CS 682: Project Report

Terence Henriod

December 15, 2014

Abstract

For this project an attempt was made to find a way to correct errors made by Optical Character Recognition methods in the scanning of French documents from 1880-1910 for historical research. While no final results were reached, some groundwork was laid for future endeavors.

1 Project Description

Dr. Christopher Church of the University of Nevada, Reno does research by studying historical context through sentiment analysis of news documents from relevant periods (specifically periodicals from 1880-1910 written in French). Dr. Church has experienced difficulty due to the fact that Optical Character Recognition (OCR) techniques are imperfect when scanning documents that are not high quality.

At the suggestion of Dr. Richard Kelley, a spelling correction approach was agreed upon. Since many of the words (or even characters) from the documents are correctly produced by the OCR methods, it is feasible to assume that a few words would be easily corrected with spelling correction techniques.

It has been demonstrated that spelling correction can be done with "toy" effectiveness with a simple approach. Peter Norvig has demonstrated (in a highly referenced blog post found at: http://norvig.com/spell-correct.html) that using simple edits of misspelled words, possible correct spellings within a relatively short "edit distance" of the misspelled word can be used along with a dictionary of correctly spelled words and their relative frequencies to determine a probable replacement for the misspelled word.

This concept can be improved upon, however, by using context to find better replacements for misspelled words. As Norvig has suggested, using adjacent words in a sentence is a promising extension of the basic spelling corrector he describes.

Dr. Kelley suggests using the Google N-gram data, which is freely available and quite sizable. Google's data spans over one hundred years, and includes many languages, including French. Google provides data for 1- to 5-grams, inclusive.

For this project, the Java programming language was chosen because it offers a good balance of cross-platform portability and performance, relative to some other languages. There is also a large number of companion libraries and development tools.

2 Data Collection

Collecting and managing the data for this project was a large task in and of itself. Google's N-gram data is so extensive that it took multiple machines several days to download and filter the data. While this was a simple task (a simple Java filter was written in under an hour, internet request and tar.gz stream library research and all), the data set is just so large that it presents issues in and of itself. Further, possible encoding issues may have arisen - the resulting files featured "words" that consisted of symbols such as lone commas. This should be easily remedied, but character encoding is an important issue that must be addressed for this task for more reasons than simple correct data transmission.

It should also be noted that the amount of data was incredibly large. The subset of French N-gram data that consisted of words used in 1880-1910 documents was over 600 GB. Not only is this far more data than can fit into the RAM of a typical computer system, it is more than a typical hard-drive can hold if it is not dedicated for this purpose. This increases the difficulty of accessing the data as necessary.

3 Basic Spelling Corrector

A basic spelling corrector was written according to Norvig's algorithm in Java. Currently it has precisely the same spelling correction as Norvig's Python version, and roughly the same speed performance, the Java version is perhaps slightly faster. However, it is expected that the Java version will be more easily expandable. It currently features the ability to be trained using N-gram data, although this capability is currently untested.

4 Performance for the Task

Unfortunately, the simple spelling corrector has not been tested on the task as of yet. This is largely due to the fact that the Google N-gram data is difficult to manage and utilize. Further, this is partly due to the fact that spelling corrections cannot be verified without someone who knows French.

5 Future Work

In order to make a fully functional French post-processor, I recommend work in the following three areas:

5.1 Character Encoding

Character encoding is a vital issue to address. Since there are many characters used in French that exist in the Unicode standard, but not the ASCII one, *a functional spelling corrector must handle UTF-8 or UTF-16 character encoding.* This should be relatively simple, but may require searching for suitable helper libraries if the standard Java String proves too unwieldy for the task.

5.2 Managing the Data

The data is large. It comes in files as large as 19 GB. There are hundreds of files. For this reason, a special indexing scheme should be considered to help improve manageability and performance. Dr. Kelley has also suggested compression, perhaps using the Google Protocol Buffer library.

5.3 Addition of Context Utilization

The whole aim of the project was to create a spell checker that uses context to achieve effective word correction. This functionality will rely less on coding algorithmic logic and more on finding a way to effectively search and utilize the N-gram data.

6 Conclusion

What this project may produce in the time to come is exciting - both in terms of producing an effective system for mitigating OCR errors in languages other than English as well as producing a homegrown system that is capable of effectively managing a very large dataset. This may result in some kind of distributed system, although it would be particularly exciting if it could be done on a single, high-performance machine.