MIDDLE EAST TECHNICAL UNIVERSITY

CENG 463

HW #1

Onur Yılmaz

1627868

Implementation Idea

In this spell checker, the main idea is based on taking a word and creating possible variations of the word by deleting, inserting, transposing or replacing letters. After creating variations, their existence are checked at Brown corpus and their occurrences are weighted according to their distance to the original word. With this idea, if the original word is found on corpus without any modification, its probability is increased. On the other hand, as the distance from the original word increases, probability of seeing modified word is decreased.

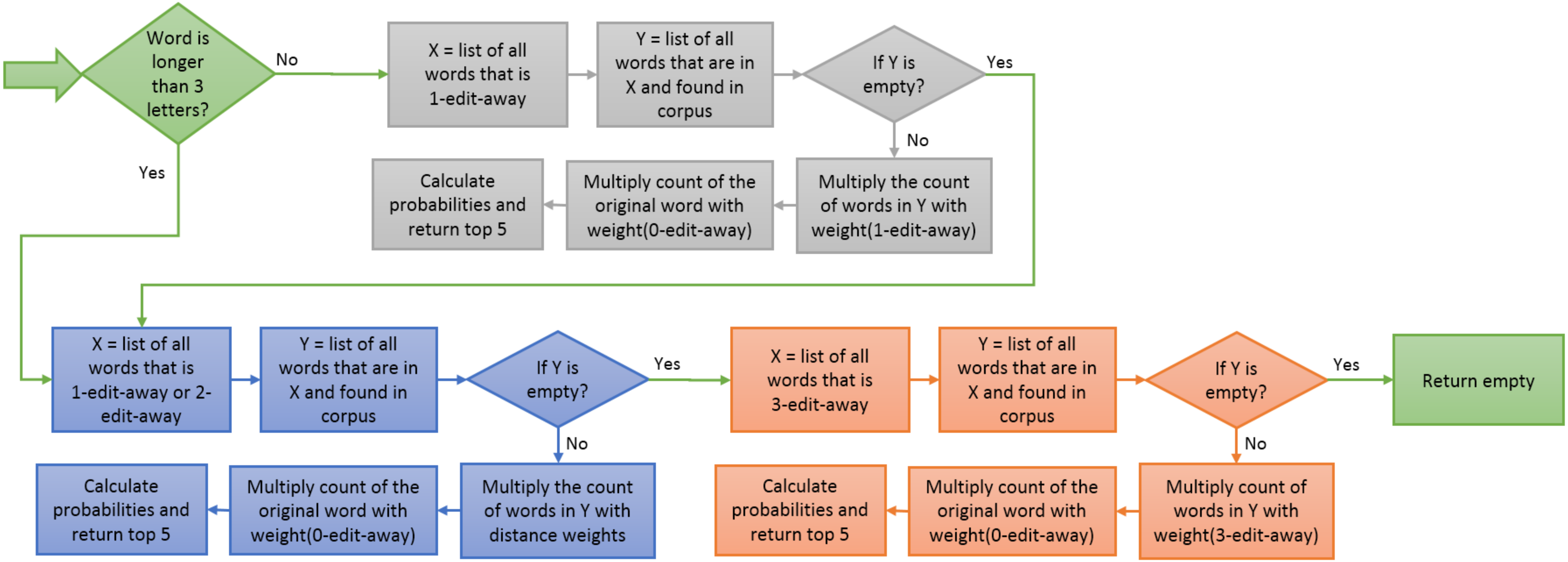
While increasing and decreasing probabilities, numbers taken from [1] is used. According to this paper where typing experiments are undertaken, probability of 1-edit-away is calculated as 94% and 2-edit-away is 5% and 3-edit-away is 1%. Using these percentages, counts of the created words are multiplied as following:

Distance	Count Multiplier
1-edit-away	95
2-edit-away	4
3-edit-away	1
Original	100

In addition, it is observed that short words can easily be converted to other words in 2 edits and those created words can be very popular in corpus. In such situations, very different words are suggested for correction. In order to eliminate this, when number of letters is less than 4, only 1-edit distance is considered.

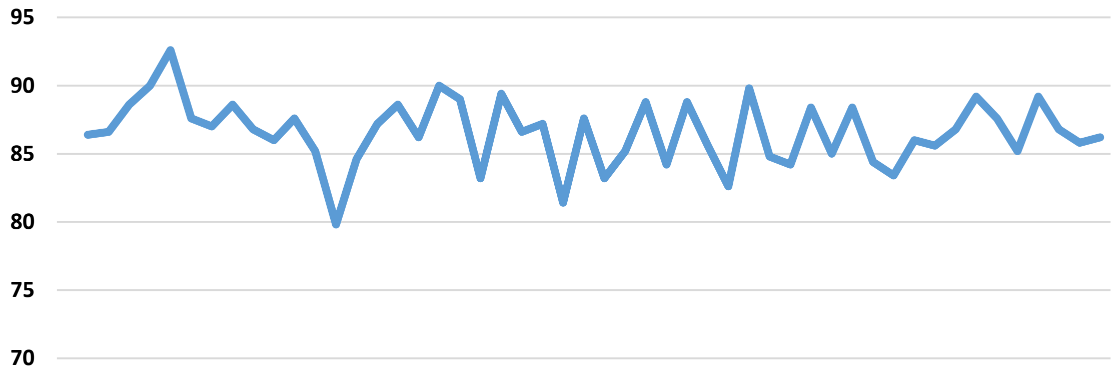
Flow of operations is shown as a diagram in the following page.

[1]: http://www.sciencedirect.com/science/article/pii/S016792360200115X


Word is longer than 3 letters?
No
Yes
X = list of all words that is 1-edit-away
Y = list of all words that are in X and found in corpus
If Y is empty?
Yes
No
Multiply the count of words in Y with weight(1-edit-away)
Multiply count of the original word with weight(0-edit-away)
Calculate probabilities and return top 5
X = list of all words that is 1-edit-away or 2-edit-away
Y = list of all words that are in X and found in corpus
If Y is empty?
Yes
No
Multiply the count of words in Y with distance weights
Multiply count of the original word with weight(0-edit-away)
Calculate probabilities and return top 5
X = list of all words that is 3-edit-away
Y = list of all words that are in X and found in corpus
If Y is empty?
Yes
No
Return empty
Multiply count of words in Y with weight(3-edit-away)
Multiply count of the original word with weight(0-edit-away)
Calculate probabilities and return top 5

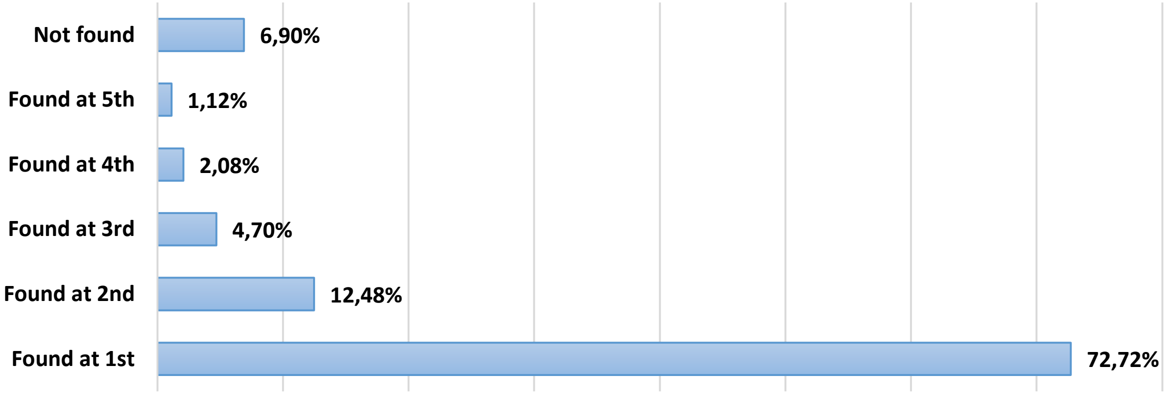
Analysis of Implementation

The method is quantitatively analyzed according to the criteria given in homework text by a test file written for this purpose. When 1 point is given to the 1st suggestion, 0.8 to the 2nd and so on, the following statistics are found for a series of trials:



Mean	86,58		Min	79,8
Std. Dev.	2,40		Max	92,6

Moreover, positions of the correct suggestions are analyzed as following:



As can be seen from above figures, on the average model returns with 86 points out of 100 with a small standard deviation. In addition, at 70% of the time the model suggests the correct word at 1st location; on the other hand nearly at 7% the model fails to correct given word.

Possible Problems and Improvements

The main problem related to this model is that if a very popular word is created by manipulating the original word, it is returned as correction. For some cases, this suggestion is not acceptable. For instance,

Original:	Misspelled:	Corrections:
trunk	truk	[('true', 0.7021), ('truck', 0.1804), ('took', 0.0578), ('turn', 0.0308), **('trunk', 0.029)**]
blood	blook	[('look', 0.492), ('book', 0.2386), **('blood', 0.155)**, ('block', 0.0903), ('took', 0.0242)]

In order to solve this problem, instead of one word, bigrams can be used so that a more meaningful suggestion can be provided.

Secondly, this implementation does not consider the role of the word in context. If there would be an improvement that considers the POS tag of the word, it could yield better solutions. For instance, where "take" is a verb, "the" is suggested by model. Likewise, where "have" is a verb, a pronoun "he" is suggested as correction.

Original:	Misspelled:	Corrections:
take	taek	[('the', 0.7412), **('take', 0.1622)**, ('talk', 0.0427), ('they', 0.0328), ('them', 0.0211)]
have	hve	[('he', 0.6272), **('have', 0.3718)**, ('eve', 0.0006), ('hive', 0.0003), ('hue', 0.0002)]

In order to solve this problem, again not only the one word but also other words or the complete sentence is required.

The third problem of this model is the computational problems when 3-edit-away words are searched especially for long words. In some cases, computer can even freeze while listing 3-edit-away words. To solve this problem, number of words to create by 3 edits can be limited.

Considering the overall success level of 86 points, and 70% of correct suggestion at 1st location, when the possible problems are solved, this model can be used more effectively.