CS 517 - Natural Language Processing

Programming Assignment #3

Due date: 05/07/14 16:00 hrs

Statement

So far you have implemented a spelling correction model based on edit distances of it from a valid dictionary word. In this assignment, we will try to improve it further using language model. You are required to implement a language model based on n-gram modeling. Here, input will be a complete sentence and program will try to correct it.

Algorithm

EditDistanceModel: We have already implemented it where we just return the most frequent valid word. But for this assignment, we will return a set of possible words (1-edit-distance) with their probability (frequency / total). As per our training data, number of misspellings are only ~10%, which means $P(x|x) = 0.9$. Hence, this model tweaks the probability preferring same word 90% of the time, if it is a valid word.

ArgMax: For every word in the sentence, find all possible words returned by EditDistanceModel. Try computing probability of every possible sentences replacing one word at time. Return the sentence for which edit-probability * n-gram probability is maximum.

LanguageModel: Here, we will try to predict a word based on the context of the sentence given using an n-gram model. In n-gram model, with n>1, it is very important to capture start and end of a sentence. To facilitate this, every sentence is annotated with begin and end marker before passing to the model. So a sentence passed to you would look like: `<s> word1 word2 … wordn </s>`.

Basic implementation of both *EditDistanceModel* and *ArgMax* are provided. You only need to implement the ***LanguageModel***.

Code

Download pa3.zip where the basic framework for `SpellChecker.java` has been provided. You only have to implement `LanguageModel` class with *two* public and *one* private methods as follows:

- `public void train(Data corpus)`
  - This will be called only once with the entire corpus to learn
  - This method already implements learning of ***EditDistanceModel***, which is used in `correctSentence()` method. If you think, your language model is good

enough to ignore ***EditDistanceModel***, feel free to remove it. But most likely, you would need it.
    - Add your logic to this method to learn language model from the corpus.
- `private double score(List<String> sentence)`
    - This method scores given sentence on the basis of learnt langage model.
    - Score returned in log-probability scale and used by `correctSentence`() method. So, you change scale from log-probability to something else, you need to change `correctSentence()` as well.
- `public List<String> correctSentence(List<String> sentence)`
    - This method implements ArgMax algorithm to score between all possible sentences.
    - You don't need to modify this method to get good score. But encouraged to do so, if you think you need it.

Assumptions:

- For simplicity, this assignment assumes that only one word in entire sentence can be misspelled.
- Also, all misspellings are just 1 edit-distance away from original word.

Code Structure

`data/` -> This directory contains some tagged spelling data.
    `train` -> will be used for learning
    `test` -> will be used for testing your model
`java/` -> This directory contains all the JAVA code. You only need to modify `LanguageModel.java.`
`run.py` -> A tool to compile, run & test your code.

Extract and Run

Download the file pa3.zip. Extract it. Let's say you extracted into 'pa3' directory.
From terminal:

```
$cd pa3
# To run and evaluate your code
$./run.py
```

Compatibility

Note that, all assignments (including this one) will be tested under Linux environment with Python and ***OpenJDK 1.6*** (if you have 1.7 installed, make sure it compiles with `javac -target 1.6 *.java`) is installed. Given code might work on other platforms (like Windows, etc.) but has not been tested. Hence, it is encouraged to develop and test your code in a Linux based environment.

Submission

You should only modify and upload `LanguageModel.java` to Blackboard. **Any change in the filename will not be accepted and you will not be evaluated in that case**.

Evaluation

There is some held out data-set against which your code will be tested and evaluated. Hence, it is encouraged to write generic code that work in most of the cases. You are expected to employ Laplace Bigram language model. To give an idea on accuracy in given data set, a model with edit distance of 1 produces ~17% with uniform unigram model, ~21% with Laplace Unigram model, ~36% with Laplace Bigram model. You are free to read more techniques (e.g. Stupid Backoff) and use them to improve your accuracy.

Honor Code

I encourage students to discuss the programming assignments including specific algorithms and data structures required for the assignments. However, students should not share any source code for solution.

One of the most important principles of modern natural language processing is the use of a true unseen test set. Therefore, whenever we give a problem with training data, development data, and test data, you may not tamper with the submission process or scripts so as to attempt to look at the test data.

Code exists on the web for many problems including some that we may pose in problem sets or assignments. Students are expected to come up with the answers on their own, rather than extracting them from code on the web. This also means that we ask that you do not share your solutions to any of the homework - programming assignments, or problem sets - with any other students. This includes any sort of sharing, whether face-to-face, by email, uploading onto public sites, etc. Doing so will drastically detract from the learning experience of your fellow students.