# SpellCheckingAnalysis

## Introduction

Analysis is a very important factor in spell checking. Stemming and other techniques that change tokens is not recommended since it will result in giving stems as suggestions. Instead, you should use a very minimal tokenization/analysis process like the StandardAnalyzer or even the WhitespaceTokenizer plus a simple lower casing filter and a filter that removes apostrophes and the like. As with most things in search, there are always tradeoffs and you should evaluate the results in your application.

That being said, a common configuration for spell checking is:

```
<fieldType name="textSpell" class="solr.TextField" positionIncrementGap="100" omitNorms="true">
  <analyzer type="index">
    <tokenizer class="solr.StandardTokenizerFactory"/>
    <filter class="solr.StopFilterFactory" ignoreCase="true" words="stopwords.txt"/>
    <filter class="solr.LowerCaseFilterFactory"/>
    <filter class="solr.StandardFilterFactory"/>
  </analyzer>
  <analyzer type="query">
    <tokenizer class="solr.StandardTokenizerFactory"/>
    <filter class="solr.SynonymFilterFactory" synonyms="synonyms.txt" ignoreCase="true" expand="true"/>
    <filter class="solr.StopFilterFactory" ignoreCase="true" words="stopwords.txt"/>
    <filter class="solr.LowerCaseFilterFactory"/>
    <filter class="solr.StandardFilterFactory"/>
  </analyzer>
</fieldType>
```

Furthermore, on the field that will get this type, use `omitTermFreqAndPositions="true"` to save a little space and time during indexing.

Use a <copyField> to divert your main text fields to the spell field and then configure your spell checker to use the "spell" field to derive the spelling index.