COMP90049 Knowledge Technologies - Project 1 Report

Anonymous

1. Introduction

The purpose of this report is to evaluate the performance of approximate matching methods. We have used the Levenshtein Distance and N-Gram Distance approximate matching methods on the problem of spelling correction. These methods are applied to find the correct spelling for a misspelled word and then their efficiency is evaluated using Evaluation Metrics which include the measurable: Precision and Recall.

2. Dataset

The data used (Saphra & Lopez, 2016) in this project consists of three text files: misspell.txt, correct.txt & dictionary.txt.

2.1. misspell.txt

It consists of a list of 716 words, one per line, which have been identified as misspelled.

2.2. correct.txt

It consists of a list of 716 words, one per line, which are the correct spellings for each word in the misspell.txt file.

2.3. dictionary.txt

It consists of a list of about 400,000 English language words, which have been compiled from different sources.

3. Evaluation Metrics

Throughout this report, the following terms will be used to evaluate each algorithm:

Precision: For systems that predict one or more word(s), it is the fraction of correct predictions over the attempted predictions.

The computing formula is:

$$Precision = \frac{The\ number\ of\ correct\ predictions}{The\ total\ number\ of\ attempted\ predictions}$$

Recall: For systems that predicts one or more word(s), it is the proportion of words which have a correct response.

The computing formula is:

$$Recall = \frac{The\ number\ of\ tokens\ with\ a\ correct\ prediction}{The\ total\ number\ of\ tokens}$$

4. Implementation

Both Levenshtein Distance and N-Gram Distance methods have been implemented in Java programming language. Please check the README.txt file to gain further information.

5. Levenshtein Distance

Levenshtein Distance, which is also referred to as Edit Distance is used to find the similarity between two words. This is done by calculating the number of operations that need to be completed to make two words equal to each other. These operations consist of substitution, deletion or insertion of an alphabet in the word (Manges, 2005).

5.1. Working

The system finds the Edit Distances between each word in the misspell.txt file with each word in dictionary.txt file (i.e. it performs 716*400,000 = 286,400,000 calculations).

Since, this is a multiple-predictions system, to compute Precision and Recall, the system keeps the results of words where the distance is less than 2 as attempted predictions and then compares each attempted prediction with the corresponding word in the correct.txt file to ascertain the number of correct predictions.

After observation, the words with Edit Distances below 2 were considered to be attempted predictions because these were the words that were close to the misspelled word and the probability of finding the correct spelling out of these was high. If the distance had been 4 or 5, then number of attempted responses would increase greatly. Thereby, reducing the precision of the algorithm.

5.2. Result

The results of Edit Distance are as follows:

Evaluation Metrics	Values (%)
Precision	4.38
Recall	40.64

Table 1: Result of Edit Distance

Correct predictions	Attempted predictions
291	6642

Table 2: Attempted & Correct predictions

5.3. Analysis

Using the Edit Distance method on the data provided, it resulted in a Precision of 4.38% and a Recall of 40.64% (Table 1). This means that the algorithm was able to find the correct spellings for few words (291, as shown in Table 2) out of a total of 716 misspelled words. The reason for this being, the method is able to find the correct spellings for only those misspelled words where the Edit Distance is below 2 i.e. for only those words which are really close to the misspelled word and the spelling is slightly incorrect. For example, the misspelled word, 'aeroplane' where the correct word is 'airplane', the system was not able to detect the correct spelling and it detected the words such as, 'aerophane', 'aeroplane', 'aeroplaner' and 'aeroplanes', which are all close to 'aeroplane' but are not correct.

So, if the maximum allowed Edit Distance is increased from 2 to 3, the Recall would increase significantly but the Precision would reduce as the number of predictions increase. This can be illustrated with the help of the following tables:

Results with Edit distances less than 2:

Misspelled word	Correct word	Attempted predictions	Right/ Wrong?
accually	actually	actually	✓
aeroplane	airplane	aerophane	✗
		aeroplane	✗
		aeroplaner	✗
		aeroplanes	✗
backwords	backward	backswords	✗
		backwards	✗
		backwoods	✗
		backword	✗

Table 3: Example of Edit Distance

Precision = (1/9) * 100 = 11.1%
Recall = (1/3) * 100 = 33.33%

Results with Edit distances less than 3:

Misspelled word	Correct word	Attempted predictions	Right/ Wrong?
accually	actually	accusably	✗
		actually	✓
		annually	✗
		factually	✗
		tactually	✗
aeroplane	airplane	aerophane	✗
		aerophone	✗
		aeroplane	✗
		aeroplaner	✗
		aeroplanes	✗
		aerovane	✗
		airplane	✓
		gyroplane	✗
		seriplane	✗
backwords	backward	backboards	✗
		backloads	✗
		backports	✗
		backsword	✗
		backswords	✗
		backward	✓
		backwards	✗
		backwinds	✗
		backwood	✗
		backwoods	✗
		backwoodsy	✗
		backword	✗
		backworm	✗
		backwort	✗
		backyards	✗
		hackworks	✗
		rackworks	✗

Table 4: Example of Edit Distance

Precision = (3/31) * 100 = 9.67%
Recall = (3/3) * 100 = 100%

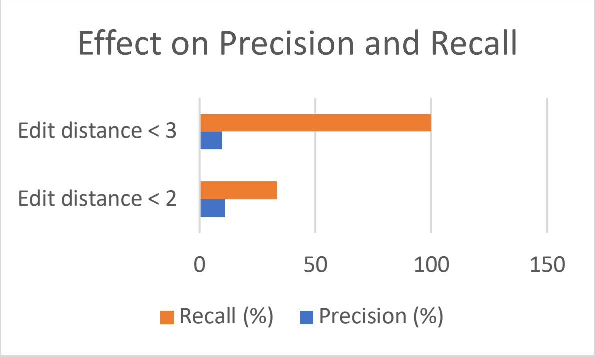

Figure 1: Effect on Precision & Recall

As shown in Figure 1, the increase in the maximum allowed Edit Distance has resulted in increasing the attempted predictions from 9 to 31. Thereby, reducing Precision from 11.1% to 9.67% but increasing Recall from 33.33% to 100%.

6. N-Gram Distance

N-Gram Distance works on the same principle as Edit distance i.e. to find the best match for a misspelled word, but here the method breaks a word into substrings of length 'n' to make comparisons between them (Nicholson, Zobel, Verspoor & Kotagiri, 2018).

6.1. Working

Using N-Gram Distance method, where N = 2, the system computes the distances of each word in the misspell.txt file with each word in the dictionary.txt file (i.e. performing 716*400,000 = 286,400,000 calculations).

Since this is a multiple-predictions system, to compute Precision and Recall, the system chooses the distances where the distances are greater than 0.5 and considers them as attempted predictions for the misspelled words, it then compares each attempted prediction with the corresponding word in the correct.txt file to find if the system found the correct prediction or not.

After implementing the method, the words with distances less than 0.5 were discarded, as these were the words which were not at all similar or close to the correct spellings of misspelled words and the method would not have been able to find the correct prediction out of these. When the distance was increased, we found words that were really close to the misspelled word. Therefore, increasing our chances of finding the correct spelling of the misspelled word.

6.2. Results

The results of N-Gram Distance (where N = 2) is as follows:

Evaluation Metrics	Values
Precision	3.09%
Recall	30.73%

Table 5: Results of N-Gram

Correct predictions	Attempted Predictions
220	7124

Table 6: Attempted & correct Predictions

6.3. Analysis

The use of N-gram Distance method on the dataset has resulted in a Precision of 3.09% and a Recall of 30.73% (Table 5). This means that the algorithm is not able to find correct spellings for a lot of misspelled words (As shown in Table 6, 220 out of 716).

Precision and Recall in N-Gram Distance depend a lot on the value of 'N' i.e. the type of N-Gram Distance that is being performed on the given strings, it can be a 2-Gram, 3-Gram, 4-Gram and so on. If we keep on increasing the value 'N', we will be making it difficult for the algorithm to find a match between two words. This can be understood with the help of an example, for the words 'crat' and 'cart'. The 2-grams are '#c, cr, ra, at, t#' and '#c, ca, ar, rt, t#' respectively and the 3-grams are '#cr, cra, rat, at#' and '#ca, car, art, rt#' respectively. The matched substrings will be, 2 for 2-gram and none for 3-gram. Therefore, making it difficult to find similarity between the two words.

This can also be illustrated with the help of some of the words that were given to us in the misspell.txt file, like 'celfie', 'akward', 'waz' & 'toi'.

Misspelled word	Correct word	Predicted word	Right/ Wrong?
celfie	selfie	celiocele	✕
		selfie	✓
akward	awkward	adward	✕
		award	✕
		awkward	✓
		hoardward	✕
		parkward	✕
		peakward	✕
waz	was	-	-
toi	to	ototoi	✕
		toi	✕
		toitoi	✕
		topoi	✕

Table 7: Example of 2-Gram

Precision = (2/12) * 100 = 16.67%
Recall = (2/4) * 100 = 50%

Misspelled word	Correct word	Predicted word	Right/ Wrong?
celfie	selfie	-	-
akward	awkward	awkward	✓
waz	was	-	-
toi	to	toi	✕
		toitoi	✕

Table 8: Example of 3-Gram

Precision = (1/3) * 100 = 33.33%
Recall = (1/4) * 100 = 25%

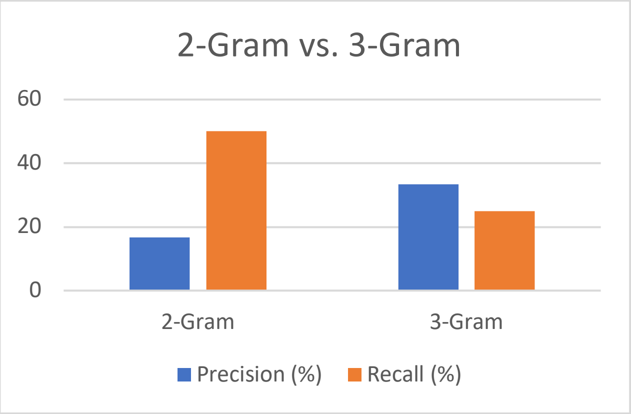

Figure 2: Effect of 2-Gram & 3-gram on Precision & Recall

As shown in Figure 2, the use of 2-Gram Distance resulted in higher Recall and lower Precision, whereas the 3-Gram has higher Precision and lower Recall because of the different lengths of substrings in both.

7. Conclusion

After careful analysis of both methods, Edit distance and N-gram with respect to different variables in each method, Edit distance was able to find more correct predictions with lower attempted predictions than the N-Gram method and had 1.29% more Precision and 9.91% more Recall than N-gram distance, making Levenshtein Distance a better method out of the two for approximate string matching.

References

Manges, B. (2005). *The use of Levenshtein distance in computer forensics.* Retrieved from ‘https://goo.gl/UUV3yr’.

Nicholson, J., Zobel, J., Verspoor, K. & Kotagiri, R. (2018). Approximate Matching [Lecture Notes]. Retrieved from ‘https://goo.gl/RqUTFg’.

Saphra, N. & Lopez, A. (2016). *Evaluating Informal-Domain Word Representations With UrbanDictionary*. Retrieved from ‘http://www.aclweb.org/anthology/W16-2517’.