* Scripts containing data preparation and evaluation
* Documentation describing datasets, tools and metrics you used
* A short report that describes your approach and ideas summarises the results and discusses the challenges you encountered
* Instructions on how to run and locally reproduce the results

In my approach, I tried using multiple spell-checking methods. One of them was the Python library pyspellchecker, which uses the Levenshtein Distance algorithm to find the best-suited word to replace an incorrectly spelled word in a sentence. The advantage of this method is that it does not generate nonsensical words, so the spelling is always correct. However, it does not consider the sentence's context and only selects the closest word based on Levenshtein distance. There are usually several candidate words for replacement, but the best one is not always chosen because the method ignores the sentence's context.

Another approach I tried involved using large language models (LLMs) pretrained for grammar correction. Although I tested several, the results were disappointing. The best-performing model I found was T5, fine-tuned for grammar correction on Hugging Face. T5-base is an encoder-decoder model from Google designed for text-to-text generation. Its main advantage is that it evaluates the context of the sentence to select the correct word and replace the misspelled one. However, this method sometimes replaces words with better-fitting alternatives, which can be a drawback if we only aim for correct spelling.

The biggest challenge I encountered was finding a dataset that fit my requirements. I also had difficulties identifying the best model, as some of the ones I tested proved to be ineffective.

The evaluation metrics I am using are ROUGE-1 and accuracy (percentage of sentences correctly spelled). ROUGE-1 measures the unigram (word-level) overlap between the generated text and the correct text, providing an indication of how closely the predicted output matches the target.

*Subham Sahu, Yogesh Kumar Vishwakarma, Jeevanlal kori, Jitendra Singh Thakur . Evaluating performance of different grammar checking tools, International Journal of Advanced Trends in Computer Science and Engineering, Vol. 9 No. 2 pp. 2227 – 2233, April 2020.*