E6893 Final Project Submission

Team Member(s): Kyle White

Project Summary: The project addresses the issue of distributed training for statistical language models in response to the motivating factor of ever increasing data sets available for training speech recognition language models, along with the increasing data volume needs of classifiers to yield high performance on speech processing problems.

Project Home: The project has been released as open source software under the Apache License, version 2.0. The library is hosted on Google Code with the project name 'ripley', and can be accessed at the link below. In addition to the source code, the site also contains a brief summary of the project, and an initial revision of a user guide.

http://code.google.com/p/ripley

Project Status: The project is following a popular semantic versioning scheme (i.e., http://semver.org), and thus will be considered released in its first official release version at 1.0.0. The revision has started at 0.0.1. The Google Code page allows for listing current Issues, and these represent the next steps in regards to planned technical contributions.

Graphical Tour: The user manual describes installing and running the project. Here, I will attach two scripts and example data to give a feel for using the library for the problem of distributed training for n-gram language models. There are two stages of processing. In the first stage, transcripts are input from HDFS and intermediate n-gram documents are written to the HDFS via distributed processing. In the second phase, the intermediate n-gram counts are read from HDFS and a language model representation is written to HDFS in an ARPA document format.

Phase I script to kickoff language model training:

```
#!/bin/bash

# This script is written to execute the speechtools library class LMTrainer
# The LMTrain is inteded to initiate a Hadoop MapReduce job to perform
# n-gram language model training on data in HDFS
# ${1} - input data path (file or directory) relative to the HDFS
# ${2} - output data path (file or directory) relative to the HDFS for ngram count int
ermediate document

export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:$(hadoop classpath)

export LIBJARS1=~/Software/lucene-4.10.2/analysis/common/lucene-analyzers-common-4.10.
2.jar,~/Software/lucene-4.10.2/core/lucene-core-4.10.2.jar

# Run hadoop job to create intermediate ngram document file via mapreduce job
hadoop --config $HADOOP_CONF_DIR jar ~/workspace/speechtools/target/speechtools-0.0.1-
SNAPSHOT.jar ripley.speechtools.client.LMTrainer -libjars $LIBJARS1 ${1} ${2}
```

This script kicks off distributed n-gram counting via the MapReduce implementation in Hadoop, and leads to intermediate n-gram count documents. An example of input to this step is below.

```
 6 <s> Global six victor bravo turn right at yankee two </s>
 7 <s> Acey fifty eight forty six fly runway heading runway three zero cleared for takeoff </s>
 8 <s> Beechjet one delta romeo contact Potomac departure </s>
 9 <s> Attention all aircraft atis quebec's current Dulles altimeter seven niner niner eight </s>
10 <s> Acey fifty eight forty six turn left heading two seven zero contact Potomac departure </s>
```

The output for phase I processing is multiple document, where one is created for each order of n-gram extracted from the training transcripts. An example of a tri-gram training document is below; it should only contain trigrams and their counts.

```
75122 zulu please </s>    1
75123 zulu ramp's one 1
75124 zulu roger taxi 1
75125 zulu runway one 1
75126 zulu runway three   2
```

The collection of n-gram count documents, where each line consists of the n-gram and the observation count, represents the input to phase II processing. The output of phase two processing is a language model representation saved to file in an ARPA format. A snippet of a constructed ARPA format language model document using the ripley library is below.

```
21304 -3.499307   you outside -0.155091
21305 -3.027635   uhh we  -0.155091
21306 -1.048397   papa turn   -1.030153
21307 -0.636688   national's airspace -0.155091
21308 -2.247552   it one  -0.155091
21309 -4.720337   <s> ac- -0.155091
21310 -2.517931   alpha left  -0.155091
21311 -2.371505   atis lima   -0.155091
21312 -2.335966   departure with  -0.632213
21313 -3.664310   two forty-  -0.155091
21314 -1.160186   block they  -0.155091
21315 -2.284937   care baw264 -0.155091
21316 -1.080543   westwind on -0.155091
21317 -2.438184   please i'll -0.155091
```

At this point the task of taking transcript training data and generating a statistical language model in the common ARPA language model format has been completed. The ARPA language model format is commonly supported by speech recognition toolkits, which is currently the target use of statistical language models by this project.