**NLP Project Milestone 2 Report**

Ziad Amr Shamseldein (52-0295)

Hazem Sherif (52-5272)

Youssef Bayoumi (52-11219)

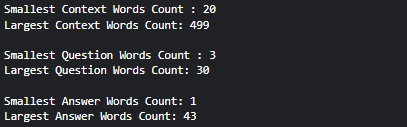
**Introduction**

This report examines the development of a sequence-to-sequence question answering model using the SQuAD dataset. The implementation progressed through two phases: a basic encoder-decoder architecture and an enhanced version incorporating attention mechanisms and dual encoders for question and context processing.

**Data Processing**

The preprocessing pipeline began with selecting 15,000 training and 5,000 evaluation samples from SQuAD based on context length. Text normalization included lowercase conversion and punctuation removal. The tokenization process created a vocabulary with special tokens for answer boundaries and used pre-trained GloVe embeddings (50-dimensional) while handling out-of-vocabulary words. Sequences were padded to a uniform length determined through analysis of the dataset statistics.

Postprocessing involved converting the model's token index outputs back to text while removing special boundary markers before final evaluation. The data analysis revealed that answers in the subset ranged from single tokens to multi-word phrases, with contexts varying significantly in length even within the shortened selection.



**Model Architecture**

The system evolved through two architectural approaches. The initial implementation featured a single LSTM encoder (256 units) processing questions and a corresponding decoder generating answers token-by-token. This basic seq2seq model demonstrated the fundamental challenges of QA generation, particularly in maintaining relevance across longer sequences.

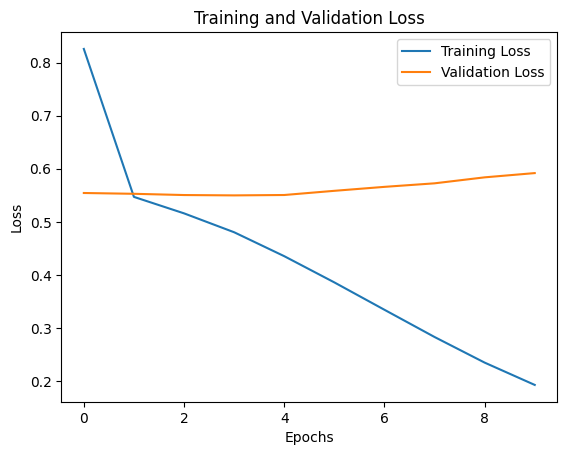
The enhanced architecture introduced several key improvements: separate encoders for questions and contexts (concatenated together), a Bahdanau attention mechanism, and an expanded decoder (512 units). This design allowed the model to better align question intent with relevant context segments during answer generation. The attention layer computed dynamic weights to focus on different parts of the input sequence at each decoding step, while the concatenated encoder outputs provided richer semantic representations.

**Training Methodology**

Training employed Adam optimization with sparse categorical cross entropy loss, appropriate for the token classification task. The model processed batches of 32 sequences over 10 epochs, with a 20% validation split monitoring token-level accuracy. Teacher forcing helped stabilize training by feeding correct previous tokens during decoder steps.

The hyperparameter selection balanced computational constraints against model capacity, with the 50-dimensional embeddings providing reasonable coverage while the 256/512-unit LSTMs offered sufficient representational power.

**Results and Limitations**



Training curves indicated reasonable convergence but suggested potential overfitting, possibly due to the model's lack of regularization techniques.

The model was evaluated using the SQuAD evaluation metric which includes:

* Exact Match (EM): Percentage of predictions matching ground truth exactly
* F1 score: Token-level overlap between predictions and answers

The results showed the following:

* Phase 1:
  + Exact Match: 0.020.
  + F1 score: 0.61
* Phase 2:
  + Exact Match: 0.08
  + F1 score: 0.40

Evaluation using SQuAD metrics revealed the challenges of exact answer generation, with the enhanced model showing moderate improvements over the baseline. The attention mechanism particularly helped with longer answers and more complex questions requiring context integration. However, several limitations emerged: the vocabulary coverage proved insufficient for some domain-specific terms, the dataset contained spell errors, GLoVe could not detect words with apostrophes, and the model struggled with answering from the context.

**Future Directions**

Several promising avenues exist for improving the model's performance. More sophisticated text cleaning could preserve meaningful punctuation, while subword tokenization could improve vocabulary coverage. Processing the dataset through a spell checker could also prove useful. Furthermore, the training process could benefit from learning rate scheduling, larger embedding dimensions (e.g., 300D GloVe), and more training samples. The implementation demonstrates that while basic seq2seq approaches can establish a QA framework, modern solutions require more sophisticated architectures and training methodologies to achieve competitive performance.