Language Informed Modeling of Code-Switched Text

***Khyathi Raghavi Chandu, *Thomas Manzini, *Sumeet Singh, Alan W Black**

Language Technologies Institute, Carnegie Mellon University

kchandu, tmanzini, sumeets, awb(@cs.cmu.edu)

1. Motivation

Code Switching (CS) is the alternation between languages in conversations. Wide-spread among Multi-lingual speakers. Examples,

- Spanglish = Spanish + English *(Puerto-Rico, Mexico, US)*
- Hinglish = Hindi + English *(South-East Asia)*
- Chinglish = Cantonese + English *(Hong Kong)*

Why Care?

A large section of humanity ignored by NLP community because **they lack:**

- Standardized Datasets
- Language Models
- Language Parsers
- And More!

2. Introduction

Why Language Modeling(LM)?

- Important for downstream applications: Machine Translation, Speech Recognition, etc.
- CS data severely lacks annotation - needs unsupervised methods.

Challenges for NLP?

- Undefined grammar rules (*Matrix* vs *Embedded*).
- Occur in informal contexts (*hard to get, extremely noisy*).

3. Dataset

What Data?

- Authors collected and curated 59189 unique CS sentences. Data was gathered from a web crawl of eight Hinglish blogging websites that were returned by popular search engines (such as Google and Bing).
- All sentences were drawn from the domains of health and technology.
- Lexical language identification was performed at the word level for all sentences.
- Sentences that did not have at least one word each from both languages were discarded to channel our problem towards tackling intra-sentential code-switching.

Criteria	Train	Dev	Test
# Sentences	35513	11839	11837
Multilingual Index	0.8892	0.8905	0.8914
Language Entropy	0.6635	0.6639	0.6641
Integration Index	0.3304	0.3314	0.3312
Unique Unigrams	35,769	18,053	19,330
Unique Bigrams	276,552	125,108	130,947
Unique Trigrams	553,866	219,098	229,967

Table 1: *Hinglish Blogs Data Statistics*

Spell Normalization is done by:

- Language Identification.
- Transliteration of *romanized* Hindi to its most likely *Devanagari* spelling using *soundex* encodings.

4. Approach

We address the task of language modeling in *Hinglish* text with a dual objective:

- Predicting the next word.
- Predicting the language of the next word.

Underlying Theme: *Simultaneous learning on multiple language tasks with shared parameterized layers improves generalization.*

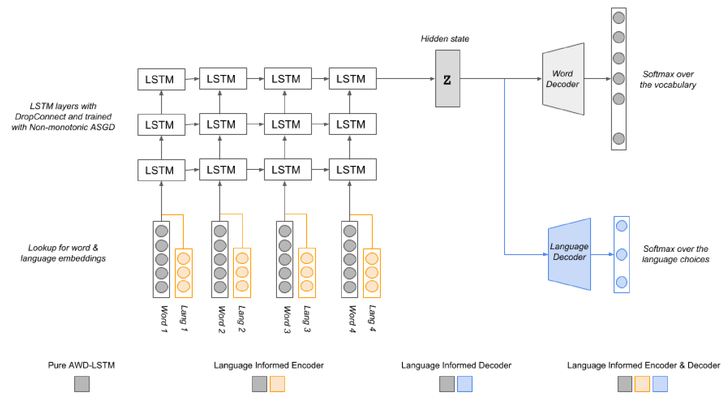

Figure 1: *CS LM models that we explored.*

Model Architecture

- Implemented a baseline model of AWD-LSTM.
- 4 model architectures differ depending upon whether the language information is encoded or decoded.
- The distributional aspects of switching between languages captured by 16 dimensional language embedding.

5. Results and Discussion

What Happened?

- We trained and evaluated 4 different models.
- Note that these experiments are ongoing and we believe that we can improve these results with further hyperparameter tuning.

Model/Data	Train	Dev	Test
Base AWD-LSTM Model	10.08	19.73	20.92
Language Aware Encoder AWD-LSTM	10.07	19.00	20.18
Language Aware Decoder AWD-LSTM	11.60	20.72	22.01
Language Aware Encoder & Decoder AWD-LSTM	9.47	18.51	19.52

Table 2: *Perplexity scores of different models*

- Language Aware Encoding with the AWD-LSTM gives the best perplexity.
- This aligns with our hypothesis that providing language information aids in learning switch points.
- Model encodes the language of the current word and decodes the language of the next word.

What Could Be Better?

- Since most of the articles belong to the topics of e-commerce, latest technology and health, robustness of the model may be affected.
- Non-standardized spellings in the *romanized* Hinglish text cannot always be found in the MUSE embeddings.

6. Conclusion and Future Work

What Did We Do?

- We are able to improve the State-of-The-Art language model for monolingual text by explicitly encoding the language information to perform this task for CS domain.
- Incorporating both language encoder and decoder performed comparatively better.

What Next?

- We aim to incorporate FastText word embeddings for Devanagiri.
- End to End character level models is another direction to try.
- **We welcome any and all feedback on any portion of this work.**

7. References

[1] Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2017. Regularizing and optimizing lstm language models. arXiv preprint arXiv:1708.02182.

[1] **Authors contributed equally to this work.*