**English Grammar Correction Application**

****

University of Missouri-Kansas City

**Spring 2020**

**Date:**Jan 31 - 2020

**Course:**COMP-SCI 5560

**Class:**Knowledge Discovery and Management

**Team Members:**

1. Abbas Alshaikh
2. Andrew Wilson
3. Srichakradhar Nagireddy

**English Grammar Correction App**

**Possible Names**

Our application’s name is not important at this stage and will be determined later, but these are some suggestions: *Spell Checker, English Corrector, Auto Grammar Corrector, Auto English Corrector*.

**Introduction**

The grammar checker is a program or website that helps you discover spelling and typographical errors and corrects them or recommends modifying them. The program automatically detects errors without the need for direct human intervention.

**Motivation**

With pressure at work, multiple tasks, and difficulties in life, people easily become distracted. As a result, typos and spelling errors occur frequently. Employees, students, and journalists need to be precise in their written work. Sometimes, typographical errors can be deadly or costly.

**Significance of Project**

This application simplifies the process of checking English grammar and English spelling by checking for and correcting errors of the entered text. Users will be able to upload text files or by typing directly. The program is supposed to offer recommendations for some words, typos, and others.

**Features**

* Simplicity of use.
* No user registration required.
* Automatic checking.
* Display recommendations.
* Ignoring option for some labels from correction.
* Support different types of files; (doc, docx, odt, pdf, rtf, tex, txt , xls).
* Save user history.
* Export the results as a file.
* Simply, it's free app.

**Stretch Goals (Processing steps)**

* Entering texts (Typing or Uploading text file).
* Detecting languages (Whether supports language or not).
* Creating and using glossaries (The hardest part - word processing).
* Product names.
* Ambiguous words.
* Borrowed words.
* Making batch requests (Divide batches for processing "query").
* Long-running operations (Operation status "cancelling ").
* Reporting usage with labels (Adding user-defined labels).
* Return results (Show the final edited text to the user).

**Possible API’s**

 WebSpellChecker API – <https://rapidapi.com/webspellchecker/api/webspellchecker>

JSpell Checker API - <https://rapidapi.com/jspell/api/jspell-checker>

LanguageTool API - <https://rapidapi.com/dnaber/api/languagetool>

**Datasets**

The [CoNLL-2014](https://www.comp.nus.edu.sg/~nlp/conll14st/conll14st-test-data.tar.gz) dataset contains over one thousand English sentences, with information on their correct or incorrect grammar.

**Possible Tech**

Here is a list of some of the possible techniques that can be used to implement the project;  (Python, MEAN Stack, Adobe Photoshop, MySQL, Flutter, IntelliJ, PyCharm, GitHub).

**References**

1. <https://rapidapi.com/collection/grammar-spellcheck-api>

2. <https://cloud.google.com/translate/docs/how-to>

3. <https://apigility.org/documentation/recipes/upload-files-to-api>

4. <https://developers.google.com/drive/api/v3/manage-uploads>

5. <https://towardsdatascience.com/5-heroic-tools-for-natural-language-processing-7f3c1f8fc9f0>

6. <https://stanfordnlp.github.io/CoreNLP/index.html>

7. <https://umkc.box.com/s/eoc0h80vn5o6szz7jbfr9k5pmywv0wmb>

       If there is any reference link will by provide later in the final report.