FileBasedSpellChecker

Intro

The FileBasedSpellChecker uses a flat file to build a Lucene contrib/spellcheck index. There is currently no frequency information used in calculating spelling suggestions, so the SpellCheckComponent will not be able to provide "only more popular" or other frequency based information. As a workaround, it isn't all that hard to create an index from a file and have it weight the terms.

Format

The file format is one word per line, as in

```
pizza
history
junk
```

Hints

The FileBasedSpellChecker is very similar in operation to v3's primary IndexBasedSpellChecker, both of which require a separate Lucene index to be built; this can be a bit confusing to those who've only ever used v4's DirectSolrSpellChecker which doesn't have that requirements. In particular, you still need to index the dictionary file once by issuing a search with **&spellcheck.build=true** on the end of the URL; if you system doesn't update that dictionary file, then this only needs to be done once. This manual step may be required even if your configuration sets build=true and reload=true.

In the default solrconfig.xml there's a sample commented out configuration, **<str name="name">file</str>**, that can be used as a template. The name "file" is arbitrary and you can have several file based spellcheckers, pointing to different flat files, with different names. Elsewhere in solrconfig you'd reference the named **file** configuration with <str name="spellcheck.dictionary">file</str>.

Make sure the "field" parameter points to a valid field name in your schema.xml file.

If you're going to define more than one instance of FileBasedSpellChecker, for example to have two different lists of terms e.g. medical terms and electrical terms, make sure you give each one a different value for the **spellcheckIndexDir**.

Using a full English dictionary is possible and may seem like a good idea at first, but a full dictionary may have words that don't appear in your main Solr index, so suggesting those words to users would typically be a bad idea because they'd get zero hits if they tried to search with them. If you're using the collator then it should filter out corrections that have zero hits, so might avoid the problem, although it brings into question the value of having brought in those extra suggestions in the first place.

If you do have some use case that would benefit from full-language spelling suggestions (perhaps as an education tool, vs. creating clickable links), there are numerous open source dictionaries available for different languages. A google search for **open source dictionaries** should help; both **GNU ASpell** and **OpenOffice** have open source dictionaries, however they are often in various nested and compressed formats so some preprocessing will be needed to get them into Solr's plain text format. Indexing of the entire OpenOffice English dictionary took less than a minute on a 2012 MacBook pro.