Using Catalyst CR
To Search The Japanese

By Jim Eidelman
Catalyst Consulting
Catalyst Repository Systems
Using Catalyst CR to Search Japanese

Catalyst CR is built using the FAST search engine. FAST is a highly-sophisticated, grid-based search engine developed by a Norwegian public company, which has since been acquired by Microsoft.

Introduction to Search

Aside from speed and scalability, one of the most attractive features about the FAST search engine is that it supports search in over 70 languages. Most important for purposes of this paper, FAST can search the CJK languages: Chinese, Japanese and Korean.

CJK languages are more difficult to index than Western languages because they don’t use punctuation or spaces to define word boundaries. Word boundaries are critical for computerized search because search engines create large indexes based on the words in the underlying files being searched.

When users run searches, the search engine goes to the index to find where words are rather than run through each of the individual documents. This practice of creating and using a “word” index is the reason modern search engines can bring back results quickly, even for searches involving millions of pages of documents. It is the method followed by Google, Westlaw and Lexis, among many others.

Understanding Tokens

In order to create an index, the indexer must be able to recognize word boundaries. In technical terms, this is called “tokenization,” and the individual word units are called tokens. An index consists of all the tokens the indexer found when it indexed through the documents. We call these tokens rather than words because the indexer doesn’t make a distinction between a real word and a collection of characters that look like a word but aren’t really a word, like a misspelling.

Here are some sample English words and the number of tokens for each:

<table>
<thead>
<tr>
<th>Words in document</th>
<th>Tokens</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer</td>
<td>1</td>
</tr>
<tr>
<td>USA</td>
<td>1</td>
</tr>
<tr>
<td>U.S.A.</td>
<td>3</td>
</tr>
<tr>
<td>Mc Intosh</td>
<td>2</td>
</tr>
</tbody>
</table>
In English, understanding tokenization doesn't matter much for straight searching, such as searching for documents that simply CONTAIN a word or a phrase, even when you use wild cards to find variations.

It is important if you are using the NEAR proximity locator, since an extra punctuation mark or initial can add to the number of words (tokens) that they are apart.

**Japanese Tokenization**

Unlike English, Japanese presents challenging tokenization issues. This is because of a combination of factors:

1. There are generally no spaces between words, so how does the computer know where to break between words? Here is an example of the word “selling” in a Japanese sentence, with “selling” highlighted in yellow:

   既に大量に同規格の物を作って販売しております

2. Many Japanese words are compound words, with a combination of symbols for smaller words. For example, the word for “selling” is 販売しております. This is actually a combination of three words, meaning “to sell,” “to do,” and “to be,” like this: 販売(to sell)して(to do)おります(to be).

The software needs to decide how to tokenize a continuous stream of Japanese characters, figuring out where to put spaces between the words and also “de-compounding” the compound words into sub-words.

The consequences are that if the search expression is not tokenized the same way the text of the document is tokenized, then the search will find zero documents that satisfy the search request.

The tokenizer CR uses will break the word for “selling” down into the three parts with a space in between each, with each part treated like a separate word that is part of a three-word phrase. 販売しております. If the computer were to search for 販売しております, without the spaces, it wouldn't find 販売しております. Fortunately, CR handles this automatically.

In addition, if you are using the NEAR connector in a proximity search, the word “selling” looks to the computer like three words, not one.

Also note that a consequence of this is that searching for just one of the sub-words will find the document. Thus, in this case, if you search for just 販売, which means “to sell,” you will find it as a part of the word for “selling.”
Tokenizing a foreign language document

FAST indexes each document when it is loaded on the CR system. As part of the indexing process, FAST looks at the document and its Language Identifier figures out what the primary language is, and applies the appropriate tokenization method. This is done using a mathematical model.

For Japanese, Chinese and Korean, the system will automatically determine the ONE CJK language that is primary, and apply tokenization for that language, in addition to tokenizing and indexing the English or other western-style languages. So a document can have a mix of Japanese and English, and FAST will handle it correctly.

If there is a mix of Japanese and Chinese, the system will figure out which one is primary, and will tokenize the entire document for that language, but it can't do both in the same document.

The FAST indexer will also give the document a code that is stored in the database for what primary language and what secondary languages are used in the document, which makes it easy to group, batch, search, and review documents by language. This secondary languages field can be used as a “workaround” to find documents that have, say, Japanese as the primary language and Chinese as a secondary language, so that those documents can be reviewed even if the search engine did not find the Chinese.

The FAST search engine we use has licensed the CJK tokenization software from Basis Technology, the same suite of software that is used by Google, Amazon, and other major web sites and software houses. It uses a combination of many sophisticated methods to decide where to break words, and is very smart in the way it does that. It has a dictionary of millions of Japanese words and their component parts, and it handles both the segmentation to put spaces between words and the “de-compounding” to put spaces between the individual sub-words of compound words, as in the “selling” example above.

Tokenizing a foreign language search

The same tokenizer is used to tokenize both the search and the document, so the same words will be segmented in the same places, and the system will find them. In some cases, however, the language being searched can be ambiguous. As an example, certain characters might be used in both the Japanese and Chinese languages. How is FAST to know which language is being searched?

To help FAST answer that question, we provide a dropdown menu that allows the user to tell FAST which language is being used in the search. It looks like this:
Sticking with the same example as above, if you type the Japanese for “Selling” in the search window and select the Japanese language for the search, the system will find it. This is because it will automatically be broken down into its three component sub-words with spaces in between in both the search and the document, so there will be a match.

販売しております

If you forget to select Japanese for the search and leave it set for “Generic,” the document might not be found.

The search utility that Catalyst uses to run hundreds or even thousands of searches sequentially also has this feature built in, so the spreadsheets used for importing the searches have a column for language, and for Japanese, it will have a “ja.”

**Multi-Language Searches**

In some cases a user will want to search across multiple languages in a single search. FAST allows you to do this with certain caveats.

First, you can search Japanese and English in a single statement:

```
    doctext co (クリスマス or Christmas)
```

That will find any document that has “Christmas” in Japanese or English.

You can do the same for Chinese or Korean. The tokenizer will always handle English searches in conjunction with one or the other CJK languages.
The FAST search engine may not properly tokenize searches involving two or more of the CJK languages in the same document. Thus, a search using Japanese and Chinese may bring back incorrect results from one language or the other. Contact Catalyst Consulting for assistance in working through the best ways to handle documents that have two or more of the CJK languages in the same document.

You can combine any of the Western languages. These all follow common tokenization rules.

**How do NEAR proximity searches work in Japanese?**

The NEAR/N command works the same in English and Japanese in the sense that it finds words or tokens separated by no more than the number stated.

```
(Christmas) near/25 (Product)
```

```
(クリスマス) near/25 (商品)
```

These two searches are equivalent in that they search for Christmas separated by no more than 25 words or tokens from Product in the appropriate language. However, note that it takes a higher number in Japanese for this to work. That is because tokenized Japanese words have more white spaces separating the tokens than the English equivalent.

Besides just separating “words,” there are several different other times the tokenizer adds spaces:

1. In the “selling” example, the word “selling” in Japanese actually is tokenized by the computer as three sub-words.
2. Japanese also adds characters called “particles” to explain the role of the word or phrase in the sentence (subject, object, etc.), and each particle is also separated by a space and treated as a “word” or token.

Thus, the key to proximity in Japanese searching is to expand the number of “words” in your proximity range. You may want to experiment, but we have found that it is generally safe to double the number of words apart if the search is in Japanese.

**Honorifics**

The Japanese use “honorifics” to indicate the difference in formality or status in the relationship between the speaker/writer and the listener/reader. For example, an employee will use honorifics meaning “most sincere” or “most honorable” when speaking to his or her boss, and the boss may use language that does the
opposite when giving instructions to an employee. These are typically different groups of characters that are added to the “verb” part of a word.

To stick with the example we have been using,

販売しております -- “Selling” in the honorific form, with the formal “to be” in yellow, used for a manager to an employee, or a company to its customers.

販売しています -- “Selling” in normal form, as two co-workers would write to each other.

What is important for searching is that the FAST tokenizer will insert a space before the honorific or normal form of “to be” that is part of the word. Therefore you will find both forms if you search for the first part of the word, 販売 or 販売して, but you will not find both if you search for the whole word “selling.” Again, since there is already a space hidden there, putting a wild card (販売して*) is unnecessary, although it won’t do any harm.

About Japanese Scripts (Alphabets)

There are four different scripts (alphabets) that you may encounter in the Japanese language, and they may be inter-mixed in a continuous stream of text:

• Kanji;
• Hiragana;
• Katakana; and
• Romaji (English or other western style).

Inter-mixing of these scripts doesn’t matter to the tokenization. The tokenizer will put spaces in the appropriate places, no matter which of the four scripts are used.

Nonetheless, you may want to search for alternatives—separated by “OR” in each search—if you think the documents may contain the same words in different scripts.

About Using Parentheses

Parentheses in English and in the Japanese font look the same to the human eye, but the computer treats them differently. A Japanese parenthesis is treated by the computer as a Japanese character to search for. English parentheses are used to group parts of a search as part of the search syntax, as in
doctext co (クリスマス or 商品)

You should NOT use Japanese font parentheses in your searches where it is used as part of the search command syntax. You will either get syntax errors or unintended results (often zero hits) if you do.

A Linguist’s View

Japanese and English take different approaches and have different rules. Japanese is a “pro-drop language,” which means the user can omit the subject or object if it is obvious from the context. In fact, there aren’t many rules in Japanese, so as long as the reader understands, there is a lot of flexibility in Japanese that we don’t have in English. Japanese is also “topic prominent,” which means that the topic will often come first, then the subject and object, like “As for elephants, their noses are long.”

There are dozens of other aspects in which Japanese searches will be different than English searches, so there is no substitute for having an experienced Japanese linguist creating the searches. Catalyst Consulting can assist the Japanese attorneys or other linguists with the best way to formulate and execute the searches.

To Learn More About Complex Search Issues

Contact Jim Eidelman to discuss or learn more about Japanese search issues at:

Jim Eidelman
Senior Consultant with Catalyst Consulting
Certified Catalyst Search Specialist
jeidelman@catalystsecure.com
303-834-0856

Jim can discuss the issues with you, and we can also provide a professional Japanese linguist who can assist you in formulating queries.

Jim has authored hundreds of articles on legal technology, and was co-editor with John Tredennick of Winning with Computers, the ABA’s best-selling book on the use of computers in litigation. Jim is Catalyst’s lead consultant on strategic searching for culling, relevance/issue searching, privilege searches, and use of the Catalytics™ suite of analytical tools. He has worked with clients to build and run sets of thousands of searches and to create predictive ranking of the results.

Catalyst can also provide reviewers and search consultants who not only speak Japanese but who are also specialists in technical areas for patent, medical, or other technical cases.