LING 165 Lab 2: Minimum edit distance and the noisy channel model for spelling correction

Synopsis

(1) Write a function that calculates the minimum edit distance between two strings.

(2) Train a channel model for spelling correction following Kernighan et al. (1990).

Task 1

(1) Write a function that calculates the minimum edit distance between two strings assuming the following:

- three types of edit operations allowed: substitution, insertion, and deletion
- cost of edit operation is 1 for all three

(2) Using the function, calculate the minimum edit distance for every word-error pair in the data file (`spelling_error.pairs`) described below.

(3) Email me the following:

- percentage of pairs with distance of 1
- percentage of pairs with distance of 2
- where I can find your code for this task

Data

Use `/home/ling165/lab2/spelling_error.pairs` on the gray server. It has word-error pairs collected from a few misspelling corpora. Each line in the file specifies a pair in the following format:

```
word \t error
```

For example,

```
abattoir        abbatoir
aberration      aberation
```

Task 2

I want to build a noisy channel model for spelling correction similar to Kernighan et al. (1990). I already developed most of it. I just need the probability estimates for the channel model, i.e. $Pr(t|c)$ in p. 206 of Kernighan et al.'s paper. I want you to provide the probabilities estimated from the training data (`spelling_error.edits`). I also want you to evaluate the finished model on the test data (`correct_me.txt`). More specifically,

(1) Create a copy of `/home/ling165/lab2/sc.py` on the gray server and open it.

(2) Follow the instructions in the comments under `##TODO:` (lines 86–115). It basically says create a Python dictionary called `cmd` that contains channel model probabilities.

(3) Provide the `cmd` dictionary below the line in your `sc.py` that says

```
## Provide the cmd dictionary in the line below:
```

(4) Copy the following files from `/home/ling165/lab2/` and put them in the same directory as your `sc.py`:

- kn.py
- 2gram.kn
- vocab
- correct_me.txt

(5) Run your sc.py, which should fix errors in the test data (see below).

(6) Email me the following:

- Correction accuracy, i.e. percentage of spelling errors correctly fixed in the test data
- Where I can find your sc.py

Data

The spelling error data for this task consists of two files available on the gray server:

(1) /home/ling165/lab2/spelling_error.edits

(2) /home/ling115/lab2/correct_me.txt

Use (1) to train the channel model. It has word-error pairs from spelling_error.pairs for task 1 that are one Damerau-Levenshtein distance away from each other along with information on how the word needs to be edited to derive the error. Each line in the file has the following format:

word \t error \t edit,x,y

edit is delete, insert, substitute, or transpose. x and y are two letters involved in edit. More specifically,

- delete,x,y means delete y after x (i.e. xy → x).
- insert,x,y means insert y after x (i.e. x → xy).
- substitute,x,y means change x to y (i.e. x → y).
- transpose,x,y means swap x and y (i.e. xy → yx).

For example,

```
ability abilty  delete,l,i
ability abillity        insert,l,l
ability abilaty substitute,i,a
able    abel    transpose,l,e
```

The spelling corrector (sc.py) including the channel model you developed will be tested on (2). Each line in the file has the following format:

left-context \t <ERR targ=word> error </ERR> \t right-context

The spelling error is surrounded by ERR tags. The correct form is specified after targ=. left-context and right-context refer to texts to the left and right of the spelling error. For example,

```
i       <ERR targ=watch> wach </ERR>    it each night
the police      <ERR targ=came> cam </ERR>      out to look for the car
```

After you run sc.py, two more tab-separated fields will be inserted at the beginning of each line: (1) whether the program fixed the error correctly (1 for yes and 0 for no) and (2) how the program fixed the error. For example,

```
0       wash    i       <ERR targ=watch> wach </ERR>    it each night
1       came    the police      <ERR targ=came> cam </ERR>      out to look for the car
```

So the correction accuracy will simply be the percentage of lines that begin with `1`.

References

Kernighan, M. D., Church, K. W., and Gale, W. A. (1990). A spelling correction program based on a noisy channel model. In *Proceedings of the 13th conference on Computational linguistics-Volume 2*, pages 205–210. Association for Computational Linguistics.