Technology Review
CS410 - Text Information Systems - Fall 2020

Tools for Sentiment Analysis in Lingpipe

Introduction

Sentiment analysis is the problem of determining the gist of a chunk of text, whether it be positive or negative, in an automated fashion. It is a common problem tackled in the field of NLP and Machine Learning. Many businesses deal with customer reviews and feedback on a scale that would be expensive to deal with manually, so tools to classify them as positive or negative with little human effort are valuable. The problem has been explored for many years now, so there are a number of libraries and tools available to help in this sphere.

This review will discuss how one might approach sentiment analysis with LingPipe as an introduction to LingPipe and how it is applied to NLP problems. It will compare this to the same problem, tackled with NLTK, another toolkit.

What is LingPipe?

Lingpipe is a toolkit and library for linguistic analysis. It is maintained and supported by alias-i, who sells several different kinds of licenses and offers consulting services to train and support it. It provides a Java API that offers classes that can be straightforwardly used in Java code to tackle NLP problems.

Alias-i provides a lot of online documentation and tutorials that make trying it out very simple. The classes it offers sound a bit like the classes found in sci-kit learn, i.e. various classifier classes and clustering tools, but with a focus on NLP. The classes offer a simple API for training and testing models, as will be shown later.

Sentiment Analysis in LingPipe

Sentiment analysis (S.A.) is a specific application of ML and NLP that classifies articles as positive or negative sentiment. LingPipe offers powerful general-purpose classifiers and tools that can be put together to build a sentiment analysis classifier. It does not perform S.A. out-of-the-box. However, alias-i's extensive documentation gives a solid tutorial that covers how to put one together with not very much code. This tutorial also gives a good introduction to how LingPipe's classes work.

The tutorial implements, in LingPipe, Bo Pang and Lillian Lee's 2004 ACL paper "A sentimental education." This paper builds a sentiment analyzer by first classifying sentences as subjective

or objective, and then running the subjective ones through a polarity classifier (positive or negative). It performs better than a simpler polarity classifier. The individual classifiers can be easily tuned.

```
String[] sentences = data.split("\n");
System.out.println("# Sentences " + category + "=" + sentences.length);
int numTraining = (sentences.length * 9) / 10;
for (int j = 0; j < numTraining; ++j) {
    String sentence = sentences[j];
    numTrainingChars += sentence.length();
    Classified<CharSequence> classified
        = new Classified<CharSequence>(sentence,classification);
    mClassifier.handle(classified);
}
```

Training the model is as easy as instantiating “Classified” objects based on labeled training data, and feeding these into the classifier object.

This tutorial used a “DynamicLMClassifier” -- one of many classifiers offered by the API -- to classify polarity. It is a language-model, configured to nGram=8 in this case. Other classifiers could easily be swapped in.

To implement Pang and Lee’s paper, the tutorial steps the reader through writing a simple subjectivity classifier, writing that trained model to disk, and then consuming it in a hierarchical model that applies both the subjectivity and the polarity classifiers. Then, it shows how to roll this combined classifier into a single class implementation, that implements the same interface as the other LingPipe classifiers. The tutorial, though many years old, still runs cleanly on the latest LingPipe download.

What is it good for?

There are many options for implementing sentiment analysis today. There are even several off-the-shelf products that will perform it on well-known social media platforms, tracking a hashtag or a brand, without even needing to download an application. LingPipe can probably be used to do that very thing, but it will take writing a whole application around it to process the data. LingPipe is:

- Mature, and established. It has been around for 15-20 years.
- Commercial, not academic. LingPipe is designed to scale and support production applications. They have a licensing and pricing structure that go with that.
- Built on Java. Many tools today are for Python. If Java is already in use, that may make it a good fit.
- Well-supported. Alias-i offers consulting services, and claims they are designed to teach the product, rather than promote dependency. The documentation is also high-quality, and extensive.
- Class-based (Java-like), flexible, and customizable.

For comparison, NLTK is a Python toolkit in this space that could be used to write a custom sentiment analysis tool. Compared to LingPipe, NLTK has similar maturity (20 years). Its heritage is academic, so it may not be designed to scale as much as it is designed to teach. Looking through a tutorial for sentiment analysis in NLTK, it appears to offer a similar level of customizability, but the tutorial covers more data cleaning and preparation that wasn't required for LingPipe's tutorial's dataset. Where Alias-i provided many getting-started and tutorial documents as well as API documentation, NLTK offers just API documentation. However, there are plenty of examples of its use elsewhere online.

Conclusion

LingPipe is a stable, established, well-documented and supported Java API that's well equipped to implement sentiment analysis, and many other computational linguistics and text processing tasks. It has a free license for research if the data is publicly available, so depending on the project, it may be just as good for early experimentation and research as more academic toolkits such as NLTK. It is advertised as ready to be taken to production, too.

References

A sentimental education: Sentiment analysis using subjectivity
Bo Pang and Lillian Lee
Proceedings of ACL, pp. 271--278, 2004
http://www.cs.cornell.edu/home/llee/papers/cutsent.home.html

LingPipe documentation by alias-i
The Sentiment Analysis tutorial:
http://www.alias-i.com/lingpipe/demos/tutorial/sentiment/read-me.html
Comparisons to other toolkits:
http://www.alias-i.com/lingpipe/web/competition.html
API documentation of the classifier used in the Sentiment Analysis tutorial.
http://www.alias-i.com/lingpipe/docs/api/com/aliasi/classify/LMClassifier.html

Several modern, commercial, ready-to-use Sentiment Analysis tools.
https://blog.hubspot.com/service/sentiment-analysis-tools

A tutorial for sentiment analysis using NLTK.
https://www.digitalocean.com/community/tutorials/how-to-perform-sentiment-analysis-in-python-3-using-the-natural-language-toolkit-nltk

NLTK documentation.
http://www.nltk.org/py-modindex.html