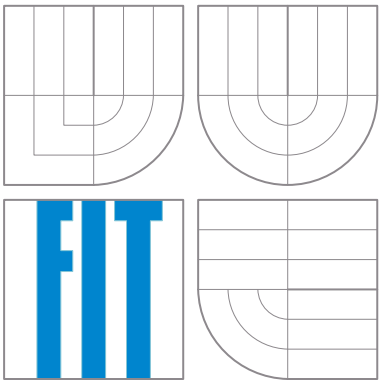

FACULTY OF INFORMATION TECHNOLOGY
BRNO UNIVERSITY OF TECHNOLOGY

Documentation of the project for the KNOT group

SEC API

July 16, 2014

1 Introduction

This describes extensions to the SEC API (Application Programming Interface) that make available two key functions of the Semantic Annotator – services `annotate` and `get_entities`. One other meta-service providing a list of semantic types included in the knowledge base and a definition of corresponding attributes stored – `get_entity_types_and_attributes` is also described. The services can be accessed by clients with varying needs, including the other two key components of the DECIPHER system – the Content Aggregator and Storyscope.

The interface defines an HTTP-based protocol with data in JSON. Clients send data through the standard HTTP POST method. Field API in the HTTP header shall have value SEC_API, field version shall have value 1.2.

2 Service `annotate`

The service returns annotations for a given document. Annotations are primarily represented by their begin offsets, end offset, and internal IDs linking the mentions to the knowledge base. If the parameter `disambiguate` is set to 0, the Semantic Annotator does not return disambiguated entities but all possible entities corresponding to each particular name. By default, the parameter is set to 1.

The annotation output can be also produced in one or more formats specified by the parameter `annotation_format`. The following values of the parameter are supported (example output of these formats are shown below):

- `html` – The output is an HTML document that contains the original text where the annotated parts are underlined. If the document is opened in a browser, then mouse over the underlined text is displayed in a frame annotation on this text requested by `types_and_attributes`. The output includes annotated and non-annotated text.
- `index` – Annotations will be provided in a plain text form intended for fulltext indexing. The output includes annotated and non-annotated text.
- `nif` – An NIF version of the annotation will be produced. Output only contains annotated text.
- `rdf` – An RDF version of the annotation will be produced. Output only contains annotated text.
- `sxml` – An internal XML version of the annotation will be produced. Output only contains annotated text.
- `text` – Annotations will be provided in a plain text form intended for human reading, they will be directly included the textual output. Output only contains annotated text.
- `xml` – An XML version of the annotation will be produced. The output includes annotated and non-annotated text.

For each format included in the parameter `annotation_format`, an actual content of annotations expected in the output needs to be specified by a relevant parameter `types_and_attributes`.
The request has to provide:

- `input_text` – JSON type: string – The text to be annotated.
- `annotation_format` – JSON type: array – A list of formats in which results should be formatted.

Optional fields of the request are:

- `disambiguate` – JSON type: number (int) – 1 (default) - produce an disambiguated output, 0 - do not disambiguate
- `types_and_attributes` – JSON type: string OR object – Specifies which types are annotated and which attributes are to these types listed. The default value is string "all", which means that all types will be listed with all their attributes. You can also enter a value JSON object (name/value pairs) with the name representing the type and value which can be either the JSON string "all", which means that all his attributes will be listed, or an JSON array of his attributes (JSON strings) which should be listed.

- document_uri – JSON type: string – URL address at which the text, which is also in the parameter input_text. This parameter is required only if parameter annotation_format also contains the value "nif", otherwise it is ignored.

The response contains one field corresponding to format specified in the request.

Syntax
Request:

```
{
   "annotate": {
      "input_text": "",
      "annotation_format": ["text"|"index"|"xml"|"html"|"rdf"],
      "disambiguate": 1,
      "document_uri": "",
      "types_and_attributes": { ... },
   }
}
```

Response:

```
{
   "annotation": ""
}
```

Example
Request:

```
{
   "annotate": {
      "annotation_format": [
         "html",
         "index",
         "nif",
         "rdf",
         "sxml",
         "text",
         "xml"
      ],
      "document_uri": "http://wiki-link.nlp2rdf.org/data/1a/0a/06434
         d8769660b5aabe0e047ccb5/67cd7b68aa9a54ab8db92562b2a6edca",
      "input_text": "William Orpen was an Irish portrait painter, who worked mainly
         in London.\n",
      "types_and_attributes": {
         "artist": [
            "display_term",
            "wikipedia_url"
         ],
         "location": [
            "name"
         ]
      }
   }
}
```

Response (modified-commented):

```
{  // HTML annotation format
   "annotation": "<!doctype html>\n<html>\n<head>\n\t<meta http-equiv=\"Content-Type
      \" content=\"text/html; charset=UTF-8\">\n\t<style type=\"text/css\">\n\t\tspan
      .annotation {\n\t\t\ttext-decoration: none;\n\t\t}\n\t\tspan.annotation img {\n
      \t\t\tmax-width: 300px;\n\t\t}\n\t\tspan.annotation span.annotation {\n\t\t\
```

```
      ttext-decoration: underline;\n\t\t\tcolor: #00f;\n\t\t}\n\t\tspan.annotation
      span.annotation_features {\n\t\t\tdisplay: none;\n\t\t\tcolor: #000;\n\t\t}\n\t
      \tspan.annotation:hover {\n\t\t\tposition: relative;\n\t\t}\n\t\tspan.
      annotation:hover span.annotation_features {\n\t\t\tdisplay: block;\n\t\t\twidth
      : 300px;\n\t\t\tposition: absolute; top: 10px; left: 0;\n\t\t\t/* formatting
      only styles */\n\t\t\tpadding: 5px; margin: 10px; z-index: 100;\n\t\t\
      tbackground: #f0f0f0; border: 1px dotted #c0c0c0;\n\t\t\topacity: 0.9;\n\t\t\t
      /* end formatting */\n\t\t}\n\t</style>\n</head>\n\n<body>\n<span class=\"
      annotation\">\n<span class=\"annotation artist\">William Orpen</span>\n<span
      class=\"annotation_features\">\n<span class=\"annotation_type\">artist</span>\n
      <span class=\"att_name display_term\">display_term</span>: <span class=\"
      att_value display_term\">Sir William Orpen</span><br>\n</span>\n</span>\n was
      an Irish portrait painter, who worked mainly in <span class=\"annotation\">\n<
      span class=\"annotation location\">London</span>\n<span class=\"
      annotation_features\">\n<span class=\"annotation_type\">location</span>\n<span
      class=\"att_name name\">name</span>: <span class=\"att_value name\">London</
      span><br>\n</span>\n</span>\n.\n</body>\n</html>\n"
},
{  // INDEX annotation format
   "annotation": "William Orpen[artist;Sir_William_Orpen_displayterm] was an Irish
      portrait painter, who worked mainly in London[location;London_name].\n"
},
{  // NIF annotation format
   "annotation": "<http://wiki-link.nlp2rdf.org/data/1a/0a/06434
      d8769660b5aabe0e047ccb5/67cd7b68aa9a54ab8db92562b2a6edca#char=0,13>\n\ta nif:
      String , nif:RFC5147String ;\n\tnif:referenceContext <http://wiki-link.nlp2rdf.
      org/data/1a/0a/06434d8769660b5aabe0e047ccb5/67cd7b68aa9a54ab8db92562b2a6edca#
      char=0,> ;\n\tnif:anchorOf \"William Orpen\"^^xsd:string ;\n\tnif:beginIndex
      \"0\"^^xsd:long ;\n\tnif:endIndex \"13\"^^xsd:long ;\n\ta nif:Phrase ;\n\
      titsrdf:taIdentRef <http://collection.britishcouncil.org/artist/artist
      /30783/17590> .\n\n<http://wiki-link.nlp2rdf.org/data/1a/0a/06434
      d8769660b5aabe0e047ccb5/67cd7b68aa9a54ab8db92562b2a6edca#char=66,72>\n\ta nif:
      String , nif:RFC5147String ;\n\tnif:referenceContext <http://wiki-link.nlp2rdf.
      org/data/1a/0a/06434d8769660b5aabe0e047ccb5/67cd7b68aa9a54ab8db92562b2a6edca#
      char=0,> ;\n\tnif:anchorOf \"London\"^^xsd:string ;\n\tnif:beginIndex \"66\"^^
      xsd:long ;\n\tnif:endIndex \"72\"^^xsd:long ;\n\ta nif:Phrase ;\n\titsrdf:
      taIdentRef <http://en.wikipedia.org/wiki/London> .\n"
},
{  // RDF annotation format
   "annotation": "<?xml version=\"1.0\" encoding=\"utf-8\"?>\n<rdf:RDF\n xmlns:a=\"
      http://knot.fit.vutbr.cz/annotations/AnnotSchema/\"\n xmlns:rdf=\"http://www.w3
      .org/1999/02/22-rdf-syntax-ns#\"\n>\n <a:artist rdf:nodeID=\"
      Ne178390394684f5db27a2c47c74698cf\">\n <a:display_term>Sir William Orpen</a:
      display_term>\n </a:artist>\n <a:location rdf:nodeID=\"
      N05063b135e374419b060a33e33ed1bd7\">\n <a:name>London</a:name>\n </a:location>\
      n</rdf:RDF>\n"
},
{  // SXML annotation format
   "annotation": "<?xml version=\"1.0\" encoding=\"utf-8\"?>\n<suggestion>\n <text
      e_offset=\"13\" s_offset=\"0\" string=\"William Orpen\">\n <annotation type=\"
      artist\">\n <attribute name=\"display_term\" type=\"string\">Sir William Orpen
      </attribute>\n <attribute name=\"wikipedia_url\" type=\"uri\"/>\n </annotation
      >\n </text>\n <text e_offset=\"72\" s_offset=\"66\" string=\"London\">\n <
      annotation type=\"location\">\n <attribute name=\"name\" type=\"string\">London
      </attribute>\n </annotation>\n </text>\n</suggestion>\n"
},
{  // TEXT annotation format
   "annotation": "William Orpen\n=============\n[artist]\ndisplay_term: Sir William
      Orpen\n\nLondon\n======\n[location]\nname: London\n\n"
```

```
},
{  // XML annotation format
   ”annotation”: ”<annotation><artist display_term=\”Sir William Orpen\”
      wikipedia_url=\”\”>William Orpen</artist> was an Irish portrait painter, who
      worked mainly in <location name=\”London\”>London</location></annotation>”
}
```

Response – part HTML annotation format (printed):

```
<!doctype html>
<html>
<head>
      <meta http-equiv=”Content-Type” content=”text/html; charset=UTF-8”>
      <style type=”text/css”>
            span.annotation {
                  text-decoration: none;
            }
            span.annotation img {
                  max-width: 300px;
            }
            span.annotation span.annotation {
                  text-decoration: underline;
                  color: #00f;
            }
            span.annotation span.annotation_features {
                  display: none;
                  color: #000;
            }
            span.annotation:hover {
                  position: relative;
            }
            span.annotation:hover span.annotation_features {
                  display: block;
                  width: 300px;
                  position: absolute; top: 10px; left: 0;
                  /* formatting only styles */
                  padding: 5px; margin: 10px; z-index: 100;
                  background: #f0f0f0; border: 1px dotted #c0c0c0;
                  opacity: 0.9;
                  /* end formatting */
            }
      </style>
</head>

<body>
<span class=”annotation”>
<span class=”annotation artist”>William Orpen</span>
<span class=”annotation_features”>
<span class=”annotation_type”>artist</span>
<span class=”att_name display_term”>display_term</span>: <span class=”att_value
   display_term”>Sir William Orpen</span><br>
</span>
</span>
 was an Irish portrait painter, who worked mainly in <span class=”annotation”>
<span class=”annotation location”>London</span>
<span class=”annotation_features”>
<span class=”annotation_type”>location</span>
<span class=”att_name name”>name</span>: <span class=”att_value name”>London</span><
   br>
</span>
```

```
</span>
.
</body>
</html>
```

Response – part INDEX annotation format (printed):

```
William Orpen[artist;Sir_William_Orpen_displayterm] was an Irish portrait painter,
   who worked mainly in London[location;London_name].
```

Response – part NIF annotation format (printed):

```
<http://wiki-link.nlp2rdf.org/data/1a/0a/06434d8769660b5aabe0e047ccb5/67
   cd7b68aa9a54ab8db92562b2a6edca#char=0,13>
      a nif:String , nif:RFC5147String ;
      nif:referenceContext <http://wiki-link.nlp2rdf.org/data/1a/0a/06434
         d8769660b5aabe0e047ccb5/67cd7b68aa9a54ab8db92562b2a6edca#char=0,> ;
      nif:anchorOf "William Orpen"^^xsd:string ;
      nif:beginIndex "0"^^xsd:long ;
      nif:endIndex "13"^^xsd:long ;
      a nif:Phrase ;
      itsrdf:taIdentRef <http://collection.britishcouncil.org/artist/artist
         /30783/17590> .

<http://wiki-link.nlp2rdf.org/data/1a/0a/06434d8769660b5aabe0e047ccb5/67
   cd7b68aa9a54ab8db92562b2a6edca#char=66,72>
      a nif:String , nif:RFC5147String ;
      nif:referenceContext <http://wiki-link.nlp2rdf.org/data/1a/0a/06434
         d8769660b5aabe0e047ccb5/67cd7b68aa9a54ab8db92562b2a6edca#char=0,> ;
      nif:anchorOf "London"^^xsd:string ;
      nif:beginIndex "66"^^xsd:long ;
      nif:endIndex "72"^^xsd:long ;
      a nif:Phrase ;
      itsrdf:taIdentRef <http://en.wikipedia.org/wiki/London> .
```

Response – part RDF annotation format (printed):

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF
  xmlns:a="http://knot.fit.vutbr.cz/annotations/AnnotSchema/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
>
  <a:location rdf:nodeID="Nbadead3876c147feaf0b63bcd14f0eb9">
    <a:name>London</a:name>
  </a:location>
  <a:artist rdf:nodeID="N68c5e429561e4d06a2613bc08e7a382d">
    <a:display_term>Sir William Orpen</a:display_term>
  </a:artist>
</rdf:RDF>
```

Response – part SXML annotation format (printed):

```
<?xml version="1.0" encoding="utf-8"?>
<suggestion>
  <text e_offset="13" s_offset="0" string="William Orpen">
    <annotation type="artist">
      <attribute name="display_term" type="string">Sir William Orpen</attribute>
      <attribute name="wikipedia_url" type="uri"/>
    </annotation>
  </text>
  <text e_offset="72" s_offset="66" string="London">
    <annotation type="location">
      <attribute name="name" type="string">London</attribute>
```

```
   </annotation>
  </text>
</suggestion>
```

Response – part TEXT annotation format (printed):

```
William Orpen
=============
[artist]
display_term: Sir William Orpen

London
======
[location]
name: London
```

Response – part XML annotation format (printed):

```
<annotation><artist display_term="Sir William Orpen" wikipedia_url="">William Orpen</
    artist> was an Irish portrait painter, who worked mainly in <location name="London
    ">London</location></annotation>
```

3 Service `get_entities`

The service returns data on entities stored in the knowledge base sorted by confidence.
The request has to specify:

- `input_string` – JSON type: string – A name of entity or an initial part of that name ended by the asterisk sign (*) which will be searching in knowledge base.
- `types_and_attributes` – JSON type: string OR object – Specifies which types will be searching in knowledge base and which attributes are to these types will be listing. The default value is string "all", which means that all types will be listed with all their attributes. You can also enter a value JSON object (name/value pairs) with the name representing the type and value which can be either the JSON string "all", which means that all his attributes will be listed, or an JSON array of his attributes (JSON strings) which should be listed.

Optionally, it can also provide:

- `max_results` – JSON type: number (int) – A maximal number of entities returned (the default is 10).

The response contains a list of entities (values of attributes specified in the input parameter `types_and_attributes`).

Syntax
Request:

```
{
   "get_entities":{
      "input_string": "",
      "types_and_attributes": { ... },
      "max_results": int
   }
}
```

Response:

```
{
   "data": [
      {
         type: { attribute: "", ... }
      },
```

```
      ...
   ]
}
```

Example
Request:

```
{
   "get_entities": {
      "input_string": "Met*",
      "max_results": 3,
      "types_and_attributes": {
         "artist": [
            "display_term",
            "wikipedia_url"
         ]
      }
   }
}
```

Response:

```
{
   "data": [
      {
         "artist": {
            "display_term": "Meta Vaux Warrick Fuller",
            "wikipedia_url": "http://en.wikipedia.org/wiki/Meta_Vaux_Warrick_Fuller"
         }
      },
      {
         "artist": {
            "display_term": "Valentin Metzinger",
            "wikipedia_url": ""
         }
      },
      {
         "artist": {
            "display_term": "Marty Metalgod Ross",
            "wikipedia_url": ""
         }
      }
   ]
}
```

4 Service `get_entity_types_and_attributes`

The service returns types of entities and their respective attributes stored in the knowledge base.

Syntax
Request:

```
{
   "get_entity_types_and_attributes": {}
}
```

Response:

```
{
   "data": [
```

```
      {
         "type": "",
         "attributes": [ ... ]
      },
      ...
   ]
}
```

Example

Request:

```
{
   "get_entity_types_and_attributes": {}
}
```

Response:

```
{
   "data": [
      {
         "type": "artist",
         "attributes": [
            "type",
            "display_term",
            "preferred_term",
            "other_term",
            ...
         ]
      },
      ...
   ]
}
```