# DSGA 1011: Assignment 4

**Vaibhav Rouduri**
vr2470

## Q0. 1.

Please provide a link to your github repository, which contains the code for both Part I and Part II : `https://github.com/vaibhavrouduri/NLP_A4`

## Q2. 1.

Describe your transformation of dataset.

Transformation description:

1. I created a list of stopwords which I felt covered most words which don't hold too much meaning

2. I made the whole sample lowercase

3. I used the tokenizer to tokenize the sample

4. I removed stopwords with probability 0.5

5. For all tokens in the sample, I checked if it's a number, and left it as is. I got the synonyms of each word left in the sample now, and discarded the original word if it was in the list of it's own synonyms. I then only kept the synonyms which were comprised of letters only, and made sure that the synonym was greater than 2 characters in length. This prevented a number being chosen as a synonym of a word, as I felt that would change the meaning too much.

6. I replaced a word with a random synonym from its final list of synonyms with a probability of 0.2

I felt these transformations kept the meaning intact for the most part, but could definitely also reduce the accuracy

## Q3. 1

**Report & Analysis**

- Report the accuracy values for both the original and transformed test data evaluations.
  1. Accuracy of non-augmented model on original test set: 0.92368
  2. Accuracy of non-augmented model on transformed test set: 0.87008
  3. Accuracy of augmented model on transformed test set: 0.91736
  4. Accuracy of augmented model on original test set: 0.93192
- Analyze and discuss the following: (1) Did the model's performance on the transformed test data improve after applying data augmentation?: Yes, the accuracy increased from 0.87008 to 0.91736

  (2) How did data augmentation affect the model's performance on the original test data? Did it enhance or diminish its accuracy?: Data augmentation increased the accuracy relative to the non-augmented model on the original test set. Accuracy increased from 0.92368 to 0.93192

- Offer an intuitive explanation for the observed results, considering the impact of data augmentation on model training.: I feel that data augmentation increased the accuracy on the transformed test set because the model was trained on additional examples which were transformed in the same way as the transformations done on the test set. The model's weights were better suited to deal with the transformations relative to the non-augmented set. It also had more examples to train itself on, which would generally improve the model performance.

  I also feel data augmentation increased the accuracy on the original test set simply because there more more training samples. The increase is considerably smaller than the increase on the transformed set
- Explain one limitation of the data augmentation approach used here to improve performance on out-of-distribution (OOD) test sets: One limitaton here is that we augmented data in the exact same way as the transformations done on the test set. This is very specific, and not a good way to achieve a general increase in accuracy on OOD test sets.

## Part II. Q4

| Statistics Name | Train | Dev |
|---|---|---|
| Number of examples | 4225 | 466 |
| Mean sentence length | 18.097 | 18.071 |
| Mean SQL query length | 217.373 | 211.054 |
| Vocabulary size (natural language) | 792 | 466 |
| Vocabulary size (SQL) | 556 | 396 |

Table 1: Data statistics before any pre-processing. You need to at least provide the statistics listed above, and can add new entries.

| Statistics Name | Train | Dev |
|---|---|---|
| **T5 fine-tuned model** | | |
| Mean sentence length | 12.977 | 12.994 |
| Mean SQL query length | 217.373 | 211.054 |
| Vocabulary size (natural language) | 721 | 418 |
| Vocabulary size (SQL) | 556 | 396 |

Table 2: Data statistics after pre-processing. You need to at least provide the statistics listed in Table 1 (except for the number of lines), and can add new entries.

## Q5 — T5 Model Details

| Design choice | Description |
|---|---|
| Data processing | I did not use any custom preprocessing. I fed the raw natural language queries directly to the tokenizer and did the same for SQL targets. No stopword removal, cleaning, or normalization. This matches the intended baseline. |
| Tokenization | I used the default `google-t5/t5-small` tokenizer for both encoder input and decoder output. I used `<extra_id_0>` as the BOS token for decoder inputs. No custom vocabulary or new tokens. |
| Architecture | I fine-tuned the entire T5-small encoder-decoder: embeddings, encoder layers, decoder layers, cross-attention, and LM head. No parameter freezing. |
| Hyperparameters | Learning rate = 1e-3, batch size = 8, test batch size = 8, cosine scheduler, warmup=0, epochs=5, patience=2. Optimizer: AdamW. |

Table 3: Best-performing T5 model configuration (fine-tuned).

## Q6 — Results

| System | Query EM | F1 score |
|---|---|---|
| **Dev Results** | | |
| T5 fine-tuned | 0.00 | 0.1180 |
| **Test Results** | | |
| T5 fine-tuning | 0.00 | 0.1180 |

Table 4: Development and test results.

**Quantitative Results**

**Qualitative Error Analysis**

| Error Type | Example Of Error | Error Description | Statistics |
|---|---|---|---|
| Empty SQL output | Model output: `""` | The model often generated empty sequences or stopped generation prematurely, causing SQL execution errors and empty results. | 59/59 dev queries |
| Invalid SQL grammar | Example includes tokens like `fromfrom` or `==`. | Model produced invalid SQL syntax not supported by the database engine, leading to consistent execution failures. | ∼55/59 |
| Missing WHERE clauses | GT: `SELECT ... WHERE ...`<br>Model: `SELECT ...` | Model generated partial queries missing filtering constraints, causing incorrect or empty result sets. | All non-empty queries |

Table 5: Qualitative analysis of common error types on the development set.

## Q7

Provide a link to a Google Drive containing the model checkpoint used for generating the submitted outputs. **Google Drive:** `https://github.com/vaibhavrouduri/NLP_A4`

## Extra Credit (Optional)

If attempted, describe system here and add checkpoint link.