DOI: 10.15439/2021F142
ISSN 2300-5963 ACSIS, Vol. 26

Applying Machine Translation Methods in the Problem of Automatic Text Correction

Wojciech Jarmosz
Adam Mickiewicz University
in Poznań
ul. Uniwersytetu Poznańskiego 4, 61-614 Poznań, Poland
Email: wojjar3@st.amu.edu.pl

***Abstract*—This document describes the problem of automatic text corrections. The author presents a classification of errors, a process of correcting texts and a proof of concept as a containerized version of machine translation system - MSuedin.**

I. Introduction

A. Groups of errors

TEXT CORRECTION is a broad topic, with many approaches leading to slightly different results. According to the classification introduced in Naber[1], we can divide errors into four main categories:

- **spelling errors** - words that don't belong to the language. Analysis doesn't require knowledge about the context.
- **grammatical errors** - known also as real-word errors, which break the grammatical rules of the language. Very often, correction of this type of error requires analysis of context.
- **stylistic errors** - they don't break grammatical or morphological rules of the language but include repetitions, colloquialisms, and overcomplicated structures.
- **semantic errors** - information not related to facts and knowledge about the world. Such errors are very difficult to detect and correct. This type of error demands building some sort of knowledge model.

B. Text correction process

Building error correction systems requires a process. According to Kukich[2], it can include:

- error detection,
- generating candidates to correction,
- rating candidates,
- choosing candidate with the highest score.

C. Approaches to automatic text correction

Over the years, approaches to creating text correction systems have evolved. Rule-based architecture, used for example in MS Word, is expensive, complex, and requires linguists involved in defining such rules. On the other hand, we have statistical approaches and classification methods, used more commonly in contemporary tools. Each of these approaches is good in some specific areas, such as selecting the best candidate from the confusion set, correcting prepositions or articles. Nowadays, the most effective and widely used method is based on machine-translation techniques, including the usage of the transformers models. The author of this article focuses on the implementation of MarianNMT[3] transformer-based system - MSuedin[4], and prepares a docker image for it, allowing the user to easily run such a program with different parameters, on multiple GPUs without worrying about system dependencies and framework integrations.

II. Proof of concept

Author prepared a docker image for the MSuedin system using the docker-nvidia, CUDA toolkit, virtualenv and python3. The container was built and run on Linux Ubuntu 20.04 with installed CUDA Toolkit 11.3 and 19GB GPU. Containerization makes the program independent from the host operating system and other dependencies. The technique is called "write once, run anywhere". It helps our grammar correction system to scale better and start up faster. Instead of working on system configuration and program installation, machine learning engineers can now focus on providing high-quality training data.

III. Potential usage

The project can be used to support the translator's job by reducing time spent on correcting errors manually (it can takes many hours) and form a good basis to create automatic text correction systems with any language corpus due to virtualization and containerization.

IV. Research status

MSuedin[5] - one of the winning GEC systems in the BEA 2019 shared task is an example of a transformer-based machine translation system that corrects English grammatical errors. On the other hand, there are no such systems for niche languages like Polish, German or Finnish, which will provide satisfactory results in correcting text errors.

V. Research goals

The main goal of the research is to prepare a containerized version of the machine translation system for text error correction, which can be run on multiple graphic cards. Another goal is to train models on the language corpora less popular than English.

VI. Future experiments

The author will build a corpus, containing pairs of sentences with and without the error. To increase the number of samples, there may be also a need to generating synthetic data using some sort of thesaurus or confusion sets. After that, the author will divide the dataset into a train and test set in a proportion of 8:2. Using available tools in the MarianNMT framework, the model will be trained (there will be a need for multiple GPU usage). The model will be loaded in a containerized version of MSuedin and evaluated using F-score and accuracy metrics.

VII. Conclusion

A successful experiment includes preparing a model and running it in the containerized version of MSuedin. Error correction of the text is a very interesting topic, which has broad and measurable usage in real-world scenarios. I believe that time-consuming tasks such as grammar error correction can be successfully automated.

References

[1] D. Naber, A rule-based style and grammar checker, GRIN Verlag 2003.

[2] K. Kukich, Techniques for automatically correcting words in text, ACM Computing Surveys 1992

[3] https://marian-nmt.github.io

[4] https://github.com/grammatical/pretraining-bea2019

[5] R. Grundkiewicz, M. Junczys-Dowmunt, K. Heafield, Neural Grammatical Error Correction Systems with Unsupervised Pre-training on Synthetic Data, BEA 2019