Submission write up

Ankit Sharma

Approach

- **Generate all the words with 1 and 2 edit distance away from the input query and look up in the dictionary.** (Peter Norvig's approach)
  - This seemed a promising approach considering the time and task at hand.
  - Much better approaches exists like.
    - Faroo's Approach : Deriving deletes only with an edit distance<=2 both from the query term and each dictionary term.
    - Approach used Symspellpy package
    - Training charcter level language model using RNN’s
  - This approach is still better than the naïve approach of calculating the edit distance between the query term and every dictionary term.

Handling merged words like ‘HomeAssignment’

- Rough idea of approach:
  - (This is just brief sketch. Actual code tries to take care other corner cases as well)

If input_word is not present in the dictionary:

If (length (input_word)< 4):
generate suggestions for input word.
else
for each pair (L, R):
if (both L and R are present in the dictionary):
return L and R as it is by separating by space

else if (L is present i.e. R is not present):
return L as it is AND generate suggestions for R

else if (R is present i.e. L is not present):
return generate suggestions for L AND R as it is

else: (none of L and R are present)
generate suggestions for input word also.

Improvement areas

- A large corpus can be used to learn the joint probability of bigrams to better decompose the merged words.
- Joint probability distribution can also be used to rank the suggestions.
- Better approaches than Norvig's approach mentioned earlier can be adopted.
- Handling merged words derived by combining more than 2 words.