SimilarityScoringFilter

Similarity based Scoring

Summary

The link relevancy algorithm is based on the concept of the Cosine Similarity metric [1] to determine the semblance of fetched pages to accepted content in order to direct a NUTCH 1.X crawl in real-time. The score of how similar the text on a currently fetched page and that in user-specified gold-standard document are likely to be in terms of their subject matter is determined, the parsed page scored, and the all outlinks associated with that page scored accordingly.

Implementation

The implementation leverages the Cosine Similarity Metric [1] to compute how relevant the currently fetched page is to a given crawl. The gold-standard document and current page are converted into the term frequency vectors D0 and D1 respectively. Stopword filtering is performed during the conversion to the term frequency vectors and the stopwords are user specified through the configuration file. The formula for calculating the Cosine similarity is then given as

CosSim(D0,D1) = dot product of (vect(D0).vect(D1))

vect (D0)		vect (D1)

where |vect(X)| represents the Euclidean distance from the origin.

The formula provides the cosine of the angle of separation between the two vectors and is bounded between cos(90) = 0 and cos(0) = 1 (inclusive). Hence, the smaller the angle, the more similar the vectors are, and the score tends to 1.

The text-based features checked for during this score implementation of the Cosine Similarity Metric are:
Parsed Text from the page The HTML meta-keywords The HTML meta-description

How to use

Required file format of the model - The user should provide the goldstandard document in a text file with all the relevant text and terms, pertaining to the domain, present inside it.

1. Copy the gold-standard file into the conf directory and enter the name of this file in nutch-site.xml.

```
<property>
    <name>cosine.goldstandard.file</name>
    <value>goldstandard.txt</value>
</property>
```

Required file format for the stop words - The user can specify a custom list of stop words in a text file. Each new stopword should be on a new line.

2. Copy the stopwords.txt file to the conf directory. Now, update the path in the nutch-site.xml file

```
<property>
    <name>scoring.similarity.stopword.file</name>
    <value>stopwords.txt</value>
</property>
```

3. Enable the plugin by enabling scoring-similarity in the plugin.includes property in nutch-site.xml

```
<property>
    <name>plugin.includes</name>
    <value>protocol-http|urlfilter-regex|parse-(html|tika)|scoring-similarity|urlnormalizer-(pass|regex|basic)<
/value>
</property>
```

Example

The following is an example if how should thee goldstandard and stopwords file should look like.

Gold Standard file

This is a plain text file with no special formatting required. The contents of this file will be lowercased and split by a whitespace (Note: The tokenization process currently replaces any token which starts with a special character to a white space). A term frequency vector of the generated tokens is then created for computing the similarity.

For example, if you wanted to score pages similar to the field of robotics, your goldstandard could look like:

1. Only keyterms

```
Aerial Robotics Autonomous Agents Behaviour-Based Systems Brain Machine Interface Collision Avoidance Haptics
and Haptic Interfaces ......
```

2. Complete text (example text from a wikipedia article):

```
An autonomous robot is a robot that performs behaviors or tasks with a high degree of autonomy, which is
particularly desirable in fields such as space exploration, household maintenance (such as cleaning), waste
water treatment and delivering goods and services.

Some modern factory robots are "autonomous" within the strict confines of their direct environment. It may not
be that every degree of freedom exists in their surrounding environment, but the factory robot's workplace is
challenging and can often contain chaotic, unpredicted variables. The exact orientation and position of the
next object of work and (in the more advanced factories) even the type of object and the required task must be
determined. This can vary unpredictably (at least from the robot's point of view).

One important area of robotics research is to enable the robot to cope with its environment whether this be on
land, underwater, in the air, underground, or in space.

A fully autonomous robot can

Gain information about the environment
Work for an extended period without human intervention
Move either all or part of itself throughout its operating environment without human assistance
```

Stopword File

The stopword file should contain the words which you would like to omit while creating the term frequency vector. For example, the common terms in english grammar provide very little information about the relevance of a page and we would want them to be omitted.

Each stopword should be on a new line (Note: Stopwords do not undergo the process of tokenization, that means they will be used as they appear in the stopwords.txt file). Even if only keywords are used to create the goldstandard file, a list of stopwords should be provided, as the fetched pages may contain those terms.

Sample:

```
a
about
above
after
again
against
all
am
an
and
any
are
aren't
as
at
be
because
been
before
being
below
between
both
but
by
can't
cannot
could
couldn't
did
didn't
do
....
```

Future work

- Allow for other media formats in the gold-standard document
- Include text from title tags of images in the features checked during the Cosine Similarity Metric score calculation.
- Allow user to input multiple documents as the gold-standard document
- In Data cleaning task, perform tokenization of the documents using the Lucene tokenizer. In this case, use Lucene's STOP_SET as the default stopword.
- Currently all the outlinks are given from a given page are given the parent's score. It is proposed to parse the anchor tag and the URL string of each outlink, and perform a weighted scoring for better relevancy and boosting the score of individual URL.
- Allow user to enter a threshold for URL scores to be ingested added to the generator at the next round.