Deduplication

Document Duplication Detection

⚠️ Solr1.4

- Document Duplication Detection
- Overview
  - Goals
  - Design
- Notes
- Configuration
  - solrconfig.xml
  - schema.xml
    - Note
  - Settings

Overview

Preventing duplicate or near duplicate documents from entering an index or tagging documents with a signature/fingerprint for duplicate field collapsing can be efficiently achieved with a low collision or fuzzy hash algorithm. Solr should natively support deduplication techniques of this type and allow for the easy addition of new hash/signature implementations.

Goals

- Efficient, hash based exact/near document duplication detection and blocking.
- Allow for both duplicate collapsing in search results as well as deduplication on adding a document.

Design

Signature

A class capable of generating a signature String from the concatenation of a group of specified document fields.

```
public abstract class Signature {
  public void init(SolrParams nl) {
  }

  public abstract String calculate(String content);
}
```

Implementations:

MD5Signature	128 bit hash used for exact duplicate detection.
Lookup3Signature	64 bit hash used for exact duplicate detection, much faster than MD5 and smaller to index
TextProfileSignature	Fuzzy hashing implementation from nutch for near duplicate detection. Its tunable but works best on longer text.

There are other more sophisticated algorithms for fuzzy/near hashing that could be added later.

Notes

Adding in the dedupe process will change the allowDups setting so that it applies to an update Term (with field signatureField in this case) rather than the unique field Term (of course the signatureField could be the unique field, but generally you want the unique field to be unique)

When a document is added, a signature will automatically be generated and attached to the document in the specified signatureField.

Configuration

solrconfig.xml

The SignatureUpdateProcessorFactory has to be registered in the solrconfig.xml as part of the UpdateRequest Chain:

Accepting all defaults:

```
<updateRequestProcessorChain name="dedupe">
  <processor
    class="org.apache.solr.update.processor.SignatureUpdateProcessorFactory">
  </processor>
  <processor class="solr.RunUpdateProcessorFactory" />
</updateRequestProcessorChain>
```

Example settings:

```
<!-- An example dedup update processor that creates the "id" field on the fly
     based on the hash code of some other fields.  This example has overwriteDupes
     set to false since we are using the id field as the signatureField and Solr
     will maintain uniqueness based on that anyway. -->
<updateRequestProcessorChain name="dedupe">
  <processor class="org.apache.solr.update.processor.SignatureUpdateProcessorFactory">
    <bool name="enabled">true</bool>
    <bool name="overwriteDupes">false</bool>
    <str name="signatureField">id</str>
    <str name="fields">name,features,cat</str>
    <str name="signatureClass">org.apache.solr.update.processor.Lookup3Signature</str>
  </processor>
  <processor class="solr.LogUpdateProcessorFactory" />
  <processor class="solr.RunUpdateProcessorFactory" />
</updateRequestProcessorChain>
```

schema.xml

If you are using a separate field for storing the signature you must have it indexed (See SOLR-1908)

```
<field name="signature" type="string" stored="true" indexed="true" multiValued="false" />
```

Note

Also be sure to change your update handlers to use the defined chain, i.e.

```
<requestHandler name="/update" class="solr.XmlUpdateRequestHandler" >
  <lst name="defaults">
    <str name="update.chain">dedupe</str>
  </lst>
</requestHandler>
```

The update processor can also be specified per request with a parameter of `update.chain=dedupe`

⚠️ Note that for pre-Solr3.2 you need to use `update.processor` instead

Settings

Setting	Default	Description
signatureClass	org.apache.solr.update.processor.Lookup3Signature	A Signature implementation for generating a signature hash.
fields	all fields	The fields to use to generate the signature hash in a comma separated list. By default, all fields on the document will be used.
signatureField	signatureField	The name of the field used to hold the fingerprint/signature. Be sure the field is defined in schema.xml.
enabled	true	Enable/disable dedupe factory processing