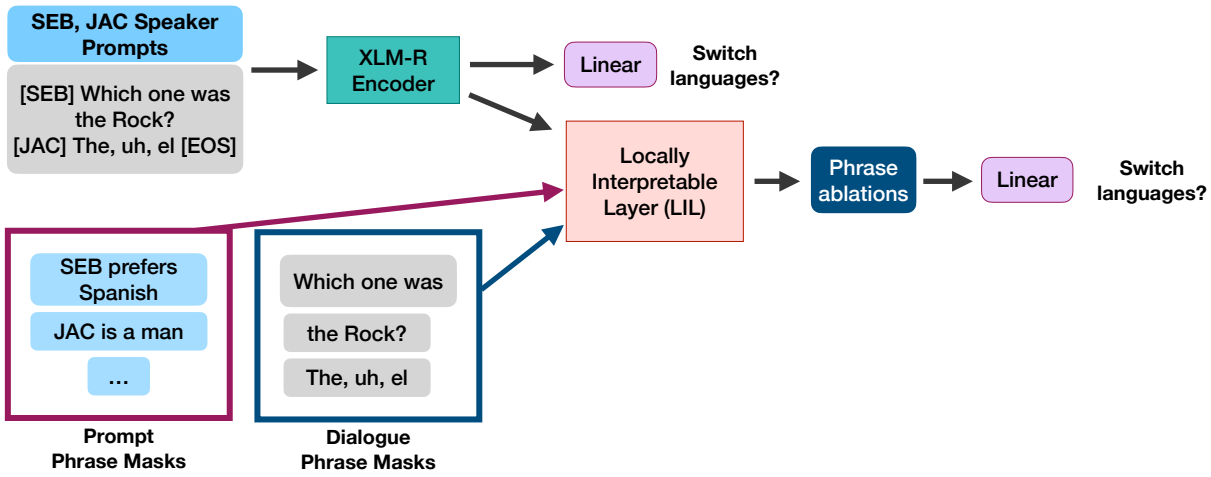

Figure 4: Architecture diagram of our proposed speaker-prompted code-switch prediction models. The input to the model is the dialogue context (gray) with descriptions of each speaker (blue) prepended to the dialogues. We encode the input using XLM-R (dark green) and use a linear layer (purple, top) to predict whether to code-switch or not. The encoded sentence, along with phrase masks, is passed to the Locally Interpretable Layer (LIL) (Rajagopal et al., 2021); using phrase ablation, LIL highlights influential phrases in the input by comparing local predictions to the full-sentence prediction. Baseline models follow a similar setup, but without any input from speaker prompts.