SPELL CHECK ASSIGNMENT

Word-Phrase-Sentence

Professor:
Sutanu Chakraborti

Sidharth Aggarwal (CS17S012)
Vinay(CS15B010)

Contents

Chapter 1

WORD SPELL CHECK

1.1 Introduction

In the Word Spell Check we have used the the dictionary count_1w100k.txt containing the words and corresponding to those words the frequency of those words occuring in the corpus. In the dictionary the words are ordered in the descending order of the frequency of the words. Further we have used the bigrams to find the spelling errors. For the word spell check I have read the paper on the word spell check with noisy channel mentioned in the references below. For the task of spell check the code is taking input as a file containing the words which needs to be checked for the correction and output file will contain the rows containing the wrong word in the start and then corrected suggesions corresponding to the wrong word.

1.2 Approach

- In the word spell check, several techniques are implemented to get the best output. In the we have used the dictionary to know whether the word for which we need to check the spelling exist in the dictionary or not. If it is not in the dictionary then we will mark it as a wrong wrong and now apply processes on it.

- Now in this we have implemented the code to check whether the miss spelled word is matching with any word having the same phonetic voicing. For this purpose we have used the doublemetaphone library of python. For this we have made a dictionary containing the key as the phonetic voice and value to that value is the list of the words which are having same phonetic voicing.

- Further as in word spell check we have used the bigrams so we have made a dictionary with key as the any bigram and value to that key is the list of the words which has that bigram.

- As from the above two dictinaries we have gathered our candidates. The procedure for that is that we will make the bigrams of the miss spelled word and then find the words which is having those bigrams and mark them as the candidates. And further we will see the phonetic voice of the miss spelled word and check whether there is any candidate with that voice present or not.

- Now, for Ranking we have implemented a formula considering the editdistance between the miss spelled word and candidate.Further that include lenght of candidate and wrong word. And in the last the probability of occurence of that candidate in the dictionary.

$$Score = (max(len(candidate), len(wrongword)) - editdis) + \frac{dic[candidate]}{totalCount}$$

- Further technique to increase the score are :-
  - First letter of wrongword matching with candidate word
  - Second letter of wrongword matching with candidate word
  - Last letter of wrongword matching with candidate word
  - The lenght of candidate and wrongword is same

1.3 Findings/Results

About the output is that we are required to find the suggesions for the miss spelled words. So for that we are sorting the candidates in the descending order by the score of the score of the related to that candidate. So in this assignment we are outputting the atmost 3 suggestions for the miss spelled words. So the output is as below:-

WrongWord	Suggesion1	Suggesion2	Suggesion3
ocurance	occurance	occurence	occurences
laguage	language	languages	lagrange
emberassment	embarrassment	embarassment	

Chapter 2

PHRASE SPELL CHECK

2.1 Introduction

In the Phrase Spell Check we have used the the dictionary count_1w100k.txt containing the words and corresponding to those words the frequency of those words occuring in the corpus. In the dictionary the words are ordered in the descending order of the frequency of the words. Further we have used the bigrams to find the spelling errors. For the phrase spell check I have read the paper on the spell check for context spell check with Bayesian approach mentioned in the references below. For the task of spell check the code is taking input as a file containing the phrases which needs to be checked for the correction and output file will contain the rows containing the wrong word in the start and then corrected suggesions corresponding to the wrong word. Further in this we have used the file containing commonly occuring confusing words. The confusing words for e.g, dessert(a food item) and desert(an arid sandy place), rime and ryme etc. We have tried to capture as many as possible. And also to see the context checking we have used the Brown corpus which is containing the sentences through which we can check the probability of the candidate words present in certain context words. In this we have used the stopwords library of python to remove the stopwords(a,the,etc) from the phrase before processing them.

2.2 Approach

- In the phrase spell check, several techniques are implemented to get the best output. In the we have used the dictionary to know whether the word for which we need to check the spelling exist in the dictionary or not. If it is not in the dictionary then we will mark it as a wrong wrong and now apply processes on it.
- If no word in the phrase is having wrong spelling then we have send all the words for processing by removing the stopwords.
- Now in this we have implemented the code to check whether the for each word in the list of wrods for checking the suggesions is matching with any word having the

same phonetic voicing. For this purpose we have used the doublemetaphone library of python. For this we have made a dictionary containing the key as the phonetic voice and value to that value is the list of the words which are having same phonetic voicing.

- So in the phrase check for getting the candidate words we have used the bigram technique and the phonetic technique.
- For getting the scores we have used the formula of the bayesian context check. In that we find the conditional probability candidate given the context words as below:-

  $$P(w|c_1,\ldots,c_{i-1},c_{i+1},\ldots,c_n) = \frac{P(c_1,\ldots,c_{i-1},c_{i+1},\ldots,c_n|w)P(w)}{P(c)}$$

  In the above formula the denominator is just the scaling factor so we can ignore that so:-

  $$P(w|c_1,\ldots,c_{i-1},c_{i+1},\ldots,c_n) = P(c_1,\ldots,c_{i-1},c_{i+1},\ldots,c_n|w)P(w)$$

  Further we know have taken the assumption that all the context words are independent to each other so we can write :-

  $$P(c_1,\ldots,c_{i-1},c_{i+1},\ldots,c_n|w) = \Pi P(c_i|w)$$

  In this way the final formula is as:-

  $$P(w|c_1,\ldots,c_{i-1},c_{i+1},\ldots,c_n) = [\Pi P(c_i|w)]P(w)$$

- In this we have used a window of size k, for checking those many words on both the sides of the candidate word in the sentence in the brown corpus to calculate each individual probability of the context word given the candidate word,i.e,$P(c_i|w)$
- Now,after getting the score for Ranking we have implemented a the above formula for each candidate word and ordered them in the desecnding order of the above probability.
- Further there was one more issue we have tried to consider that when there are multiple words in the words of the phrase for which we have to find the candidite. So finally which word to choose to show the suggesions. To resolve this we have implemented the idea that if the 1st candidate after giving the ranking is equal to word for which we are checking the candidates, then we will skip that word for the showing the suggesions and more ahead to the next word in the list.

2.3 Findings/Results

About the output is that we are required to find the suggesions for the miss spelled words. So for that we are sorting the candidates in the descending order by the score of the score of the related to that candidate. So in this assignment we are outputting the atmost 3 suggestions for the miss spelled words. So the output is as below:-

WrongWord	Suggesion1	Suggesion2	Suggesion3
Hiway	highway	hilary	airway
krossd	crossed	kross	grossed
enugf	enough	enuff	engulf

Chapter 3

SENTENCE SPELL CHECK

3.1 Introduction

In the Sentence Spell Check we have used the the dictionary count_1w100k.txt containing the words and corresponding to those words the frequency of those words occuring in the corpus. In the dictionary the words are ordered in the descending order of the frequency of the words. Further we have used the bigrams to find the spelling errors. For the sentence spell check I have read the paper on the spell check for context spell check with Bayesian approach mentioned in the references below. For the task of spell check the code is taking input as a file containing the sentences needs to be checked for the correction and output file will contain the rows containing the wrong word in the start and then corrected suggesions corresponding to the wrong word. Further in this we have used the file containing commonly occuring confusing words. The confusing words for e.g, dessert(a food item) and desert(an arid sandy place), rime and rhyme etc. We have tried to capture as many as possible. And also to see the context checking we have used the Brown corpus which is containing the sentences through which we can check the probability of the candidate words present in certain context words. In this we have used the stopwords library of python to remove the stopwords(a,the,etc) from the phrase before processing them.

3.2 Approach

- In the sentence spell check, several techniques are implemented to get the best output. In the we have used the dictionary to know whether the word for which we need to check the spelling exist in the dictionary or not. If it is not in the dictionary then we will mark it as a wrong wrong and now apply processes on it.

- If no word in the phrase is having wrong spelling then we have send all the words for processing by removing the stopwords.

- Now in this we have implemented the code to check whether the for each word in the list of wrods for checking the suggesions is matching with any word having the same phonetic voicing. For this purpose we have used the doublemetaphone library of python. For this we have made a dictionary containing the key as the phonetic voice and value to that value is the list of the words which are having same phonetic voicing.

- So in the phrase check for getting the candidate words we have used the bigram technique and the phonetic technique.
- For getting the scores we have used the formula of the bayesian context check. In that we find the conditional probability candidate given the context words as below:-

  $$P(w|c_1, \ldots, c_{i-1}, c_{i+1}, \ldots, c_n) = \frac{P(c_1, \ldots, c_{i-1}, c_{i+1}, \ldots, c_n|w)P(w)}{P(c)}$$

  In the above formula the denominator is just the scaling factor so we can ignore that so:-

  $$P(w|c_1, \ldots, c_{i-1}, c_{i+1}, \ldots, c_n) = P(c_1, \ldots, c_{i-1}, c_{i+1}, \ldots, c_n|w)P(w)$$

  Further we know have taken the assumption that all the context words are independent to each other so we can write :-

  $$P(c_1, \ldots, c_{i-1}, c_{i+1}, \ldots, c_n|w) = \Pi P(c_i|w)$$

  In this way the final formula is as:-

  $$P(w|c_1, \ldots, c_{i-1}, c_{i+1}, \ldots, c_n) = [\Pi P(c_i|w)]P(w)$$

- In this we have used a window of size k, for checking those many words on both the sides of the candidate word in the sentence in the brown corpus to calculate each individual probability of the context word given the candidate word,i.e,$P(c_i|w)$
- Now,after getting the score for Ranking we have implemented a the above formula for each candidate word and ordered them in the descending order of the above probability.
- Further there was one more issue we have tried to consider that when there are multiple words in the words of the phrase for which we have to find the candidate. So finally which word to choose to show the suggesions. To resolve this we have implemented the idea that if the 1st candidate after giving the ranking is equal to word for which we are checking the candidates, then we will skip that word for the showing the suggesions and more ahead to the next word in the list.

3.3 Findings/Results

About the output is that we are required to find the suggesions for the miss spelled words. So for that we are sorting the candidates in the descending order by the score of the score of the related to that candidate. So in this assignment we are outputting the atmost 3 suggestions for the miss spelled words. So the output is as below:-

WrongWord	Suggesion1	Suggesion2	Suggesion3
chequer	checker	cheques	cheque
flem	flee	flm	clem
rime	rhyme		

Chapter 4

Analysis and Conclusion

4.1 Analysis

- In the word spell check we analysed that it is better to find the candidates on the basis of both the EditDistance and the Phonetic voices. As the errors made by the people are either typo or they don't know exact spelling but they tried to build the spelling by knowing the phonetics of it. For e.g, if we see the word chequer, so in this the phonetic is same as checker.

- For the spell check in the phrase and sentence we analysed that for get better accuracy in the output the corpus on which we are checking the context should be dense and large enough. With small corpus not much good results are obtained. And also the change in the window size may also affect the accuracy of the output suggestions.

- Analysed the Confusion words and documented in a file. Confusion words are those which create ambiguity either in spelling or phonetics. E.g. (dessert,desert) or (brake,break), etc.

- Analysed the stopwords(a, the, we, etc) using the library of python.

4.2 Conclusion

- Bigram approach is a good technique for correcting the spelling.

- We should select the candidates on the basis of similar phonetic voicing with certain constraints that the editdistance should not be more than a threshold.

- Window Size affect the accuracy of the suggestions in the context spell check.

- Bayesian Hybrid method for context spell check work well with sufficiently large corpus.

Chapter 5

REFERENCES

5.1 Papers

- A Spelling Correction Program Based on a Noisy Channel Model
  By Mark D. KernighanKenneth Ward ChurchKenneth Ward ChurchWilliam A. Gale

- A Bayesian hybrid method for context-sensitive spelling correction
  Andrew R.Golding Mitsubishi Electric Research Labs

5.2 Libraries and Corpus

- Python editdistance package for calculating the editdistance between two words

- Python stopwords package for removing the stopwords from the input sentence

- Python doublemetaphone package for finding the words with similar phonetics

- count_1w100k.txt file for the count of the numbers