PESTO: Switching Point Based Dynamic and Relative Positional Encoding for Code-Mixed Languages (Student Abstract)

Mohsin Ali,[1] **Sai Teja Kandukuri,** [1] **Sumanth Manduru,**[1]
Parth Patwa,[2] **Amitava Das** [3, 4]
IIIT Sri City, India[1] UCLA, USA[2]
Wipro AI Labs, India[3] AI Institute, University of South Carolina, USA[4]
{mohsinali.m18, saiteja.k18, sumanth.m15}@iiits.in, parthpatwa@ucla.edu, amitava.das2@wipro.com

Abstract

NLP applications for code-mixed (CM) text have gained a significant momentum recently, mainly due to the prevalence of language mixing in social media communications in multilingual societies like India, Europe, parts of USA etc. Word embeddings are basic building blocks of any NLP system today, yet, word embedding for CM languages is an unexplored territory. The major bottleneck for CM word embeddings is switching points, where the language switches. These locations lack in contextually and statistical systems fail to model this phenomena due to high variance in the seen examples. In this paper we present our initial observations on applying switching point based positional encoding techniques for CM language, specifically *Hinglish* (Hindi - English). Results are only marginally better than SOTA, but it is evident that positional encoding could be an effective way to train position sensitive language models for CM text.

Switching Points: The Bottleneck

Switching Points (SPs) are the positions in CM text, where the language switches. Consider the text - *aap*$_{\text{HI}}$ *se*$_{\text{HI}}$ *request*$_{\text{EN}}$ *hain*$_{\text{HI}}$ (request you to). Here, when the language switches from *Hindi* to *English* (*se*$_{\text{HI}}$ *request*$_{\text{EN}}$) a **HI-EN** (HIndi-ENglish) SP occurs. Correspondingly, a **EN-HI** SP occurs at *request*$_{\text{EN}}$ *hain*$_{\text{HI}}$. In this work we to look at sentiment analysis of CM languages, specifically *Hinglish* through the lens of language modeling. We propose PESTO - a switching point based dynamic and relative positional encoding. PESTO learns to emphasis on switching points in CM text. Our model marginally outperforms the SOTA.

Background - Dataset and Positional Encoding

Data and SOTA: The SentiMix task (Patwa et al. 2020) released 20K *Hinglish* tweets, having word-level language labels and sentence-level sentiment (*positive, negative, neutral*). Liu et al. (2020a) achieved the SOTA (75% f1 score) by fine-tuning XLM-R using adversarial training.

Vaswani et al. (2017) introduced *Positional Encoding (PE)* for language modeling. PE serves as an added feature along with the word embeddings, providing both *relative* and *absolute* positional relations between a target word and its context words.

Absolute Positional Encoding (APE)

Sinusoidal PE: - A predefined sinusoidal vector $p_i \in R^d$ is assigned to each position *i*. This p_i is added to the word embedding $w_i \in R^d$ at position *i*, and $w_i + p_i$ is used as input to the model. In this way, the Transformer can differentiate the words coming from different positions and assign each token position-dependent attention Vaswani et al. (2017). Sin/-cos functions are used interchangeably to capture odd/even numbered positional words in a sequence - equation 1

$$\alpha_{ij}^{abs} = \frac{1}{\sqrt{d}} \left((w_i + p_i) W^{Q,1} \right) \left((w_j + p_j) W^{K,1} \right)^T \quad (1)$$

Dynamic PE: - Instead of using periodical functions like sin/cos, Liu et al. (2020b), proposed to learn a dynamic function at every encoder layer that can represent the positional info. A function $\theta(i)$ is introduced which can learn positional info with gradient flow. - equation 2

$$\alpha_{ij} = \tfrac{1}{\sqrt{d}} \left((w_i + \theta(i)) W^{Q,1} \right) \left((w_j + \theta(j)) W^{K,1} \right)^T \quad (2)$$

Relative Positional Encoding (RPE)

Shaw, Uszkoreit, and Vaswani (2018) introduced a learnable parameter a_{j-i}^l which learns the positional representation of the relative position *j-i* at encoder layer *l*. This helps the model to capture relative word orders explicitly - equation 3

$$\alpha_{ij}^{rel} = \frac{1}{\sqrt{d}} \left((w_i)^l W^{Q,l} \right) \left((w_i)^l W^{K,l} + a_{j-i}^l \right)^T \quad (3)$$

Switching Point based Positional Encoding

We introduce a novel, switching point based PE. Consider the *Hinglish* sequence - *gaaye*$_{\text{HI}}$ *aur*$_{\text{HI}}$ *dance*$_{\text{EN}}$ *kare*$_{\text{HI}}$. SP based indices (SPI) - i) We set the index to 0 whenever an SP occurs. Indexing would normally be = $\{0, 1, 2, 3\}$, we change it to $\{0, 1, 0, 0\}$. ii) We consider Hindi as our base language and English as the mixed language. We set the index to 0 only when the shift is from base language to mixed language. So, the resultant index would be $\{0, 1, 0, 1\}$.

Switching Point based Dynamic PE (SPDPE)

We introduce a function $S(l_i)$, which takes the word level language labels as input and returns SPI. Instead of passing an index directly as i to θ, we use $\theta(s(l_i))$ to dynamically learn the PE based on SPI - equation 4

$$\alpha_{ij} = \frac{1}{\sqrt{d}} \left(\left(w_i + \theta\left(S(l_i)\right)\right) W^{Q,1} \right) \left(\left(w_j + \theta\left(S(l_j)\right)\right) W^{K,1} \right)^T \quad (4)$$

PESTO - Switching Point based Dynamic and Relative PE (SPDRPE)

Here, in addition to the SPDPE, we use a learning parameter a^l_{j-i}, which encodes the relative position *j-i* at the encoder layer *l*. This encoding approach learns representations dynamically based on SPs along with the embedding a^l_{j-i} so that it can also capture relative word orders (equation 5).

$$\alpha_{ij} = \frac{1}{\sqrt{d}} \left(\left(w_i + \theta\left(S(l_i)\right)\right)^l W^{Q,l} \right) \left(\left(w_j + \theta\left(S(l_j)\right)\right)^l W^{K,l} + a^l_{j-1} \right)^T \quad (5)$$

Models

Baselines - Word2Vec, Multi Head Attention (MHA): We choose Word2Vec as the baseline since it does not capture position info. We also choose attention mechanism, which is widely used to capture relational dependencies, to see its effects over SPs. We experiment with two lengths - i) Length 3 to capture the local window of dependency, whereas, ii) 12 to see whether it can learn anything from the whole sentence. 12 is the average length of sentences in our corpus.

PESTO Architecture (Fig. 1): The local dependencies from Word2Vec (trained from scratch) along with SPI obtained from SPDRPE are passed to a 12 headed transformer based encoder layer. On top of the transformer, a 1D CNN is used to get the sentence level representation. We also obtain the sentence embedding using tf-idf weighted average of Word2Vec embeddings. Finally, we concatenate the representations of the CNN and the tf-idf sentence embedding and pass it to a dense layer which applies softmax to predict the sentiment. We train the entire model (2 encoder layers) from scratch, without using any pre-trained model.

Results

PESTO achieves 75.56% F1 score and outperforms SOTA (Tab. 1). The main reason for this is learning SP by aggregating both relative and dynamic PE with a variable length MHA framework. PESTO is able to generate more thrust to the switching point *weather*$_{\text{EN}}$ *achaa*$_{\text{HI}}$ (Fig. 2). The experiments were conducted on google Colab. The code is available at https://github.com/mohammedmohsinali/PESTO.

Conclusion

In this paper we report initial experiments on *Hinglish* sentiment analysis problem through the lens of language modeling. Our contribution could be seen as following - i) We introduce the idea of switching-point based positional encoding. i) We propose a relative switching point dynamic positional encoding technique named PESTO, which yields better results than SOTA. iii) It is also noteworthy that our model - PESTO achieves SOTA results without any pre-trained heavy language model, whereas all the SOTA models in the SentiMix task used models like BERT, or XLNet.

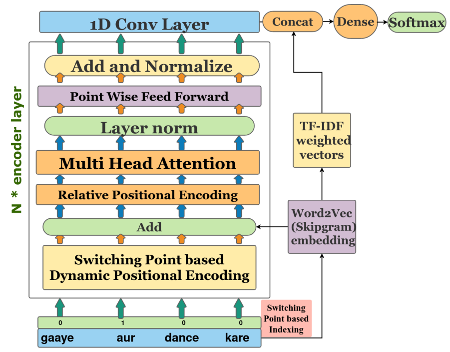
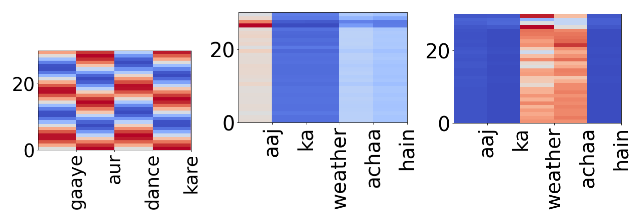


Figure 1: PESTO - Proposed model for Relative Switching Point based Dynamic Positional Representation for CM text.

Models	Positional Representation				F1 (%)
	Sin/Cos	Index	SPI	Relative	
Word2Vec + LSTM	✗	✗	✗	✗	56
BERT	✓	✗	✗	✗	60
3HA with Sinusoidal PE	✓	✗	✗	✗	65
3HA with Dynamic PE	✗	✓	✗	✗	69.7
12HA with Dynamic PE+RPE	✗	✓	✗	✓	73
12HA with RPE	✗	✗	✗	✓	73.4
12HA with SPDPE	✗	✗	✓	✗	73.52
SOTA (Liu et al. 2020a)	✓	✗	✗	✗	75
PESTO (12HA with SPDRE)	✗	✗	✓	✓	**75.56**

Table 1: Results of various position sensitive experiments on CM text. It is evident from the results that positional encoding could be an effective way to train position sensitive language models for CM text.



(a) Sinusoidal PE (b) Dynamic PE (c) PESTO

Figure 2: PESTO differentiates words coming from different positions and also pays high attention to the SPs like *weather*$_{\text{EN}}$ and *achaa*$_{\text{HI}}$.

References

Liu, J.; et al. 2020a. kk2018 at SemEval-2020 Task 9: Adversarial Training for Code-Mixing Sentiment Classification. In *SemEval*.

Liu, X.; et al. 2020b. Learning to Encode Position for Transformer with Continuous Dynamical Model. In *ICML 2020*.

Patwa, P.; et al. 2020. SemEval-2020 Task 9: Overview of Sentiment Analysis of Code-Mixed Tweets. In *SemEval 2020*.

Shaw, P.; Uszkoreit, J.; and Vaswani, A. 2018. Self-Attention with Relative Position Representations. In *NAACL 2018*.

Vaswani, A.; et al. 2017. Attention Is All You Need. In *NeurIPS*.