NATURAL LANGUAGE PROCESSING

SPELL CHECK ASSIGNMENT

September 22, 2016

1 Introduction

This paper focuses on context sensitive spelling correction. Word check, phrase check and sentence check are the main points of interest. Context words method is used for context sensitive spell checking. Context words primarily depends on presence of particular words near the target word and uses probabilistic models for the same. For word check ,scoring was based on Kernighan and Church's paper on "A Spelling Correction Program Based on a Noisy Channel Model" and for phrase check,scoring was implemented by "A Bayesian Hybrid method for context-sensitive spelling correction" by Golding. Sentence check was implemented as a combination of both word check and phrase check procedures.

2 Word Spell Checker

2.1 Candidate Generation

Given a typo the problem of generating possible candidates efficiently is very important. Brute force kind of generation doesn't help here as the time complexity is O($(52n)^k$), where n=length of the typo and k is the maxing edit distance from the typo till which we are generating the candidates. So to efficiently generate the candidate corrections we use the following method :

```
1.Divide the dictionary into sets, each of which consists of words
  of same length.
2.Now for each word in the dictionary find all trigrams and bigrams.
3.Now create a map, trigramMap, which maps from each trigram to the
  corresponding list of words which contain it(inverse indexing).
  Construct a similar map, bigramMap, for bigrams as well.
4.Now get the lists ,triList and biList, of all possible trigrams
  and bigrams of the given typo.
5.For each trigram tr in the triList, get all the words from
  trigramMap[tr].
6.If the typo length is less than 6, for each bigram br in the
  biList, get all the words from bigramMap[br]
7.Store the words obtained in steps 5 and 6 in a vector
  possibleCorrections.
```

So the above method lets us prune out the words from the dictionary that are possible corrections of the typo to a great extent. The list of possible corrections can be further pruned out. The idea is that if a typo of a length 9 is given we

don't expect to get a correct word of length 3 or 4. So there is no point in keeping the words with such lengths in possibleCorrections even though they contain a trigram of the typo. So with this we arrive at the following conclusion to prune the list further :
Given a typo t, consider only those words in the possibleCorrections vector which satisfy the following condition : $|c_i.length() - t.length()| <= t.length()/3$.

2.2 Phonetic Suggestions

To give phonetic suggestions for a typo we used a phonetic algorithm called **Metaphone**.Metaphone is published by Lawrence Philips in 1990, for indexing words by their English pronunciation.Original Metaphone codes use the 16 consonant symbols 0BFHJKLMNPRSTWXY.This table summarizes most of the rules in the original implementation and we incorporated the following rules to convert a given word to its phonetic form:

1.Drop duplicate adjacent letters, except for C.
2.If the word begins with 'KN', 'GN', 'PN', 'AE', 'WR', drop the first letter.
3.Drop 'B' if after 'M' at the end of the word.
4.'C' transforms to 'X' if followed by 'IA' or 'H' (unless in latter case, it is part of '-SCH-', in which case it transforms to 'K'). 'C' transforms to 'S' if followed by 'I', 'E', or 'Y'. Otherwise, 'C' transforms to 'K'.
5.'D' transforms to 'J' if followed by 'GE', 'GY', or 'GI'. Otherwise, 'D' transforms to 'T'.
6.Drop 'G' if followed by 'H' and 'H' is not at the end or before a vowel. Drop 'G' if followed by 'N' or 'NED' and is at the end.
7.'G' transforms to 'J' if before 'I', 'E', or 'Y', and it is not in 'GG'. Otherwise, 'G' transforms to 'K'.
8.Drop 'H' if after vowel and not before a vowel.
9.'CK' transforms to 'K'.
10.'PH' transforms to 'F'.
11.'Q' transforms to 'K'.
12.'S' transforms to 'X' if followed by 'H', 'IO', or 'IA'.
13.'T' transforms to 'X' if followed by 'IA' or 'IO'. 'TH' transforms to '0'. Drop 'T' if followed by 'CH'.
14.'V' transforms to 'F'.
15.'WH' transforms to 'W' if at the beginning. Drop 'W' if not followed by a vowel.
16.'X' transforms to 'S' if at the beginning. Otherwise, 'X' transforms to 'KS'.
17.Drop 'Y' if not followed by a vowel.
18.'Z' transforms to 'S'.
19.Drop all vowels unless it is the beginning.

Few examples of words after applying above set of rules is shown below :
FANTASTIC - FNTSTK
ECSTASY - EKTS
DISTRACT - DSTRKT

Given a typo we find we find the phonetic respresentation of the typo and each word of the possibleCorrections vector.Now find $Pr(ph_{c_i}|ph_t)$, where ph_{c_i} and ph_t are phonetic forms of the word c_i and t respectively. Now find the word c_{maxph} that has maximum $Pr(ph_{c_i}|ph_t)$.

2.3 Suggesting Corrections

After the candidates are generated and c_{maxph} is determined, the suggestions for the typo are given as follows :

```
1.For each word in possibleCorrections find its edit distance with
  the typo.
2.Store ten words with the highest probability$Pr(word_i|typo)$ in a
  max heap.
3.Traverse the words in the max heap and if
  $Pr(ph_{c_{maxph}}|ph_t) > Pr(word_i|typo)$ and if $ph_t) > Pr(word_i|typo)$ is below a
  threshhold, which was set to 5000, insert $c_{maxph}$ into the max heap.
  Retrieve top ten elements of the max heap .
3.Display the top ten possible corrections for the typo.
```

3 Phrase and Sentence Spell Checker

For phrases and sentences, we started focusing on getting the context into account. Brown Corpus was used for this section. Word disambiguation was applied for context sensitive spell check.Suppose we have candidate words $w_1, w_2, \ldots, w_n$ (this set is called confusion set) among which one word is the correct option and each word wi is ambiguous with the other in the set. For example C=peace,piece ,if spelling program sees occurence of words like cake,apple etc., it detects that the context window of piece contains cake,apple and so on. Context words basically use the idea of particular words within ±k words of the ambiguous target word. After finding frequencies of these context words from the corpus, we can estimate their conditional probabilities of occurence in presence of the word under scrutiny. Bayes rule is used to find the probability of each w_i.

3.1 Identifying the ambiguous word

Given a phrase p, we do the following :

```
1.Parse the input phrase or sentence $p$ and find if any  misspelled word
is
present.
2.If missplelled is present
    1.Try to divide the misspelled word into dictionary words that
      are greater than a frequency threshhold th, set to 1000000.
    2.Calculate the split probability which is defined as follows
      $Pr(c_1, c_2, ...c_n) = freq(c_1) * freq(c_2) * ... * freq(c_n)/dictSize^n$.
    3.Return the list of splits with maximum $Pr(c_1, c_2, ...c_n)$.
    4.If the list is empty then perform word spell check for
       the word.
    5.Else output the list of the words in the split list.
3. If misspelled word is not present then perform context word
   analysis and list the possible corrections
```

3.2 Context word analysis

Every word w_i in the confusion set has distribution of words which are in the context, so for classifying this target word, we will see set of context words which are in the k length window around it.

```
1.Parse the input phrase or sentence $p$ and find the word $w_i$ which
is in confusion set
2.Take the corresponding confusion set and select the word $w$ from
this confusion set that has more probability , when context words $c_j$
of k-length window is given of that word.
          1.The probability for each word $w$ is calculated using Bayes
          rule
          $P(w|c_{-k},..,c_{-1},c_1...c_k) = \frac{P(c_{-k},..,c_{-1},c_1...c_k|w)P(w)}{P(c_{-k},..,c_{-1},c_1...c_k)}$.
          $p(c_{-k},...,c_{-1},c_1,...,c_k|w) = \prod_{j\in -k,...,-1,1...,k} p(c_j|w)$
     2.$p(c_j|w)$ is equals to $m/M$ where m is the number of such
          occurrences for which $c$ ocurred within k length window and $M$
          is the total no of occurrences of $w$ in the training corpus.
```

4 REFERENCES

1.https://en.wikipedia.org/wiki/Metaphone
2.http://www.aclweb.org/anthology/C90-2036
3.https://files.ifi.uzh.ch/cl/study/pp/data/corrector/TR95-13.pdf