**Urdu Word Embeddings**

**Summary of Technical Contributions:**

* This published paper describes the first ever attempt to create large-scale distributed representations of Urdu words using skip-gram variant of word2vec
* Urdu corpora totaling over 140m tokens was scraped from sources such as Urdu Wikipedia the rest was collected from (Jawaid et al., 2014) and (Adeeba et al., 2014).
* **Training parameters:**
  + Context Window Sizes: 3, 5, 7
  + Word Embedding Dimensions: 100, 200, 300
  + Learning rate: 0.025
  + Minimum Frequency Cutoff: 10 words
  + No. of Epochs: 5
  + No. of Sampled Noise Words: 5
* 100K vectors were generated against the models, the trained models are publicly available for use on GitHub
* The quality of learned vectors was evaluated using 269 word pairs of WordSim-353 and 691 word pairs SimLex-999 benchmark tests

**Method:**

Selected pairs were translated into Urdu and a quantitative analysis of the model’s embeddings was performed using the Spearman’s Correlation

* **Results:** 
  + Model with window size: 5 and dimensionality: 200 outperformed the rest of the models and gave a score of 0.524 for WordSim-353
  + Model with window size: 7 and dimensionality: 300 outperformed the rest of the models and gave a score of 0.306 for SimLex-999
  + Top 10 closely related words were generated for 3 examples to show a qualitative analysis of the embeddings

**Weaknesses:**

* The paper did not explore the strengths and weaknesses of CBOW variant or other word embedding techniques like fastText, GloVe etc. for Urdu
* The effect of larger context window sizes was not explored in depth
* The paper did not cater to Roman Urdu language
* The minimum cutoff frequency was kept to 10 which meant that rare words were neglected

**Strengths:**

* First attempt to generate distributed representation of words for Urdu
* Model was trained on a significantly large corpus of 140m tokens and the learned embeddings totaled to 100,000
* The Spearman’s Correlation scores were quite close to Mikolov’s scores on WordSim-353 and Simlex-999 benchmarks for English indicating that the embeddings caught semantic relations quite well

**Improvements:**

* Evaluate learned embeddings for window sizes greater than 10
* Set minimum frequency cutoff to 0 to cater to rare words
* Evaluate embeddings for CBOW variant using the same configuration of parameters to gain a comparative analysis between both variants

**Questions:**

* How would varying sizes of tokens affect the quality of the learned embeddings?
* Which variant performs better the CBOW or skipgram on the fed tokens?
* How does the model deal with rare words?
* How does the model deal with polysemy?
* What effect does varying window sizes have on the learned embeddings?
* How would this model fare against models like fastText, GloVe or deep neural networks like ELMO and BERT in terms of speed?