DBPedia Spotlight - Training Material | WP3

Author: Bob Boelhouwer INL internal review: Jesse de Does, Katrien Depuydt – December 2013

Introduction.

DBpedia Spotlight is a tool for automatically annotating mentions of DBpedia resources in text, providing a solution for linking unstructured information sources to the Linked Open Data cloud through DBpedia. DBpedia Spotlight recognizes that names of concepts or entities have been mentioned (e.g. "Michael Jordan"), and subsequently matches these names to unique identifiers (e.g. dbpedia:Michael_I._Jordan, the machine learning professor or dbpedia:Michael_Jordan the basketball player). It can also be used for building your solution for Named Entity Recognition, Keyphrase Extraction, Tagging, etc. amongst other information extraction tasks. (From https://github.com/dbpedia-spotlight/dbpedia-spotlight/wiki/Introduction.)
The software is available on basis of an Apache License (ASL 2.0) for free.
It can be deployed in basically two ways. First, as a client using the free webservice provided by DBPedia and, second, as server software that the user can install on his own machine and that provides the same webservice.
The first option is, of course, the easiest. The second option, however, will allow the user to tune and tweak the service, install additional specialized authority databases, and have control over the speed of the service.
DBPedia Spotlight has two approaches:

1. Using Lucene and LingPipe. LingPipe is used to spot the named entities in the text. For disambiguation the wikilinks appearing in an DBPEdia article of a candidate are compared with keywords in the context of the spotted NE. The best match determines the selection. This approach is only available for the English language.

2. Using a statistical method. OpenNLP is used to spot the named entities. Disambiguation is performed using a generative probabilistic model based on the lexical context of the named entity. This approach supports several languages: English, Dutch, French, German, Hungarian, Italian, Portuguese, Russian, Spanish and Turkish.

Requirements

As was mentioned in the previous section, if the web service of DBPedia is used, no software needs to be installed. But the workflow will require an internet connection and a facility to send and retrieve information over http. An example of a simple client application is CURL, which is available for most LINUX or UNIX distributions.

Usage

Since web services can be applied to an unlimited number of purposes, we will not try to describe all possible applications. In this document we will explain how a request is compiled and how that request can be submitted to the web service with a simple command line tool like CURL. In the context of a workflow the API will likely be accessed via scripting tools like PHP or Perl. For most of these tools there are modules available that can send and retrieve information over http.

Succeed is supported by the European Union under FP7-ICT and coordinated by Universidad de Alicante.

The API is called using a REST interface. Basically, the interaction consists of a string that is sent by an http-client to a server of DBPedia. That server than performs an action on the data and returns a file with the result to the http-client. The string that is sent to the service is in a standard URL format like '`http://nl.dbpedia.org/spotlight/rest/annotate`'. GET request parameters are used to further instruct the web service. These will be explained below.
DBPedia Spotlight provides an online manual. In that manual there are several examples presented that use several features of the REST interface.
Note that there are limitations to the size of the text that is uploaded.

- Using the **text** parameter (with a plain text file, .txt): The limit is a plain text file of 460kB (which is 460000 characters)
- Using the **url** parameter (with the url of a .html file): The limit is a html file of 490kB

There are several endpoints available for the REST interface. For the first approach (using Lucene and LingPipe) the endpoint is:

'`http://spotlight.dbpedia.org/rest/`'.

This approach is only available for English. For the second approach there are several endpoints available, each corresponding with a certain language:

* English - `http://spotlight.sztaki.hu:2222/rest`
* Dutch - `http://nl.dbpedia.org/spotlight/rest`
* French - `http://spotlight.sztaki.hu:2225/rest`
* German - `http://de.dbpedia.org/spotlight/rest`
* Hungarian - `http://spotlight.sztaki.hu:2229/rest`
* Italian - `http://spotlight.sztaki.hu:2230/rest`
* Portuguese - `http://spotlight.sztaki.hu:2228/rest`
* Russian - `http://spotlight.sztaki.hu:2227/rest`
* Spanish - `http://spotlight.sztaki.hu:2231/rest`
* Turkish - `http://spotlight.sztaki.hu:2235/rest`

There are several services available, which are described below.

Spotting. Takes text as input and recognizes surface forms -- e.g. names of entities/concepts to annotate. Several spotting techniques are available, such as dictionary lookup and Named Entity Recognition (NER). The endpoint has to be augmented with '`/spot`' like

'`http://spotlight.sztaki.hu:2222/rest/spot`'

There are two parameters available for spotting:

- text: input text to annotate

- spotter: the spotter implementation to use. One of: Default, LingPipeSpotter, AtLeastOne-NounSelector, CoOccurrenceBasedSelector, NESpotter, KeyphraseSpotter, OpenNLPChunkerSpotter, WikiMarkupSpotter, SpotXmlParser, AhoCorasickSpotter

The spotters are not documented. In order to find the best candidate for an application, one has to run a number of tests.

A full request for spotting might have the following form:

'http://spotlight.dbpedia.org/rest/spot/?text=Berlin&spotter=LingPipeSpotter'

The result of this request looks like this:

```
<annotation text="Berlin"><surfaceForm name="Berlin" offset="0"/></annotation>
```

Disambiguate. Takes spotted text input, where entities/concepts have already been recognized and marked as wiki markup or xml. Chooses an identifier for each recognized entity/concept given the context. The service provides a single parameter:

- text: annotated text, e.g. the output of the 'spot' service. Note that the text has to be properly encoded in order to be sent over http. Below, in the *Example of Use*, we will present a method to do the encoding.

A full request for disambiguation using the result of the previous request will look like this:

'http:/spotlight.dbpedia.org/rest/disambiguate/?text= %3Cannotation+text%3DBerlin%3E%3CsurfaceForm+name%3DBerlin+offset%3D0%2F%3E%3C%2Fannotation%3E'

Annotate. Runs spotting and disambiguation. Takes text as input, recognizes entities/concepts to annotate and chooses an identifier for each recognized entity/concept given the context. This service is further explained in the Example of Use, below.

Candidates. Similar to annotate, but returns a ranked list of candidates instead of deciding on one.

Example of Use

In the example below, a Dutch text is to be annotated.
In this example we use the unix tool 'curl'. Curl sends a http request to the url specified in the first argument. The parameter and value pairs to that request can be specified in a number of '--data' arguments. The argument '—data-urlencode' can be used if values need to be encoded. The argument –H specifies a header field that will be sent. In the case below we inform the application that we want to receive xml-data.

```
curl -H "Accept: text/xml" http://nl.dbpedia.org/spotlight/rest/annotate \
  --data-urlencode "text=Het zijn gouden tijden voor de zanger. Naast de zeer goede
kaartverkoop, scoort hij op dit moment met zowel zijn album Duizend Spiegels als
zijn single met Trijntje Oosterhuis en het ontroerende nummer met dochter Jada,
boezemvriend John Ewbank en diens dochter Day en Lange Frans en zoon Willem."
```

This call will deliver the following result:

```
<?xml version="1.0" encoding="utf-8"?>
<Annotation text="Het zijn gouden tijden voor de zanger. Naast de zeer goede
kaartverkoop, scoort hij op dit moment met zowel zijn album Duizend Spiegels als
zijn single met Trijntje Oosterhuis Ã©n het ontroerende nummer met dochter Jada,
boezemvriend John Ewbank en diens dochter Day en Lange Frans en zoon Willem."
confidence="0.1" support="10" types="" sparql="" policy="whitelist">
<Resources>
<Resource URI="http://nl.dbpedia.org/resource/Trijntje_Oosterhuis" support="126"
types="" surfaceForm="Trijntje Oosterhuis" offset="156"
similarityScore="0.9999999829444733" percentageOfSecondRank="1.7055535449342053E-
8"/>
<Resource URI="http://nl.dbpedia.org/resource/John_Ewbank" support="46" types=""
surfaceForm="John Ewbank" offset="233" similarityScore="1.0"
percentageOfSecondRank="0.0"/>
<Resource URI="http://nl.dbpedia.org/resource/Lange_Frans_(rapper)" support="108"
types="" surfaceForm="Lange Frans" offset="269"
similarityScore="0.9999999999999432" percentageOfSecondRank="0.0"/>
</Resources>
</Annotation>
```

The named entities are provided with a link to a resource. Also, some properties of the match are returned.

- **Support**: expresses how prominent this entity is. Based on the number of inlinks in Wikipedia.
- **percentageOfSecondRank**: measure how much the winning entity has won by taking contextualScore_2ndRank / contextualScore_1stRank, which means the lower this score, the further the first ranked entity was "in the lead"
- **similarityScore**: similarity between surface form and recorded forms according to Lucene.
- **types**: list of *DBpediaType* objects.

It is also possible to retrieve an html-document with the original text in which the named entities are linked to an authority file. This can be achieved by changing the header argument in the previous call into '`-H "Accept: text/html"`' or leaving it out altogether.
The result will then be something like this:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>DBpedia Spotlight annotation</title>
<meta http-equiv="Content-type" content="text/html;charset=UTF-8">
</head>
<body>
<div>
Het zijn gouden tijden voor de zanger. Naast de zeer goede kaartverkoop, scoort hij
op dit moment met zowel zijn album Duizend Spiegels als zijn single met <a
href="http://nl.dbpedia.org/resource/Trijntje_Oosterhuis"
title="http://nl.dbpedia.org/resource/Trijntje_Oosterhuis" target="_blank">Trijntje
Oosterhuis</a> Ã©n het ontroerende nummer met dochter Jada, boezemvriend <a
href="http://nl.dbpedia.org/resource/John_Ewbank"
otitle="http://nl.dbpedia.org/resource/John_Ewbank" target="_blank">John Ewbank</a>
en diens dochter Day en <a
```

```
href="http://nl.dbpedia.org/resource/Lange_Frans_(rapper)"
title="http://nl.dbpedia.org/resource/Lange_Frans_(rapper)" target="_blank">Lange
Frans</a> en zoon Willem.
</div>
</body>
</html>
```

If it is required for some purpose to use a different NE-tagger from what is provided by DBPedia, it is possible to use the endpoint ‘disambiguate’ (see above) on a text that is NE-tagged by that different system. It will probably be necessary to convert the output of that system to the format that is expected by the endpoint ‘disambiguate’. Below is an example of that format.

```
<annotation text="Het zijn gouden tijden voor de zanger. Naast de zeer goede
kaartverkoop, scoort hij op dit moment met zowel zijn album Duizend Spiegels als
zijn single met Trijntje Oosterhuis Ã©n het ontroerende nummer met dochter Jada,
boezemvriend John Ewbank en diens dochter Day en Lange Frans en zoon Willem.">
  <surfaceForm name="Trijntje Oosterhuis" offset="156"/>
  <surfaceForm name="John Ewbank" offset="233"/>
  <surfaceForm name="Lange Frans" offset="269"/>
</annotation>
```

So, the named entities have to be listed with their offset separated from the text.

Relevant literature:

Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes, “Improving efficiency and accuracy in multilingual entity extraction”. In: *Proceedings of the 9th International Conference on Semantic Systems*, page 121--124. New York, NY, USA, ACM, (2013)

Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer, “DBpedia spotlight: shedding light on the web of documents”. In: *Proceedings of the 7th International Conference on Semantic Systems*, page 1--8. New York, NY, USA, ACM, (2011)