Data Science Specialization

Project Overview

Around the world, people are spending an increasing amount of time on their mobile devices for email, social networking, banking and a whole range of other activities. But typing on mobile devices can be a serious pain. SwiftKey, our corporate partner in this capstone, builds a smart keyboard that makes it easier for people to type on their mobile devices. One cornerstone of their smart keyboard is predictive text models. When someone types:

I went to the

the keyboard presents three options for what the next word might be. For example, the three words might be gym, store, restaurant. In this capstone you will work on understanding and building predictive text models like those used by SwiftKey.

This course will start with the basics, analyzing a large corpus of text documents to discover the structure in the data and how words are put together. It will cover cleaning and analyzing text data, then building and sampling from a predictive text model. Finally, you will use the knowledge you gained in data products to build a predictive text product you can show off to your family, friends, and potential employers.

You will use all of the skills you have learned during the Data Science Specialization in this course, but you'll notice that we are tackling a brand new application: analysis of text data and natural language processing. This choice is on purpose. As a practicing data scientist you will be frequently confronted with new data types and problems. A big part of the fun and challenge of being a data scientist is figuring out how to work with these new data types to build data products people love. The capstone will be evaluated based on the following assessments:

An introductory quiz to test whether you have downloaded and can manipulate the data.

An intermediate R markdown report that describes in plain language, plots, and code your exploratory analysis of the course data set.

Two natural language processing quizzes, where you apply your predictive model to real data to check how it is working.

A Shiny app that takes as input a phrase (multiple words), one clicks submit, and it predicts the next word.

A 5 slide deck created with R presentations pitching your algorithm and app to your boss or investor.

During the capstone you can get support from your fellow students, from us, and from the engineers at SwiftKey. But we really want you to show your independence, creativity, and initiative. We have been incredibly impressed by your performance in the classes up until now and know you can do great things.

We have compiled some basic natural language processing resources below. You are welcome to use these resources or any others you can find while performing this analysis. One thing to keep in mind is that we do not expect you to become a world's expert in natural language processing. The point of this capstone is for you to show you can explore a new data type, quickly get up to speed on a new application, and implement a useful model in a reasonable period of time. We think NLP is very cool and depending on your future goals may be worth studying more in-depth, but you can complete this project by using your general knowledge of data science and basic knowledge of NLP.

Here are a few resources that might be good places to start as you tackle this ambitious project.

[Text mining infrastucture in R](http://www.jstatsoft.org/v25/i05/)

[CRAN Task View: Natural Language Processing](http://cran.r-project.org/web/views/NaturalLanguageProcessing.html)

[Stanford Natural Language Processing MOOC](https://www.coursera.org/learn/nlp)

Course Tasks

This course will be separated into 8 different tasks that cover the range of activities encountered by a practicing data scientist. They mirror many of the skills you have developed in the data science specialization. The tasks are:

Understanding the problem

Data acquisition and cleaning

Exploratory analysis

Statistical modeling

Predictive modeling

Creative exploration

Creating a data product

Creating a short slide deck pitching your product

Assessements and Grading

To successfully complete the capstone project, you must receive a passing grade on all of the following assignments. The quizzes will be standard multiple choice quizzes. The other components are graded by peer evaluation. Your final grade will be calculated as follows.

Quiz 1: Getting Started (5%)

Milestone Report: exploratory analysis of the data set + evaluation of at least three classmate submissions (20%)

Quiz 2: Natural Language Processing I (10%)

Quiz 3: Natural Language Processing II (10%)

Final Project: your data product and a presentation describing your final data product + evaluation of at least three classmate submissions (55%)

Course dataset

[Capstone Dataset](https://d396qusza40orc.cloudfront.net/dsscapstone/dataset/Coursera-SwiftKey.zip)

Task 0: Understanding the Problem

The first step in analyzing any new data set is figuring out: (a) what data you have and (b) what are the standard tools and models used for that type of data. Make sure you have downloaded the data from Coursera before heading for the exercises. This exercise uses the files named LOCALE.blogs.txt where LOCALE is the each of the four locales en\_US, de\_DE, ru\_RU and fi\_FI. The data is from a corpus called [HC Corpora](http://www.corpora.heliohost.org). See the [readme file](http://www.corpora.heliohost.org/aboutcorpus.html) for details on the corpora available. The files have been language filtered but may still contain some foreign text.

In this capstone we will be applying data science in the area of natural language processing. As a first step toward working on this project, you should familiarize yourself with Natural Language Processing, Text Mining, and the associated tools in R. Here are some resources that may be helpful to you.

[Natural language processing Wikipedia page](https://en.wikipedia.org/wiki/Natural_language_processing)

[Text mining infrastucture in R](http://www.jstatsoft.org/v25/i05/)

[CRAN Task View: Natural Language Processing](http://cran.r-project.org/web/views/NaturalLanguageProcessing.html)

[Coursera course on NLP](https://www.coursera.org/course/nlp)

*Tasks to accomplish*

Obtaining the data - Can you download the data and load/manipulate it in R?

Familiarizing yourself with NLP and text mining - Learn about the basics of natural language processing and how it relates to the data science process you have learned in the Data Science Specialization.

*Questions to consider*

What do the data look like?

Where do the data come from?

Can you think of any other data sources that might help you in this project?

What are the common steps in natural language processing?

What are some common issues in the analysis of text data?

What is the relationship between NLP and the concepts you have learned in the Specialization?

Task 1: Getting and Cleaning the Data

Large databases comprising of text in a target language are commonly used when generating language models for various purposes. In this exercise, you will use the English database but may consider three other databases in German, Russian and Finnish.

The goal of this task is to get familiar with the databases and do the necessary cleaning. After this exercise, you should understand what real data looks like and how much effort you need to put into cleaning the data. When you commence on developing a new language, the first thing is to understand the language and its peculiarities with respect to your target. You can learn to read, speak and write the language. Alternatively, you can study data and learn from existing information about the language through literature and the internet. At the very least, you need to understand how the language is written: writing script, existing input methods, some phonetic knowledge, etc.

Note that the data contain words of offensive and profane meaning. They are left there intentionally to highlight the fact that the developer has to work on them.

*Tasks to accomplish*

Tokenization - identifying appropriate tokens such as words, punctuation, and numbers. Writing a function that takes a file as input and returns a tokenized version of it.

Profanity filtering - removing profanity and other words you do not want to predict.

*Tips, tricks, and hints*

Loading the data in. This dataset is fairly large. We emphasize that you don't necessarily need to load the entire dataset in to build your algorithms (see point 2 below). At least initially, you might want to use a smaller subset of the data. Reading in chunks or lines using R's readLines or scan functions can be useful. You can also loop over each line of text by embedding readLines within a for/while loop, but this may be slower than reading in large chunks at a time. Reading pieces of the file at a time will require the use of a file connection in R.

Sampling. To reiterate, to build models you don't need to load in and use all of the data. Often relatively few randomly selected rows or chunks need to be included to get an accurate approximation to results that would be obtained using all the data. Remember your inference class and how a representative sample can be used to infer facts about a population. You might want to create a separate sub-sample dataset by reading in a random subset of the original data and writing it out to a separate file. That way, you can store the sample and not have to recreate it every time. You can use the rbinom function to "flip a biased coin" to determine whether you sample a line of text or not.

Task 2: Exploratory Data Analysis

The first step in building a predictive model for text is understanding the distribution and relationship between the words, tokens, and phrases in the text. The goal of this task is to understand the basic relationships you observe in the data and prepare to build your first linguistic models.

*Tasks to accomplish*

Exploratory analysis - perform a thorough exploratory analysis of the data, understanding the distribution of words and relationship between the words in the corpora.

Understand frequencies of words and word pairs - build figures and tables to understand variation in the frequencies of words and word pairs in the data.

*Questions to consider*

Some words are more frequent than others - what are the distributions of word frequencies?

What are the frequencies of 2-grams and 3-grams in the dataset?

How many unique words do you need in a frequency sorted dictionary to cover 50% of all word instances in the language? 90%?

How do you evaluate how many of the words come from foreign languages?

Can you think of a way to increase the coverage -- identifying words that may not be in the corpora or using a smaller number of words in the dictionary to cover the same number of phrases?

Task 3: Modeling

The goal here is to build your first simple model for the relationship between words. This is the first step in building a predictive text mining application. You will explore simple models and discover more complicated modeling techniques.

*Tasks to accomplish*

Build basic n-gram model - using the exploratory analysis you performed, build a basic n-gram model for predicting the next word based on the previous 1, 2, or 3 words.

Build a model to handle unseen n-grams - in some cases people will want to type a combination of words that does not appear in the corpora. Build a model to handle cases where a particular n-gram isn't observed.

*Questions to consider*

How can you efficiently store an n-gram model (think Markov Chains)?

How can you use the knowledge about word frequencies to make your model smaller and more efficient?

How many parameters do you need (i.e. how big is n in your n-gram model)?

Can you think of simple ways to "smooth" the probabilities (think about giving all n-grams a non-zero probability even if they aren't observed in the data) ?

How do you evaluate whether your model is any good?

How can you use backoff models to estimate the probability of unobserved n-grams?

*Hints, tips, and tricks*

As you develop your prediction model, two key aspects that you will have to keep in mind are the size and runtime of the algorithm. These are defined as:

Size: the amount of memory (physical RAM) required to run the model in R

Runtime: The amount of time the algorithm takes to make a prediction given the acceptable input

Your goal for this prediction model is to minimize both the size and runtime of the model in order to provide a reasonable experience to the user.

Keep in mind that currently available predictive text models can run on mobile phones, which typically have limited memory and processing power compared to desktop computers.

Therefore, you should consider very carefully (1) how much memory is being used by the objects in your workspace; and (2) how much time it is taking to run your model. Ultimately, your model will need to run in a Shiny app that runs on the shinyapps.io server.

*Tips, tricks, and hints*

Here are a few tools that may be of use to you as you work on their algorithm:

object.size(): this function reports the number of bytes that an R object occupies in memory

Rprof(): this function runs the profiler in R that can be used to determine where bottlenecks in your function may exist. The profr package (available on CRAN) provides some additional tools for visualizing and summarizing profiling data.

gc(): this function runs the garbage collector to retrieve unused RAM for R.

In the process it tells you how much memory is currently being used by R.

There will likely be a tradeoff that you have to make in between size and runtime. For example, an algorithm that requires a lot of memory, may run faster, while a slower algorithm may require less memory. You will have to find the right balance between the two in order to provide a good experience to the user.

Task 4: Prediction Model

The goal of this exercise is to build and evaluate your first predictive model. You will use the n-gram and backoff models you built in previous tasks to build and evaluate your predictive model. The goal is to make the model efficient and accurate.

*Tasks to accomplish*

Build a predictive model based on the previous data modeling steps - you may combine the models in any way you think is appropriate.

Evaluate the model for efficiency and accuracy - use timing software to evaluate the computational complexity of your model. Evaluate the model accuracy using different metrics like perplexity, accuracy at the first word, second word, and third word.

*Questions to consider*

How does the model perform for different choices of the parameters and size of the model?

How much does the model slow down for the performance you gain?

Does perplexity correlate with the other measures of accuracy?

Can you reduce the size of the model (number of parameters) without reducing performance?

Task 5: Creative Exploration

So far you have used basic models to understand and predict words. In this next task, your goal is to use all the resources you have available to you (from the Data Science Specialization, resources on the web, or your own creativity) to improve the predictive accuracy while reducing computational runtime and model complexity (if you can). Be sure to hold out a test set to evaluate the new, more creative models you are building.

*Tasks to accomplish*

Explore new models and data to improve your predictive model.

Evaluate your new predictions on both accuracy and efficiency.

*Questions to consider*

What are some alternative data sets you could consider using?

What are ways in which the n-gram model may be inefficient?

What are the most commonly missed n-grams? Can you think of a reason why they would be missed and fix that?

What are some other things that other people have tried to improve their model?

Can you estimate how uncertain you are about the words you are predicting?

Task 6: Data Product

The goal of this exercise is to create a product to highlight the prediction algorithm that you have built and to provide an interface that can be accessed by others via a Shiny app.

*Tasks to accomplish*

Create a data product to show off your prediction algorithm You should create a Shiny app that accepts an n-gram and predicts the next word.

*Questions to consider*

What are the most interesting ways you could show off your algorithm?

Are there any data visualizations you think might be helpful (look at the Swiftkey data dashboard if you have it loaded on your phone)?

How should you document the use of your data product (separately from how you created it) so that others can rapidly deploy your algorithm?

*Tips, tricks, and hints*

Consider the size of the predictive model you have developed. You may have to sacrifice some accuracy to have a fast enough/small enough model to load into Shiny.

Task 7: Slide Deck

The goal of this exercise is to "pitch" your data product to your boss or an investor. The slide deck is constrained to be 5 slides or less and should: (1) explain how your model works, (2) describe its predictive performance quantitatively and (3) show off the app and how it works.

*Tasks to accomplish*

Create a slide deck promoting your product. Write 5 slides using RStudio Presenter explaining your product and why it is awesome!

*Questions to consider*

How can you briefly explain how your predictive model works?

How can you succinctly quantitatively summarize the performance of your prediction algorithm?

How can you show the user how the product works?

*Tips, tricks, and hints*

Information on [Rstudio presentation](https://support.rstudio.com/hc/en-us/articles/200486468-Authoring-R-Presentations).

Final Project Submission

*Data Product*

Does the link lead to a Shiny app with a text input box that is running on shinyapps.io?

Does the app load to the point where it can accept input?

When you type a phrase in the input box do you get a prediction of a single word after pressing submit and/or a suitable delay for the model to compute the answer?

Put five phrases drawn from Twitter or news articles in English leaving out the last word. Did it give a prediction for every one?

*Slide Deck*

Does the link lead to a 5 slide deck on R Pubs?

Does the slide deck contain a description of the algorithm used to make the prediction?

Does the slide deck describe the app, give instructions, and describe how it functions?

How would you describe the experience of using this app?

Does the app present a novel approach and/or is particularly well done?

Would you hire this person for your own data science startup company?