Overview of Apache Lucene and its indexing features for east Asian languages

Xiaofan Liu | xl109@illinois.edu | CS410 Tech Review Report | 2021 Fall

Overview

Originally wrote by Doug Cutting 22 years ago in 1999, Apache Lucene has become a powerful, accurate and efficient indexing and searching engine library for both academic and enterprise applications. Many sub-projects under Lucene, such as Solr, Compass and Swiftype, have since become top level independent projects licensed by the Apache Foundation family.

The Lucene core library is written entirely in Java, but it has been ported to many other languages such as Python, C++, Ruby and PHP. As with other text search engines, Lucene "abstracts away the complexity of indexing and searching" algorithms and allows a programmer to focus on the functionality of the application.

Main Features and Flexibility

Although Lucene's quick response is regarded as the main reason for its success, in my opinion, its flexibility in many aspects has also contributed considerably to its fame and longevity.

First of all, Lucene has the ability to search in many types of documents including but not limited to text documents, html files, pdfs, emails, etc.

Second, Lucene has over 50 ranking functions built-in for a programmer to choose from including an optimized version of the popular BM25 and Bayesain smoothing using Dirichlet priors which we are familiar with from taking the CS410 course.

Furthermore, Lucene has support for over 30 languages around the globe and it boasts multiple ways for certain languages of analyzing and tokenizing text documents in order to build the most meaningful indexes. The next section of this report takes Lucene's *analysis.cjk* class as an example to further demonstrate this point.

Indexing and analyzing east Asian languages

As mentioned above, searching through the indexes rather than the unstructured text data is one of the main reasons for Lucene's quick response time. The key to building such effective indexes is text analysis. Analyzing east Asian language has brought new challenges to traditional unigram or bag of words approach. Certain languages such as Chinese and Japanese do not have spaces between words as delimiters to denote their boundaries. Therefore, for humans, the segmentation of words is inferred through understanding of the combination of characters within the context. But how can this approach be implemented on a computer?

The cjk class of the Lucene library incorporates some solutions to the analysis and tokenization of east Asian languages, namely Chinese, Japanese and Korean. The typical cjk analyzer performs bigram term indexing for Han, Hiragana, Katakana, or Hangul characters, which generates overlapping groups of two adjacent characters. Intuitively, this approach is fast and efficient because two-character words are the most common at least in Chinese. But it is unreliable. In other words, it has a poor recall rate. For Chinese, the other two analyzers provided are the ChineseAnalyzer and the SmartChineseAnalyzer. While the former is simply a unigram analyzer, the latter attempts to segment Chinese text into meaningful words of various length in characters. It can also analyze mixed Chinese-English text.

The SmartChineseAnalyzer is the most powerful among all three analyzers available in Lucene. The text is first broken into sentences delimited by punctuation marks then each sentence is segmented into words. The segmentation process is based upon the Hidden Markov Model, and it requires a dictionary to provide statistical data such as frequently used Chinese words as well as a large training corpus for optimal word segmentation.

Hidden Markov Model

The Hidden Markov Model (HMM) is a statistical model where one tries to learn about unobservable ("hidden") states by analyzing the probabilities of observable processes that are the outcomes of the unobservable states. A popular example is a weather guessing game where one guesses the weather in a remote location by looking at the clothes a person is wearing at that location only, assuming the choice of clothes is solely determined by the weather. In this example the weather is the unobserved state and the clothes one wears is the observation.

One approach to implement HMM for Chinese lexical analysis is Hierarchical HMM. It is a multi-level framework that includes atom segmentation, simple and recursive unknown word recognition, class-based segmentation and POS tagging. An atom is defined as the smallest unit such as a single Chinese character or a punctuation mark. In this HMM, the atom is the state while the original symbol is the observation. The centerpiece of this methodology is the Class-based HMM word segmentation. During this step, each word is given a class such as name, time, number, location, etc. But before the atoms can be combined into words, several issues arise including ambiguity (one character could form two different words with the character before or after it) and unknown words (words that are not present in the dictionary of frequent words). To tackle the second problem, this approach performs pattern matching by tagging each character with its possible "role" in a word. A role is comparable to a sub-class, e.g., a surname is a role within the class of person names. Then it tests all possible sequences of these roles to find the optimal pattern that conforms to the language grammar. E.g., a surname character followed by two characters that do not form common words could be a name.

Conclusion

Lucene is a powerful tool when it comes to indexing and searching east Asian languages thanks to its incorporation of many potent analyzers. However, it is worth noting that analyzing and tokenizing Chinese, Japanese and Korean is still an actively growing field and many researchers are working on its improvement.

References

1. Apache Lucene: free search for your website.
   https://www.ionos.com/digitalguide/server/configuration/apache-lucene/
2. What is Apache Lucene?
   https://techmonitor.ai/techonology/hardware/apache-lucene
3. Searching and Indexing With Apache Lucene
   https://dzone.com/articles/apache-lucene-a-high-performance-and-full-featured
4. Retrieval of bibliographic records using Apache Lucene
   Milosavljević, B. (1), Boberić, D. (2) and Surla, D. (2) (no date) 'Retrieval of bibliographic records using Apache Lucene', Electronic Library, 28(4), pp. 525–539. doi: 10.1108/02640471011065355.
5. Apache Lucene 8.10.1 Documentation
   https://lucene.apache.org/core/8_10_1/index.html
6. Apache Lucene
   https://en.wikipedia.org/wiki/Apache_Lucene
7. The Challenges of Chinese and Japanese Searching
   https://hybrismart.com/2019/08/18/the-challenges-of-chinese-and-japanese-searching/#SmartChineseAnalyzer
8. Chinese Lexical Analysis Using Hierarchical Hidden Markov Model
   Zhang, H.P., Liu, Q., Cheng, X., Zhang, H. and Yu, H.K., 2003, July. Chinese lexical analysis using hierarchical hidden markov model. In Proceedings of the second SIGHAN workshop on Chinese language processing (pp. 63-70).