CS6370− Natural Language Processing Spell Check Assignment Report

Aravind Sunil(CS16S004), Guttula Shanmukha Chaitanya(EE13B028), Moghe Nikita Vinay Sangeeta(CS16S016), Suman Banerjee(CS16S019)

Group Number: 7

1 Introduction

According to Wikipedia, spell checker (or spell check) is an application program that flags words in a document that may not be spelled correctly. Spell checkers may be stand-alone, capable of operating on a block of text, or as part of a larger application, such as a word processor, email client, electronic dictionary, or search engine. Over the years, a lot of papers have been published, using different approaches like 'A Bayesian Hybrid Method for Context-sensitive spelling correction', 'Google Web 5-gram approach', 'Mixed Trigram Approach' and 'Winnow based Approach', among ,any others. Our goal was to build a similar spell checker. The functionality of our system is to perform spell check on a given text and to suggest corrections. Basically, the system is divided into two parts:

1. Spell check on words (context insensitive)
2. Spell check on on phrases and sentences.(context sensitive)

We have tried to implement the Noisy Channel Method and Bayesian Hybrid Methods by Andrew R Golding for word spell check and sentence spell checkers respectively.

2 Resources and Tools

2.1 Corpora

1. Norvig Corpus for Word Check.
2. Brown Corpus for Context Sensitive Spell Check
3. Alpha Dictionary for confusion words.

2.2 Implementation tools

1. Python for high level programming
2. Wordnet POS tagger
3. Natural Language Toolkit

3 Context Insensitive Spell Check: For words

We have tried to implement the paper, 'A Spelling Correction Program Based On a Noisy Channel Model' by Mark D Kernighan, Kenneth W Church and William A Gale for word checker. It has been implemented as 3 parts:

1. Generation of probable corrections with edit distances of up to 2.
2. Generation of Confusion Matrices.
3. Ranking of words using Noisy Channel Model.

3.1 Generation of similar words and ranking

Generation The generation of words in terms of edit distances was done using Trie Data structure. Each word is represented as a leaf of the prefix tree(Trie).

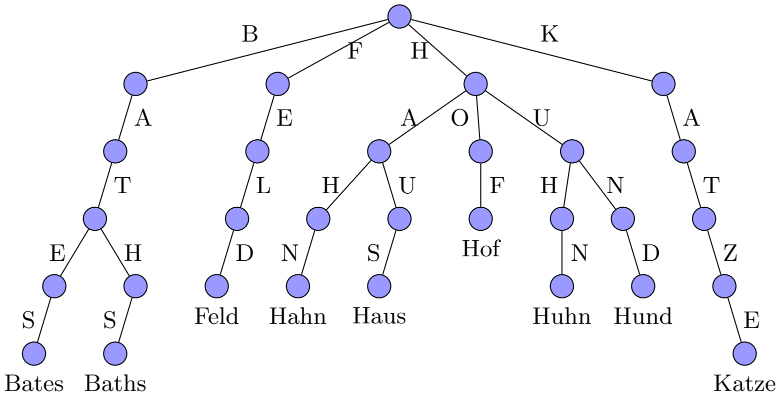

Fig. 1. Example of a trie

The nodes of Trie represents the first 'n' letters of a word and leaves represents the complete word. As evident from the above figure, we can find out the words at an edit edit distance of 'k'. Here we considered only up to edit distances of 3. In the above trie, 'baths' and 'bates' is at an edit distance of 1 with a substitution. Similarly, 'Haus' and 'Hahn' are at an edit distance 2 with 2 substitutions. Trie performed better than similar techniques in terms of time complexity and space requirements.

Case 1: Word Length $\leq$ 5 If word length of the given word is less than or equal to 5, we used candidates up to edit distance of 3.

Case 2: Word length 5 For word lengths greater than 5, we excluded candidates with 3 edit distances as occurrence chance for single and double letter typos are much more than 3 error typos. This gives more stability to the model.

Ranking The generated words were then ranked using the Noisy Channel Approach which follows Bayes Formula for probability estimation of words, w_i with edit distances up to 3 from the target word as follows:

$$p(c|t) = p(t|c) * p(c) \tag{1}$$

- p(c): probability of candidate word- frequency of candidate/total no of words
- P(t—c): probability of typo given the candidate. This was calculated using confusion matrices as:

$$p(t|c) = \begin{cases} \dfrac{del[c_{p-1}, c_p]}{chars[c_{p-1}, c_p]} & \text{if deletion} \\ \dfrac{add[c_{p-1}, t_p]}{chars[c_{p-1}]} & \text{if insertion} \\ \dfrac{sub[t_{p-1}, c_p]}{chars[c_p]} & \text{if substitution} \\ \dfrac{rev[c_p, c_{p+1}]}{chars[c_p, c_{p+1}]} & \text{if reversal} \end{cases} \tag{2}$$

The confusion matrices were obtained from count_1edit.txt of Norvig corpus, which contains counts for all single-edit spelling correction edits, from the file spell-errors.txt. The numerator of above equations refers to the numbers of times the corresponding (bigram,letter)/(letter/bigram)/(letter/letter) replacement has occurred. The denominator $char[x, y]$ or $char[x]$ refers to the number of times xy and x appeared respectively. This was obtained by counting the frequency of two letter combinations and single letters from the Norvig dictionary, count_1w100k.txt.

Inclusion of soundex Observing the results, we found that though some of the candidates with 3-edit distances had high chance of occurrence, they were getting negligibly small probabilities using Bayes Formula. Thus we included soundex approach to the existing model. Soundx generates codes for similar words. If the codes for candidates were found to be same as the target word, their probabilities were added by a α ,the mean probabilities of other candidates. This was done to include the important 3 edit candidates by some smoothing.

$$p(c|t) = p(c|t) + \alpha \tag{3}$$

Finally, the candidates were ranked with respect to probabilities to get top 10 suggestions.

4 Context Sensitive Spell Check: For phrases and sentences

In context sensitive checker, the steps followed are as follows:

1. Identifying the misspelled words.
2. Finding the set of confusion words, for the target word.
3. Generation of context words and collocations, referred to as features and calculation of feasibility of feature.
4. Matching features of target with that of the confusion words and ranking candidate words using Bayes Formula.

4.1 Identifying misspelled words

The misspelled word would be either a word absent in the lexicon or a word in the lexicon which is similar to the intended word.If word is absent in the lexicon, we follow the procedure for word checker as above.If word is present, we used the following approach. For example, consider the following sentence:

She is eating a **peace** *of cake*

In the above sentence, the word *piece* is misspelled as *peace*. *peace* is present n the lexicon and demands a context sensitive approach. Here, we have compared every word in the sentence with the set of confusion words from *alpha dictionary* , an online resource. If the word is present in the list, we can identify the probable correction and move on to the next phase. The confused word set for the corresponding ambiguous word are also available in the above resource.

whether	weather
I	me
past	passed
than	then
piece	peace
led	lead
rise	raise
except	accept
council	counsel
being	begin

Table 1. A sample of confused words

4.2 Finding context words and collocations

Context words Context words refers to the set of words around the ambiguous word, ie; the words appearing before and after. These words represents the context of our target word. By comparing the context words of target word and that corresponding confused words, words were selected from the latter list which closely matched the context of target word. NLTK was used on *Brown Corpus* for this purpose.

Collocations Collocations expresses the pattern of syntactic elements around the target word. Words and part of speech tags are the syntactic elements considered here. For example, considering the confused words, piece, peace, the following is a collocation, corresponding to sentences like 'piece of cake'.

$$\_\_\_ \, ofNN$$

Both context words and collocations are combined and used as features in the hybrid bayesian approach, as features. The combination of context words and collocations were generated from *Brown Corpus* using NLTK. First, we generated all possible bigrams and trigrams of all confused words. Part Of Speech tag sets were assigned to bigrams and trigrams using NLTK and Wordnet taggers. Consequently, collocations were generated from trigrams, where l=2, the maximum number of syntactic elements in a collocation. The features were then ranked using a reliability metric, so as to remove conflicting set of features.The reliability metric is expressed as:

$$reliability^{'}(f) = max \; p(w_i|f) \tag{4}$$

where w_i is a word in the confusion set. This value measures the extent to which the presence of the feature is unambiguously correlated with one particular w_i. We used this reliability metric to resolve conflicts among similar features such as the following set:

$$\_\_ \, walk$$
$$\_\_\_ \, V$$
$$CONJ \_\_ \, PREP$$

The feature with highest reliability metric was chosen among the conflicting set.

4.3 Ranking candidates using Bayes formula

Given the set of features $f_1, f_2, f_3 ... f_n$ and the set of confused words $w_1, w_2, w_3 ... w_n$, we find the probability for all confused words w_i using bayes formula as follows:

$$p(w_i|f_1, f_2, f_3, ...f_n) = \frac{p(f_1, f_2, f_3, ...f_n|w_i) * p(w_i)}{p(f_1, f_2, f_3, ...f_n)} \tag{5}$$

The denominator part is common for all w_i and we could ignore it. So the final estimation formula is:

$$p(w_i|f_1, f_2, f_3, ...f_n) = p(f_1, f_2, f_3, ...f_n|w_i) * p(w_i) \tag{6}$$

5 Results

5.1 Context insensitive spell check

The word spell checker passed all the test cases except few words like thruout. This failure is due to the edit distance of 3 which gave the correct word throughout a lower probability score. Some of the observations are as follows. Suggestions are in the form of suggestion-score:

- Given: bouyant
  Correction: buoyant 1.333 bryant 1.6890 courant 8.4923
- Given: extacy
  Correction: ecstacy 1.0 extech 0.1286 exotics 0.1285

5.2 Context sensitive spell check

The phrase/sentence spell checker worked well for about 87% of the given test cases. Some of the observations are as follows. Suggestions are in the form of suggestion-score:

- Given: The Parliament passed the *resoltion* to discuss the bill
  Correction: resolution 1.1667 resolutions 0.1667 resulting 0.6667
- Given: A *peace* of cake
  Correction: piece 2.0746 peace 1.0490

6 Conclusion and Future Scope

The goal of our work was to implement an effective spell checker program, working not only with non-word errors but also real world errors. The system performed well, as observed above, except in some special cases. The incorporation of soundex into the basic system helped us make it more powerful. Having said this, the spell checker always have a scope of improvement as none of the approaches are perfect till date. Latest approaches like Google N-gram and Winnow Based approaches could be included to facilitate accuracy and efficiency. It was found that 99% of non-word errors and 70% of the real word errors could be corrected using the former approach using N-grams. Winnow based approach also showed a similar rate of accuracy though both has limitations like speed. Hence, the future scope lies in optimizing the existing ideas to come up with better techniques.

References

1. A Spelling Correction Program Based on Noisy Channel Method. In: Proceedings of the 13th Conference on Computational Linguistics - Volume 2. pp. 205-210. COLING 90, Association for Computational Linguistics, Stroudsburg, PA, USA (1990)
2. Andrew R Golding: A Bayesian Hybrid Method for Context Sensitive Spelling Correction.In: In Proceedings of the Third Workshop on Very Large Corpora. pp. 39-53 (1995)
3. Natural Language Toolkit: Bird, Steven, Edward Loper and Ewan Klein (2009), Natural Language Processing with Python. OReilly Media Inc.
4. Natural Language Toolkit: Context-sensitive Spelling Correction Using Google Web 1T 5-Gram Information. In: Lebanese Association for Computational Sciences, Vol 5, May 2012.