Natural Language Processing

Arabic Autocorrect Project Presentation

Model Overview

We used a sequence-to-sequence model using TensorFlow Keras. It is designed to process sequences of tokens (words or characters) and generated a predicted (in our case, corrected) version of the same sequence. This model is especially useful in our case, an Arabic Autocorrect program.

Layers:

- **Embedding Layer**
  - Converts each token into a dense vector of size 300
  - This allows the model to learn semantic relationships between tokens.
- **Bidirectional LSTM 1**
  - Processes the input sequence in both forward & backward directions using 256 units
  - Capture context from both past & future tokens.
- **Dropout**
  - Randomly drops 30% of the output units
  - Reduces overfitting & improves generalisation by preventing the model from becoming too reliant on specific neurons
- **Bidirectional LSTM 2**
  - Stacking LSTMs enables the model to learn deeper sequential patterns
- **TimeDistributed Dense Layer**
  - Applies a dense layer to each timestep independently
  - Converts each LSTM output into a probability distribution over the vocabulary, enabling token-level prediction.

Model Compilation:

- **Loss:** sparse_categorical_crossentropy
  - Used because our targets are integer-encoded tokens, not one-hot.
- **Metrics**: accuracy
  - Used because it measures how many predicted tokens match the true tokens.
- **Optimiser**: adam
  - A robust optimiser suitable for NLP tasks; combines momentum and adaptive learning rates.

Model Parameters:

- Embedding_dim = 300 – each word will be represented as a 300-dimensional dense vector
- Input_dim=vocab_size – number of unique tokens for embedding layer
- LSTM_units = 256 – number of hidden units in the first LSTM layer (per direction, since it's bidirectional)
- Dropout(0.3) – 30% of units are dropped during training to reduce overfitting
- Activation='softmax' – turns output into a probability distribution over all tokens

Dataset Information

Our dataset was sourced from https://github.com/elnagara/BRAD-Arabic-Dataset/tree/master

It is a dataset of 510,600 book reviews in Arabic, scraped from GoodReads.com during a two-month period in 2016. Reviews are primarily in Modern Standard Arabic, but some are in dialectical arabic.

Citation:

Elnagar A. and Einea O. 'BRAD 1.0: Book reviews in Arabic dataset'. 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA), pp. 1-8, Nov 2016. DOI: 10.1109/AICCSA.2016.7945800.

Model Limitations

We have found that Bidirectional LSTM struggles to perform well when tasked with deletion, insertion, transposition and substitution errors all at the same time (89% word-level accuracy, but 12-20% sentence-level accuracy). We therefore picked the most common error (substitution) and decreased the amount of noise we trained the model and got an accuracy of ~99%, with a sentence-level accuracy of 79% (when tested on 200 random noisy-clean pairs from the dataset).

The model is primarily most successful when it is given a three-to-five word Arabic sentence, with one (or potentially two) substition error(s). Any longer than that, and it will struggle. This is due to the nature of Bidirectional LSTM and autocorrect as a concept – it can almost never be perfect.

Traditional Autocorrect systems found on our devices (such as on the iPhone) are trained on significantly larger datasets, and are trained for much longer, and are constantly being improved. It is likely impossible to achieve a sentence-level accuracy of 90%+ with the methods at our disposal.