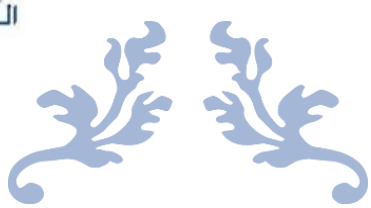

NLP Project Report

Name	ID
Zulfiqar	168459
Nourhan	177503
Faris	170916

Department of Computer Science

Shorouk city, The British university in Egypt

June 11, 2021

1-Introduction:

Indeed, into our new day to day lifestyle, we use our computers to send business reports to managers or send a text message to a friend using social media platforms as a fast effective and trendy way of communication with other people worldwide. However, most people do contextual mistakes in the process of texting. According to [7], most spelling errors has 2 types, first, real word errors which represent errors in spelling that has a real meaning in normal language dictionaries like Oxford dictionary. for example, the user could send a message saying “hello, my friend I am a little silk” instead of “I am a little sick”. The second type of error is non real word errors which are errors that has no meaning into traditional dictionaries. For example, the user could say “I am a little sek” which has no meaning in traditional dictionaries. According to [8], there are 3 different types of the non real word errors. First, mistyping error which means the user made a mistake while texting using keyboard or touching wrong letters without paying enough attention such as typing “mecessary” instead of “necessary”. Second, cognitive errors, which means that the user maybe be ignorant about how to write the word properly. Third, phonetic substitution, which can be easily represented by the coming example as the user might mistakenly write “Parashoote” instead of “Parachute”. Furthermore, According to [9], 70% - 90% of all misspelling mistakes are single errors misspellings which simply means that only one letter is missed or deleted by mistake for example “aple” instead of “apple”, or the insertion of a mistaken letter to the word like “taake” instead of “take”, or the substitution of a liter by another like “toor” instead of “tour”, or transportation mistake by writing “baout ” instead of “about”. For sure AI techniques such as Natural language processing could represent a real solution by carefully analyzed all of the previously mentioned types of errors. Putting in consideration, that it’s a necessity for an NLP model to carefully classify or differentiate between no word errors and slang language, domain specific words or special jargons. For this purpose, this report is concerned about using Natural language processing technics to autocorrect language mistakes on various platforms, websites and even search engines such as social media platforms like Facebook, Souq.com, and Google. In fact, spell correction is one of the fields that is highly enriched with lots of research funded by data leaders and leading businesses all over the world. Technically, a spell corrector

could be represented in 3 different models. According to [8], first model is error detector which is going to check for mistakes in context considering different acronyms, domain terms, special jargons, etc. Secondly, it's spelling suggester model to provide different suitable ways to correct misspelling in sentences. The third model is error corrector which is simply going to check all proposed ways of correction by second model and choose best correction for the context. For that reason, it was a necessity to avoid traditional dictionaries and focus onto more deeper datasets to feed our 3 different models and get least false positive and true negative. Indeed, our focus will be directed towards using best advanced techniques to train our models and to reach the highest accuracy.

2-Literature review:

According to Singh, S. P., Kumar, A., Singh, L., Bhargava, M., Goyal, K., & Sharma, B [1], Mistakes in any language can be divided into two kinds: spelling mistakes and grammatical mistakes. This paper aims to achieve a goal of minimizing these errors by creating a frequency-based spell checker and a rule-based grammar checker for English language.

Spell Checking

"One of the error types that can be done while writing spellings is non word error: these are the errors where the word doesn't exist in the dictionary " [1]. Example on that: He is a gud boy, she is a nce girl, gud and nce are not existed in English dictionary, the correct words are good and nice. Therefore, spell checker basic tasks are: Error Detection, and Suggestion Prediction.

1. Error detection techniques
   there are many methods for error detection such as:

   1.1 Dictionary Lookup Method
   It is based on arranged rows and columns data file, if the word is not exist, so it is considered in correct.
   It is easy to implement, on the other hand, it is not easy to extract and save all vocabulary of a specific language, in addition, the search has large cost.

   1.2 N-Gram Technique
   "An n-gram is a collection of characters of length N; If N is equal to 1 then the term used is a unigram, if N is 2 then it is called Bigram, and if N is 3 then the term is trigram and so on" [1].
   "Each word or string that is involved in the process of comparison is divided into pair of adjacent N-grams, in addition, the n-grams algorithm is also referred as neutral string matching or a language independent algorithm" [1].

2. Error Correction and Suggestion Prediction Techniques:

   2.1 Edit Distance:
   "This method is based on the assumption that a person usually types only a few incorrect letters while typing a word, therefore for each dictionary word, lesser the number of the basic edit

operations (insertion, deletions, substitutions) necessary to convert the dictionary word into the non-word, higher the probability that the user intended to type that dictionary word in place of the misspelled word" [1].
Ex: the distance between good and gud will be 2

G	G	Same
O	U	Substi
O	-	Deleti
D	D	Same

2.2 Grammar Checking [Rule based approach]
"This is the approach where we match the text with a set of rules and that has been at least POS tagged, in addition, the rules are in accordance with the grammar of the language of interest" [1].
This method has many advantages such as:

- Gives immediate feedback.
- easy to configure and understand by users.
- it can explain grammar rules and give detailed error messages with comments.
- Its extension is easy, starting with just one rule and then extending it rule by rule.

The proposed system was the following:

1- Spell checker based on Frequency Based Suggestion Prediction that applies the following steps
   1.1 Text Retrieval: check user words against the dictionary efficiently, and remove tags in the cleaning part.
   1.2 Dictionary Lookup: if the word is correct, increase its frequency by 1
   1.3 Edit Distance: if the word is not correct, using query optimization techniques to find the closest words to this word.
   1.4 Frequency Based Suggestion Prediction: show the user suggestions of minimum word distance and maximum frequency word

2- Grammar Checker
   2.1 Based on tense rules
   2.2 POS tagging
   Example: "Those people is eating sweets
   POS Tag-Those/DT People/NNS is/VBZ eating/VBG Sweets/NNS" [1].
   Tag the incoming sentence using POS tagger
   2.3 Parsing
   "it is the process of analyzing a string or symbols, conforming to the rules of grammar, then we get a tree which is known as parsed tree which will divide noun phrase and verb phrase." [1].
   2.4 Match with the Tense Rules
3- Find suggestions

Using the same approaches, another methodology was proposed for text paragraphs written in punjabi language using hybrid approach according to Kaur, H., & Kaur, N [2], using the following steps:
"

Step I: Input the source string

Step II: Tokenize the input of first step into words

Step III For each Token compare it with the Dictionary

Step IV Check whether it is correct or not. If it is correct, then go to Step III, otherwise apply Rule Bases Approach

Step V Again find the word from dictionary

If word is found go to Step III, otherwise apply Edit Distance Approach

Step VI Find the minimum distance from this Token to the word in the Dictionary

Step VII Sort these words in ascending order of their distance

Step VIII Check the words obtained with same distance by comparing previous and next word of the target word to obtain best possible suggestion" [2].

According to Tolentino, H. D., Matters, M. D., Walop, W., Law, B., Tong, W., Liu, F., … Payne, D. C [3], The goal of this paper was to create a UMLS-based spelling error correction tool as a first step in the NLP pipeline for adverse events following immunization reports.

Spelling checking steps:

1- error detection
determining whether the word in the dictionary dataset or not (smaller first then and bigger one)
2- word list generation

- POS tagging (helped carry out POS disambiguation in the next step)
- Extracting the word list from the custom dictionary using these word-list generation algorithms:
  - Metaphone (search for similar sounding words).
  - Header (looks for words with the same first 4 characters)
  - N-gram (search for words containing the next 4 characters after the first one)
  - Transposition (searched for words where any 2 characters are switched)
  - Deletion (search for word matches by sequentially inserting a wildcard character in the misspelled word to simulate a character deletion)
  - Insertion (search for word matches by sequentially deleting a character in the misspelled word to simulate a character insertion)
  - Substitution (searched for word matches by simulating character substitution)

"The last four algorithms are based on Damerau's findings that 80% of all misspelled words contained a single instance of one of the four error types (transposition, deletion, insertion and substitution), also known as a class of single error misspellings" [3].

3- word list disambiguation

- "the objective of this step was to rank the candidate words by determining which word from the

word list in step 2 had the lowest Levenshtein score, which is the number of edits needed to transform a misspelled word to any of its possible corrections" [3].

- This lowest Levenshtein score candidate word then considered a correction word.
- Levenshtein is a built-in function to calculate the number of possible corrections by simulating the four types of errors (insertion, deletion, transposition and substitution)
  If 2 or more words have the same score that means a deadlock or tie situation.
  The solution is to use a smoothing algorithm such as: "
  - UMLS concept algorithm
  - metaphone algorithm (assumes words that sound the same have the greater propensity to be misspelled)
  - homonym algorithm (if the misspelled word had the same first letter and metaphone as the candidate word).
  - N-gram algorithm (if the misspelled word contains similar character subsets as the candidate word, i.e., "wretch" and "retch")
  - length algorithm (assuming the clinician did not intend to misspell with a longer term)
  - POS algorithm (if the candidate word has a similar part-of speech tag as the misspelled word)
  - history algorithm (if the candidate word already existed with a probability distribution in our NLP database)

  The result is a final set of ranked words from which the lowest Levenshtein score was used to select the correction" [3]

4- error correction
   based on the disambiguating and selecting the most probable corrected word, replace it.

In conclusion, performance measurements were sensitivity, specificity, and positive predicted value by comparing the number of words that both the spell checker and the human observers, according to this paper, they found that, "During training, the spell checker had a sensitivity of 93%, a specificity of 100%, and a PPV of 64%, on the other hand, the test data set had a sensitivity of 74% , a specificity of 100%, and a PPV of 47%" [3].

According to X. Li, H. Liu, and L. Huang [4], they developed a stand-alone spelling correction model which detects and corrects the spelling of each token on its own as a sequence labeling task. This is done through utilizing spelling information and global context representations. This model is trained on a dataset made from the 1-Billion-Word-Language-Model-Benchmark (Chelba

*et al.*in [4]), the dataset made by generating noisy inputs for the data retrieved from the benchmark model, then the dataset is split into training and testing sub-datasets with 80% for training and 20% for testing. The architecture of the model is composed of 2 sub-models based on transformer-encoder architecture that tackles different parts of the problem. Firstly, word+char encoder which is used to extract the global context information (word) and encoding the spelling information (character). Then comes the subword encoder, this phase uses sub-word tokenization to address the spelling and context information by passing the noisy subword sequence into the subword transformer encoder to predict the correct word token. The model achieved $F_{0.5} = 0.959$ which is considered state-of-the-art performance compared with other models [4].

According to S. M. Jayanthi, D. Pruthi, and G. Neubig [5], they built an open-source toolkit that tackles spelling correction in the English language. This toolkit contains multiple models which solve the problem of spelling correction. All the models are trained using spelling errors in context and using contextual representations. Some of the models used in the toolkit are: SC-LSTM (Sakaguchi *et al.* in [5]) which corrects the misspelt words by feeding bi-LSTM with semi-character representation, CHAR-LSTM-LSTM (Li *et al.* in [5]) which passes the characters to 2 different bi-LSTMs in order to build word representation and predict the correction, CHAR-CNN-LSTM (Kim *et al.* in [5]) which works in the same way as the previous model but instead of building the word representations with LSTM it is built with CNN, and BERT (Devlin *et al.* in [5]) which uses pre-trained transformers network to obtain the word representations then feeding it to a classifier to predict the correction. All the models were trained on 1.6M sentences retrieved from the 1-Billion-Word-Language-Model-Benchmark (Chelba *et al.*in [4]) while using different noising methods to noise 20% of the corpus. After testing all the models, it is noted that the BERT model is the best and the most stable among all the noising methods and all the test datasets used in the testing phase (~96%) [5].

Based on M. Flor, M. Fried, and A. Rozovskaya [6], addressed the avalibalilty of the data by building an annotated dataset of 6121 spelling errors retrieved from a corpus of essays written by English language learners (TOEFL-Spell Corpus). Then, developed a context-aware spelling correction model which contains 3

processing phases: error detection and it is done using the lexicon dictionary. Candidate generation which are generated using the same dictionary used in the last phase and a candidate is all dictionary words within a specific distance and some constraints specified in the model. Finally, ranking of the candidate corrections which is done through multiple methods like orthographic similarity and phonetic similarity. Also, there is a lot of contextual features involved in choosing the best candidate like N-gram support, Dejavu, Word Embeddings. The model achieved state-of-the-art result which is accuracy= 88.12%. and F1 score equal 99% in both error detection and candidate generation [6].

According to [10], there are different approaches used during the history for spelling checking. First at 1918, soundex algorithm was used to detect misspelled words as the core of this algorithm is to code words with similar homophones or pronunciation with the same index. Certainly, there were rules for this indexing process. First, the code starts with words' first letter then remaining letters will be converted to a specific number from (0-3). So, at the end, the result will be 1 letter and 3 coded numbers. For example, Soundex("Nice") =N121. As the following letters are coded like "i,e=1, c=2".

Recently, this model is considered to be primitive because of the new models and advances for better algorithms like Bayesian noisy channel model. According to [11], the core of the previously mentioned algorithm is to consider all misspelled words as a noisy signal which deviated from normal signals and if we can know how this signal is deviated then it's very easy to guess the correct words to replace wrong one as shown in figure (1).

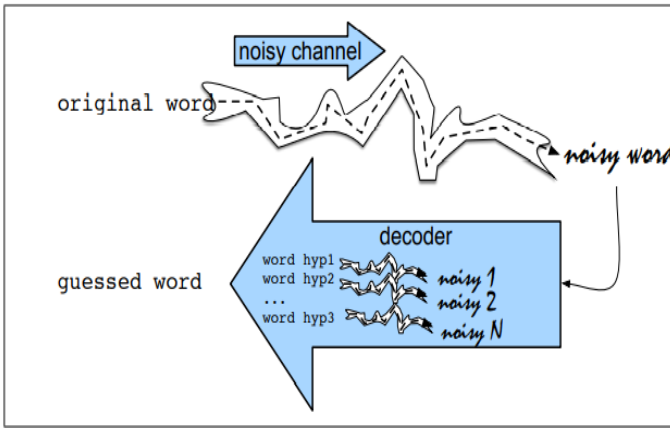

Figure 1: shows the intuition behind "Bayesian noisy channel model".

However, the core of Bayesian is probabilistic based model which aims to predict the word w that led to the current misspelled word based onto 2 statistical factors to be computed P(w) and the likelihood probability P(O|w) as shown in figure (2). Paying attention that P(O|w) should be the highest as possible. we want the word with maximum P(O|w) as denoted by

argmax () function out of the whole corpus or all possible corrections list words. As a matter of fact, this model showed good results into auto correcting but not in all cases. For that reason, there were the question again to have a better probabilistic to be able to predict w more accurately.

Figure 2: shows the probabilistic calculations for “Bayesian noisy channel model”.

$$\hat{w} = \underset{w \in V}{\operatorname{argmax}} P(x|w)\, P(w)$$

According to [12], N-gram model developed by the Russian mathematician makrov is the new probabilistic model to be used to predict w. In contrast with Bayesian model, this time N-gram model is going to predict w based on neighboring words or context. As N-gram is simply representing number of words to be used as neighboring words as shown in figure (3). specifically, 2-gram model is used to avoid in some cases complications.

Figure 3: shows the probabilistic calculations for “N-gram model”.

$$P(w_1^n) \approx \prod_{k=1}^{n} P(w_k|w_{k-1})$$

3-Dataset description:

The birkbeck file contains 36,133 misspellings of 6,136 words. It is a blend of errors taken from the native speakers (British or American) of the brikbeck spelling errors corpus, a set of files of spelling errors and mistakes assembled from different sources which are available from Oxford text archive with their detailed description. It incorporates the results of spelling tests and errors from free writing, taken for the most part from schoolchildren, university students, or grown-up proficiency understudies. The majority of them were originally handwritten. In this dataset, each correct word is gone before by a dollar sign and followed by its incorrect spellings, each on the same line without duplicates. Correct spellings are given in Oxford English structure. Where the misspellings were taken from American authors, endeavors at explicitly American structures.

References:

[1] Singh, S. P., Kumar, A., Singh, L., Bhargava, M., Goyal, K., & Sharma, B. (2016). Frequency based spell checking and rule based grammar checking. *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, 4435–4439. IEEE.

[2] Kaur, H., & Kaur, N. (2016). *Spell checking and error correcting system for text paragraphs written in Punjabi language using hybrid approach*. doi:10.5281/zenodo.46488

[3] Tolentino, H. D., Matters, M. D., Walop, W., Law, B., Tong, W., Liu, F., … Payne, D. C. (2007). A UMLS-based spell checker for natural language processing in vaccine safety. *BMC Medical Informatics and Decision Making*, *7*(1), 3.

[4] X. Li, H. Liu, and L. Huang, "Context-aware stand-alone neural spelling correction," *arXiv [cs.CL]*, 2020.

[5] S. M. Jayanthi, D. Pruthi, and G. Neubig, "NeuSpell: A Neural Spelling Correction Toolkit," arXiv [cs.CL], 2020.

[6] M. Flor, M. Fried, and A. Rozovskaya, "A benchmark corpus of English misspellings and a minimally-supervised model for spelling correction," in Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications, 2019.

[7] F. Ahmed, E. W. De Luca, and A. Nürnberger, "Revised N-Gram based Automatic Spelling Correction Tool to Improve Retrieval Effectiveness," *Polibits*, vol. 40, pp. 39–48, Dec. 2009, doi: 10.17562/PB-40-6.

[8] Y. Bassil, "Parallel Spell-Checking Algorithm Based on Yahoo! N-Grams Dataset," vol. 3, no. 1, p. 8, 2012.

[9] K. Kukich, "Techniques for automatically correcting words in text," *ACM Comput. Surv.*, vol. 24, no. 4, pp. 377–439, Dec. 1992, doi: 10.1145/146370.146380.

[10] “Soundex,” *FamilySearch Wiki*. https://www.familysearch.org/wiki/en/Soundex (accessed May 01, 2021).

[11]- B.pdf (stanford.edu)

[12] P. Gupta, “A Context-Sensitive Real-Time Spell Checker with Language Adaptability,” in *2020 IEEE 14th International Conference on Semantic Computing (ICSC)*, San Diego, CA, USA, Feb. 2020, pp. 116–122, doi: 10.1109/ICSC.2020.00023.

Dataset Link: https://www.dcs.bbk.ac.uk/~ROGER/missp.dat.