



BASIS
TECHNOLOGY



Building a Multilingual Search Engine with Apache Lucene

Your company has a web application, be it an online catalog or a large e-commerce site. Your boss has asked you to add a search box so visitors can find what they need. Where do you start?

You could add a small piece of HTML code to let Google search your site. However, this works only if your dynamic content is reachable by following static links. If you're running a small bookstore with only 1,000 titles that don't change much over time this might be an option. But once your store grows to 100,000 titles, it can become unmanageable to maintain in static pages.



Introducing Solr: Enterprise-Ready Open Source Search

Apache [Solr](#) (pronounced like “solar”) to the rescue. Solr is an open-source enterprise search server that runs on top of the open-source search toolkit Apache [Lucene](#). It provides the high quality search experience that online visitors expect, while offering open source flexibility and affordability. Solr comes with extensive search functionality (including drill-down faceted searching), and is easy to deploy and maintain.

Solr and Lucene are Apache projects that have become very popular for web site search. You can learn more about them in the IBM Developer Work's web site article [Search smarter with Apache Solr](#), which has step-by-step instructions introducing Solr.



Sprechen Sie Deutsch? 日本語話しますか？

Okay, you decided to implement search using Solr, so the problem has been solved, right? Well, maybe, if your site only needs to support English. But if your customers communicate in other languages, you are likely to have problems that Solr doesn't solve out of the box.

Consider what happens when customers try searching in German. Solr's knowledge of language issues is contained in a component called the Standard Tokenizer. The Standard Tokenizer handles German text to some degree – it won't crash, and German words separated by spaces will be indexed. But language features like compounds and plural forms will be ignored and your German customers will be discouraged when they can't search like they do on the big web search engines.

- Compound nouns: German is known for really-long-words-by-putting-words-together-without-spaces like Fernsprechteilnehmerverzeichnis (remote participant directory). If only the whole word is in the index, a search for the equivalent of “remote participant” will not find it
- Plurals: If you search for “Garten” (garden), Solr will miss documents where only the plural form “Gärten” is used.

Solr does have filters (token transformation components) that help with part of this. The GermanStemFilter or SnowballFilter can fix the plural issue, but there are no standard components that help with compounds and some other German linguistic issues.

Next you're asked to index some Japanese text. Japanese, like Chinese, is written without spaces to separate words. How many words do you see here?

日本語話しますか？ (*Do you speak Japanese?*)

The correct answer is three words + one punctuation mark: 日本語, 話します, か, ？

Solr includes a CJKTokenizer that is supposed to handle Chinese, Japanese, and Korean as the name CJK implies (CJK = Chinese, Japanese, Korean). But does it? Instead of splitting text into real words, CJKTokenizer splits them into pairs of neighboring CJK characters.

The above sentence would result in these words (CJKTokenizer silently removes punctuation):

日本, 本語, 語話, 話し, しま, ます, すか

Not one of these tokens is a word from the original sentence!

This technique is called *bi-gram indexing*. But it's cheating, as you can see. It works sometimes but not always. For example, a search for 京都 (Kyoto, the old capital of Japan) will end up finding documents about 東京都 (Tokyo-to, the current capital of Japan). (Read the whitepaper [N-Gram vs. Morphological Analysis](#) for more details.)

Finally, you'd better hope your boss doesn't ask you to add Arabic content to your website — Solr doesn't come with an analyzer for that language.

Rosette Linguistics Platform and Lucene: A Perfect Combination

Basis Technology's [Rosette Linguistic Platform](#) (RLP) enables many different kinds of text analysis, including tokenization and normalization at different levels, for many languages, from popular languages like English, Spanish, and Chinese to less common languages like Farsi and Urdu. The software provides linguistic intelligence for many commercial and government applications, and brings accuracy to multilingual searching far beyond Solr or Lucene alone. A new RLP component, *Rosette Linguistic Platform For Lucene*, is now available that makes applying RLP's language capabilities to Lucene and Solr applications easier than ever:

- For Lucene customers, install RLP and RLP For Lucene, and just modify your code to use one of the analyzers provided with the Lucene package. (An analyzer in Lucene and Solr combines a tokenizer and zero or more filters.) If none of the stock analyzers is suitable, you can copy and modify the RLP For Lucene source code to meet your needs.
- Solr customers have an even easier time. No code modification to the Solr servers is required. Simply modify schema.xml (one of the Solr configuration files) and your application starts speaking the language(s) you need for immediate language-aware searching.

With RLP and Lucene or Solr, you can build the global-ready search server in a snap.

Still Not Convinced?

Try it for yourself. Request an evaluation kit by contacting Basis technology at info@basistech.com. Tell us what languages you need to work with and your computer platform (Windows XP, Linux Redhat, etc.).

Need More Help?

Don't even want to deal with Solr or Lucene? You just want to have a solution? Basis Technology can help build an integrated search solution using Lucene and Solr or a variety of commercial search solutions from our partner companies.

To learn more, please contact us at info@basistech.com.

