



# Improving Elasticsearch for Chinese, Japanese, and Korean Text Search through Language Detector

Ki-Ju Kim<sup>1</sup>, and Young-Bok Cho<sup>2\*</sup> , Member, KIICE

<sup>1</sup>Support Team, Elastic, Seoul 06168, Korea

<sup>2</sup>Assistant Professor, Department of Information Security, Daejeon University, Daejeon 34520, Korea

## Abstract

Elasticsearch is an open source search and analytics engine that can search petabytes of data in near real time. It is designed as a distributed system horizontally scalable and highly available. It provides RESTful APIs, thereby making it programming-language agnostic. Full text search of multilingual text requires language-specific analyzers and field mappings appropriate for indexing and searching multilingual text. Additionally, a language detector can be used in conjunction with the analyzers to improve the multilingual text search. Elasticsearch provides more than 40 language analysis plugins that can process text and extract language-specific tokens and language detector plugins that can determine the language of the given text. This study investigates three different approaches to index and search Chinese, Japanese, and Korean (CJK) text (single analyzer, multi-fields, and language detector-based), and identifies the advantages of the language detector-based approach compared to the other two.

**Index Terms:** CJK, Elasticsearch, Full text search, Language detector, Open source

## I. INTRODUCTION

Elasticsearch is an open source search and analytics engine that can search petabytes of data [1]. It returns a list of matching documents sorted by scores. The scores are calculated based on the term and inverse document frequencies. The term frequency (TF) for a term in a document increases when the term appears more often in the document, whereas the inverse document frequency (IDF) decreases when the term appears more often in the index. This is based on the assumption that if a term appears often in a document, it must be an important one; however, if it appears often in the whole index, it may be a common word such as “a,” “is,” or “the” that is not considerably important.

Furthermore, it uses analyzers to convert text into a stream of tokens or terms. For example, the standard analyzer converts “평창은 제 123 차 IOC 총회에서 2018 년 동계올림픽대

회의 개최지로 선정되었습니다.” into “평창은”, “제 123 차,” “ioc,” “총회에서,” “2018 년,” “동계올림픽대회의,” “개최지로,” and “선정되었습니다.”

This paper presents the differences between the default analyzer and the language specific analyzers and introduces three different approaches to index Chinese, Japanese, and Korean (CJK) text: (i) single analyzer approach that uses the standard analyzer, (ii) multi-fields approach that uses multiple analyzers and multi-fields, and (iii) language detector-based approach that uses multiple analyzers, multiple indices, and a language detector.

## II. RELATED WORKS

Search engines are programs that search using data sets. A special program called robot automatically collects and


Received 14 November 2019, Revised 23 March 2020, Accepted 24 March 2020

\*Corresponding Author Young-bok Cho (E-mail: ybcho@dju.ac.kr, Tel: +82-42-280-2406)

Assistant Professor, Department of Information Security, Daejeon University, Daejeon 34520 Korea

Open Access <https://doi.org/10.6109/jicce.2020.18.1.33>

print ISSN: 2234-8255 online ISSN: 2234-8883

 This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Copyright © The Korea Institute of Information and Communication Engineering

indexes data on the web and provides a search form for the user to find the intended information. Since the inception of the web, search engines have been one of the most active tools, with Google being the current major search engine. Additionally, the growing volume of information and the speed of expansion of the web can be attributed to the capabilities of current search engines, e.g., continuous production of big data in the Internet of Things environment, computing data accumulated for decades, and data production beyond imagination through smart devices and social network service [2].

Internet search engines can be classified depending on the operation method or search target as follows [3-8]:

**General search engine (GSE):** GSE searches across all topics of interest. Examples of GSE are Google, Yahoo, and Bing. Such a search engine, which is most familiar to general users, collects and indexes maximum possible web pages related to the topics of interest of most users, divides them according to the topics, and provides results according to the user's search terms.

**Vertical search engine (VSE):** VSE, which is also known as domain-specific search engine focuses on specific rather than all areas of interest.

**Meta search engine (MSE):** MSE is a search engine that queries user input keywords on various search sites, retrieves the search results from the actual search site, and displays the results. It does not have its own database; however, it supports various information searches. Consequently, the user can obtain consolidated search results from multiple search sites without actually having to navigate to them.

GSE, which is the most popular type of search engine has grown rapidly owing to its ease of use, smooth accessibility, and universality of data. However, in the current data big-bang situation, the volume of information and the expansion rate of the web are increasing in real time. Therefore, it is not sufficient to utilize a universal search engine that presents regularly crawled and indexed search results.

### III. CJK TEXT SEARCH WITH LANGUAGE DETECTOR USING ELASTICSEARCH

#### A. Search Engine Design

Fig. 1 presents the design of a search engine.

##### 1) TF-IDF

Search engines return a list of matching documents sorted by scores. One of the most popular algorithms to determine the score of a matching document is TF with IDF (TF-IDF) [9]. It is known that Google uses TD-IDF or a similar algorithm. Lucene and Elasticsearch use BM25 [10], which is based on TF-IDF. TF-IDF calculates scores based on TF and

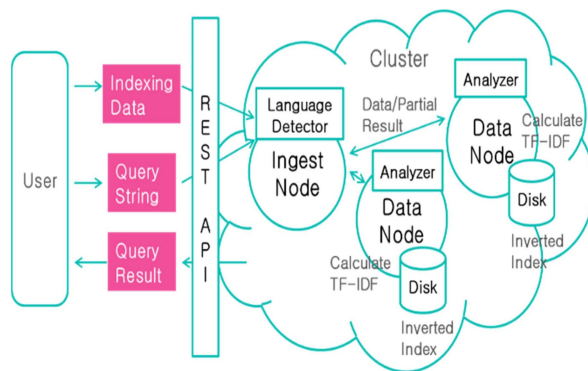


Fig. 1. Architectural design of search engine.

IDF. TF represents how often a term appears in the field and IDF represents how often each term appears in the index. Therefore, documents that appear often in the field and less often in the index obtain higher scores and appear at the top of the result list. Table 1 describes the types of search engines.

Table 1. Types of search engines

Classification	Description
Single Analyzer	One text unit comprises one field One mapping analyzer must be specified
Multi-Field	One text comprises several fields Requires multiple analyzers and causes wastage of time and space
Detector-based	Constructed using an index for each text Analysis time and memory wastage can be mitigated by using an index

#### 2) Text Analysis

Text analysis is an automated process to extract information (e.g., token) from a block of text [11, 12, 13, 14]. For example, Google provides the natural language API to extract sentiment, entity, etc., and Microsoft Azure provides the text analytics API to extract sentiment, key phrases, etc. Elasticsearch provides language-specific analyzers to extract tokens by considering vocabulary and syntax of each language.

#### 3) Language Detection

Language detection [15, 16] can improve multilingual text search because it allows the selection of correct analyzers to process the text. It is based on statistical approaches and character encoding detection. CJK languages use different alphabets, and the language detection works satisfactorily with them.

#### B. Single Analyzer Approach

The first and simplest approach to search and index CJK text is the single analyzer approach. It indexes and searches

the text using the standard analyzer. The standard analyzer, which is the default analyzer divides text into terms on word boundaries, removes punctuation and lowercase terms [2]. If we do not specify an analyzer in the mapping, the standard analyzer is used. Table 2 presents an example of the usage of the standard analyzer.

**Table 2.** Standard analyzer examples

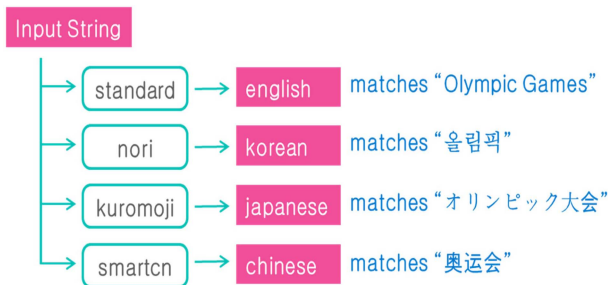
Examples
<pre>PUT test/_doc/1 {   "message": "평창은 제123차 IOC 총회에서 2018년 동계올림픽대회의 개최지로 선정되었습니다." }</pre>
<pre>PUT test/_doc/2 {   "message":   "平昌は123回IOC総会で2018年冬季オリンピック大会の開催地 に選ばれました。"</pre>
<pre>PUT test/_doc/3 {   "message":   "平昌于第123届国际奥委会全会上被选定为2018年冬季奥运会 的主办地。"</pre>
<pre>PUT test/_doc/4 {   "message": "PyeongChang was selected as the host city of the 2018 Olympic Winter Games at the 123rd IOC Session."</pre>

However, the standard analyzer does not separate postpositions (e.g., “의”) from nouns (e.g., “동계올림픽대회”), and it produces tokens such as “평창은” “제 123 차” “ioc” “총회 에서” “2018년” “동계올림픽대회의” “개최지로” and “선 정되었습니다.” We cannot find the above document when we search with “올림픽대회.”

### C. Multi-fields Approach

Elasticsearch’s “multi-fields” feature is used to index the same text in different ways. It can be used to analyze the same text using multiple language-specific analyzers as shown in Fig. 2.

The analyzers nori [3], kuromoji [4], and smartcn [5] are



**Fig. 2.** Multi-fields approach

required to be installed to define a mapping with multi-fields using the analyzers as follows:

```

PUT /test{
  "mappings": {
    "_doc": {
      "properties": {
        "message": {
          "type": "text",
          "fields": {
            "korean": {
              "analyzer": "nori",
              "type": "text"
            },
            "japanese": {
              "analyzer": "kuromoji",
              "type": "text"
            },
            "chinese": {
              "analyzer": "smartcn",
              "type": "text"
            }
          }
        }
      }
    }
  }
}
  
```

For example, a given text is indexed as follows:

```

PUT test/_doc/1 {
  "message": "평창은 제123차 IOC 총회에서 2018년
동계올림픽대회의 개최지로 선정되었습니다."
}
  
```

The text is analyzed by all four analyzers: standard, nori, kuromoji, and smartcn for the “message” field, and the following tokens are generated correspondingly:

- “message” field: “평창은”, “제123차”, “ioc”, “총 회에서”, “2018년”, “동계올림픽대회의”, “개최지 로”, “선정되었습니다”
- “korean” field: “평창”, “123”, “차”, “ioc”, “총회”, “2018”, “년”, “동계”, “올림픽”, “대회”, “개최”, “지”, “선정”
- “japanese” field: “123”, “ioc”, “2018”
- “chinese” field: “”평”, “창”, “은”, “제”, “123”, “차”, “ioc”, “총”, “회”, “에”, “서”, “2018”, “년”, “동”, “계”, “올”, “림”, “픽”, “대”, “회”, “의”, “개”, “최”, “지”, “로”, “선”, “정”, “되”, “었”, “습”, “니”, “다”

The expected documents can now be searched with the terms: “올림픽대회,” “オリンピック大会,” and “奥运会,” respectively.

However, this approach has several drawbacks, such as

- 1) It analyzes and stores a message four times each, thereby wasting time and storage.
- 2) Sometimes, it returns unexpected search results, e.g.,
  - a Japanese document “平昌は123回IOC総会で2018年冬季オリンピック大会の開催地に選ばれました。” and a Chinese document “平昌于第123届国际奥委会全会上被选定为2018年冬季奥运会的主办地。”

#### D. Abbreviations

The following tokens were used during analysis using the standard analyzer:

- “平, 昌, は, 123, 回, ioc, 総, 会, 2018, 年, 冬, 季, オリンピック, 大, 会, ...” and
- “平, 昌, 于, 第, 123, 届, 国, 际, 奥, 委, 会, 全, 会, 上, 被, 选, 定, 为, 2018, 年, ...”

They are generated and stored in the message field of each document. When we search with the search keywords: “オリンピック大会” and “奥运会,” the search keywords are also analyzed by the standard analyzer. They match the token “会” in the message fields of both the Chinese and Japanese documents. Consequently, both are returned contrary to the expectation.

Furthermore, the borrowed word: “IOC” appears in the Korean, English, and Japanese documents, which affects the IDF in the entire index [6] and reduces relevance on such borrowed words.

### IV. LANGUAGE DETECTOR-BASED APPROACH

To avoid the drawbacks of the multi-fields approach, we can use a language detector and an ingest pipeline [7] to index the CJK messages into corresponding language-specific indices as shown in Fig. 3.

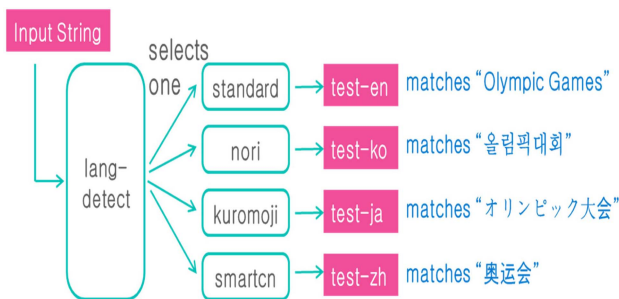


Fig. 3. Language detector-based approach.

We can install Elasticsearch langdetect ingest processor [8] and define the following pipeline and mapping:

```

PUT /_ingest/pipeline/langdetect-pipeline {
  "description": "A pipeline to index multi-language text",
  "processors": [ {
    "langdetect": {
      "field": "message",
      "target_field": "language"
    }
  },
  {
    "set": {
      "field": "_index",
      "value": "test-{{language}}"
    }
  }
]
}

PUT test-ko {
  "mappings": {
    "docs": {
      "properties": {
        "message": {
          "type": "text",
          "analyzer": "nori"
        }
      }
    }
  }
}

PUT test-ja {
  "mappings": {
    "docs": {
      "properties": {
        "message": {
          "type": "text",
          "analyzer": "kuromoji"
        }
      }
    }
  }
}

PUT test-zh {
  "mappings": {
    "docs": {
      "properties": {
        "message": {
          "type": "text",
          "analyzer": "smarten"
        }
      }
    }
  }
}
  
```

```

PUT test-en {
  "mappings": {
    "docs": {
      "properties": {
        "message": {
          "type": "text",
          "analyzer": "standard"
        }
      }
    }
  }
}

PUT test-original/_doc/1?pipeline=langdetect-pipeline {
  "message": "평창은 제123차 IOC 총회에서 2018년 동계올림픽대회 개최지로 선정되었습니다."
}

PUT test-original/_doc/2?pipeline=langdetect-pipeline {
  "message": "平昌は123回IOC総会で2018年冬季オリンピック大会の開催地に選ばれました。”
}

PUT test-original/_doc/3?pipeline=langdetect-pipeline {
  "message": "平昌于第123届国际奥委会全会上被选定为2018年冬季奥运会的举办地。”
}

PUT test-original/_doc/4?pipeline=langdetect-pipeline {
  "message": "PyeongChang was selected as the host city of the 2018 Olympic Winter Games at the 123rd IOC Session.”
}

```

All search APIs can be applied across multiple indices, and the indices can be searched as if they are a single index as follows:

```

GET test-*/_search {
  "query": {
    "match": {
      "message": "올림픽대회"
    }
  }
}

```

It can be observed that the language detector-based approach has a number of advantages over the single analyzer and multi-fields approaches: it provides improved relevance because the borrowed words in the text of other languages are indexed in separate indices. It returns only Japanese documents when we search with “オリンピック大会” and only Chinese documents when we search with “奥运会” because each message is analyzed only once using the analyzer for the

detected language. It saves analysis time and storage by detecting the language, applying only the relevant analyzer, and storing only once.

## V. DISCUSSION AND CONCLUSIONS

We searched and compared the string “평창 동계올림픽” by keyword using each search and analysis engine. Fig. 4 illustrates the results of the experiments on the search analysis engine operation time, search end rate, and conversion rate on the detail page. The experimental results indicated that the proposed method, the detector-based engine, is approximately 20 s faster.

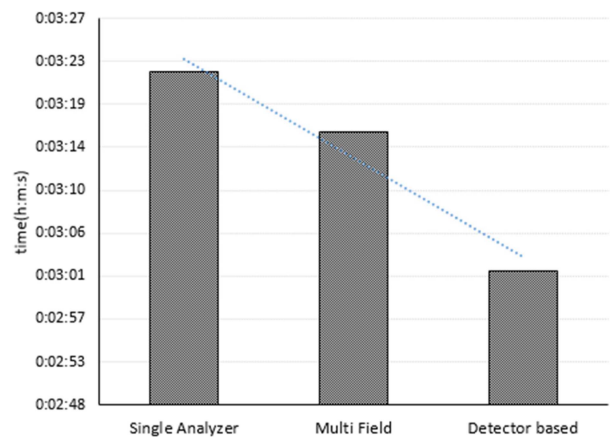


Fig. 4. Result of average search time.

Fig. 5 compares the time required to display the results post search operation; the proposed method can closing search up to 8.2% faster. Additionally, it was confirmed through experiments that conversion to an average of 8.3% faster while switching to the last page of the search.

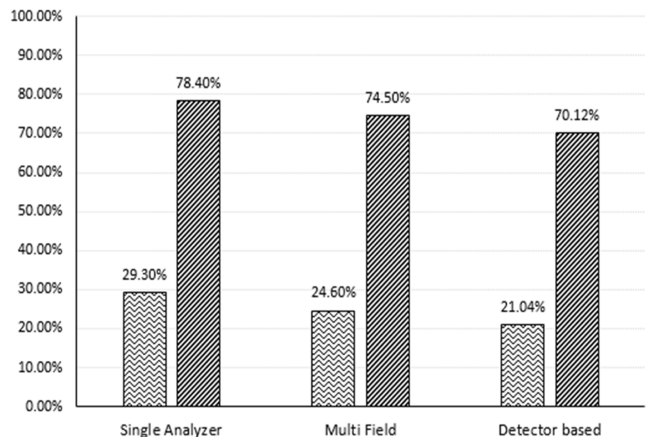


Fig. 5. Search termination rate and conversion rate on detail page

The comparison of the three approaches reveals the following:

- 1) Language-specific analyzers are required to index and search CJK text.
- 2) The language detector-based approach saves analysis time and storage because each text is analyzed and stored only once.
- 3) The language detector-based approach provides improved relevance and search results.

In future, we plan to extend this work to handle synonyms and mixed-language text.

## ACKNOWLEDGEMENTS

This research was supported by the Daejeon University Research Grants (2019).

## REFERENCES

- [1] ReadonlyREST, The Heart of the Elastic Stack [Internet], Available: <https://www.elastic.co/products/elasticsearch>.
- [2] Elastic Research Center, Analyzers [Internet], Available: <https://www.elastic.co/guide/en/elasticsearch/reference/current/analysis-analyzers.html>.
- [3] Elastic Research Center, Korean (nori) Analysis plugin [Internet], Available: <https://www.elastic.co/guide/en/elasticsearch/plugins/6.x/analysis-nori.html>.
- [4] C. Breiting, B. Gipp, and S. Langer, "Research-paper recommender systems: a literature survey," *International Journal on Digital Libraries*, vol. 17, no. 4, pp. 305-338, 2015. DOI: 10.1007/s00799-015-0156-0.
- [5] S. S. Byun, "Measurement Allocation by Shapley Value in Wireless Sensor Networks," *Journal of Information and Communication Convergence Engineering*, vol. 16, no. 1, pp. 38-42, 2018. DOI: 10.6109/jicce.2018.16.1.38.
- [6] Kuromoji, Japanese Analysis Plugin [Internet], Available: <https://www.elastic.co/guide/en/elasticsearch/plugins/6.x/analysis-kuromoji.html>.
- [7] Elastic Research Center, Smart Chinese Analysis plugin [Internet], Available: <https://www.elastic.co/guide/en/elasticsearch/plugins/6.x/analysis-smartcn.html>.
- [8] Elastic Research Center, What is Relevance? [Internet], Available: <https://www.elastic.co/guide/en/elasticsearch/guide/current/relevance-intro.html>.
- [9] Elastic Research Center, Ingest Node [Internet], Available: <https://www.elastic.co/guide/en/elasticsearch/reference/6.6/ingest.html>.
- [10] Elastic Research Center, Elasticsearch Langdetect Ingest Processor [Internet], Available: <https://github.com/spinscale/elasticsearch-ingest-langdetect>.
- [11] B. J. Jansen and S. Rieh, "The Seventeen Theoretical Constructs of Information Searching and Information Retrieval," *The Journal of the American Society for Information Sciences and Technology*, vol. 61, no. 8, pp. 1517-1534, 2010. DOI:10.1002/asi.v61:8.
- [12] F. L. Jill, "Adaptive Parsing: Self-Extending Natural Language Interfaces," *International Journal of Computational Linguistics*, vol. 18, no. 3, 1992. DOI: 10.1007/978-1-4615-3622-2.
- [13] R. Radim and K. Milan, "Language Identification on the Web: Extending the Dictionary Method," *Lecture Notes in Computer Science*, vol. 5449, DOI: [https://doi.org/10.1007/978-3-642-00382-0\\_29](https://doi.org/10.1007/978-3-642-00382-0_29).
- [14] R. Cilibrasi and M. B. Paul, "Clustering by compression," *International Journal of IEEE Transactions on Information Theory*, vol. 51, no. 4, pp. 1523-1545, 2005. DOI: 10.1109/TIT.2005.844059.
- [15] R. S. Bhandari and A. Bansal, "Impact of Search Engine Optimization as a Marketing Tool," *Jindal Journal of business Research*, March 2018,. [Online] Available: <https://doi.org/10.1177/2278682117754016>.
- [16] K. Cao, J. Lee, and H. Jung, "Keyword Analysis Based Document Compression System," *Journal of Information and Communication Convergence Engineering*, vol. 16, no. 1, pp. 48-51, 2018. DOI: 10.6109/jicce.2018.16.1.48.



**Ki-Ju Kim**

Ki-Ju Kim is a Senior Support Engineer at Elastic. He received his B.S. and M.S. degrees in the department of computer science and engineering, Pohang University of Science and Technology, South Korea in 1996 and 1998, respectively. He has conducted several official Elasticsearch Engineer trainings and spoken at several software developer conferences such as JavaOne and Elastic{ON} Tour.



**Young-Bok Cho.**

Young-Bok Cho is an Associate Professor at the Department of Computer & Information Security, Daejeon University, Daejeon, South Korea. She received her M.S. and Ph.D. degrees in computer science from Chungbuk National University, in 2006 and 2012, respectively. She has been involved in few national researches and has been an expert adviser of SME in Korea to follow-up research projects and an evaluator of Korea R&D a national research proposals. She is currently working in the department of information security, Daejeon University. Her research interests include network security, medical information security, and medical image processing.