LingPipe for Sentiment Analysis

Tech Review

CS 410 Text Information Systems

Gazi Muhammad Samiul Hoque
ghoque2@illinois.edu

Fall 2021

1 Introduction

LingPipe is a software library for implementing computational linguistics to process text data. It is commonly used to perform activities such as looking for the names of entities, locations; automatic classification of text streams like Twitter search results, fixing precise spelling, part-of-speech (POS) tagging, sentence detection, and many more [2]. In addition, we can use it to build natural language processing (NLP) applications for multiple languages and genres and for many kinds of applications [1], including performing sentiment analysis on text data.

LingPipe Architecture

The architecture of LingPipe is designed to be efficient, scalable, reusable, and durable. It supports multi-lingual, multi-domain, multi-genre models for training with new data for new tasks. In addition, it provides thread-safe models and decoders for concurrent-read exclusive-write synchronization, and character encoding-sensitive I/O are just a few of the highlights. [2]

2 Sentiment Analysis

Sentiment analysis or opinion mining is the systematic identification, extraction, quantification, and study of affective states and individual information using natural language processing, text analysis, computational linguistics, and biometrics. Sentiment analysis is commonly used in marketing, consumer service, and clinical medicine to analyze the view of the customer feedbacks, such as reviews and questionnaire replies from online and social media resources. It generally involves classifying opinions in text into categories like "positive" or "negative" and seldom with an implicit type of "neutral." These analyses can be consolidated into recommendation systems or can be used to summarize people's experiences and opinions that consist of personal emotions derived from reviews.

2.1 Sentiment Analysis using LingPipe

At a high level, the idea is to use LingPipe's language classification framework to separate subjective from objective sentences and draw a clear distinction between positive and negative reviews. Bo Pang and Lillian Lee's 2004 ACL paper "A sentimental education" [4] described this process as the following illustration:

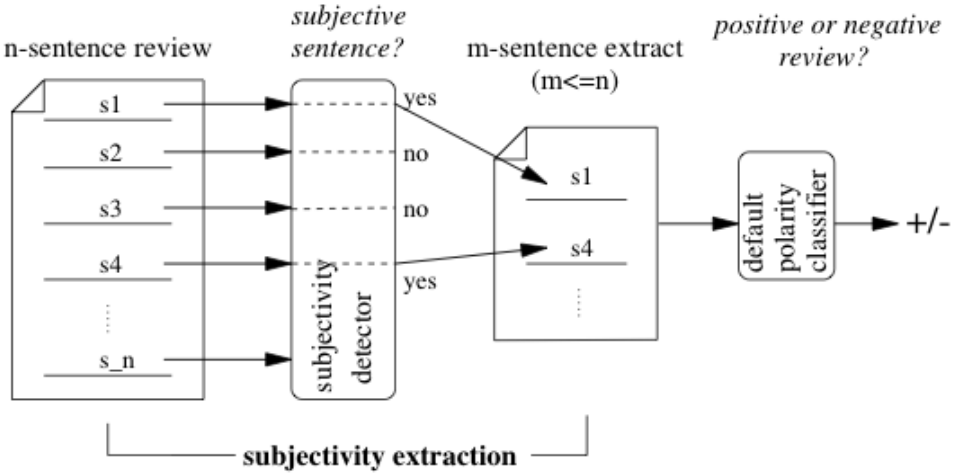

Figure 1: Polarity classification via subjectivity detection [4]

In this approach, a subjectivity detector is first employed that determines whether each sentence is subjective or not, eventually discarding the objective ones. This creates an extract that better represents a review's subjective content which is processed through a default polarity classifier. [4]

2.2 Basic Polarity Classification using LingPipe

For polarity classification, we can provide the example of classifying full-text movie reviews data from Pang and Lee's Dataset [5]. It has 1000 positive, 1000 negative full-text movie reviews extracted from IMDB's archive and heuristic scripts used to pluck the first review score from the text. The basic polarity classification works the following way: [3, 6]

Initialization First, the directory named `txt_sentoken` is set relative to the top-level polarity data directory. Then, the category array is initialized using the directory names under `txt_sentoken`, "pos," and "neg." Finally, the n-gram length is set to 8.

Tokenization The default factory is used to construct a bounded n-gram classifier with the specified categories and n-gram size. The process models are normalized for a given input length and do not model the boundaries of strings differently than other positions.

Training This method runs through the classes, of which there are two in this example. It then generates a directory using the polarity data directory and the name of the

category. Then, the training files are listed and iterated. For each training file, the text is read from the file and then used to train the classifier for the specified category.

Testing We can use the `classify` method to perform the actual work. It returns a `Classification` instance whose `bestCategory()` method returns the best category for the given text snippet as "pos" or "neg."

3 Conclusion

For interpretation of viewpoints from text data, LingPipe provides numerous accessible and robust tools. In addition, LingPipe can be an excellent tool for any NLP scientist, given that it has all other additional natural language processing features.

References

[1] Carpenter, B., and Baldwin, B. *Text Analysis with LingPipe 4*. LingPipe Publishing, New York, 2011.

[2] LingPipe. LingPipe Home. http://www.alias-i.com/lingpipe/index.html.

[3] LingPipe. Sentiment Analysis Tutorial. http://www.alias-i.com/lingpipe/demos/tutorial/sentiment/read-me.html.

[4] Pang, B., and Lee, L. A Sentimental Education. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics - ACL '04* (Morristown, NJ, USA, 2004), Association for Computational Linguistics.

[5] Pang, B., and Lee, L. Movie Review Data - Three Dataset. http://www.cs.cornell.edu/people/pabo/movie-review-data/, 2004.

[6] Reese, R. M., and Bhatia, A. S. Classifying Texts And Documents - Sentiment analysis using LingPipe. In *Natural Language Processing with Java*, 2nd ed. Packt Publishing, jul 2018.