ELSEVIER

Elsevier Fingerprint Engine ™

API Documentation

November, 2018

Elsevier Research Intelligence

Contents:

1 Introduction

The Elsevier Fingerprint Engine (also EFE, or FPE) is a back-end software system of state-of-the-art Natural Language Processing (NLP) techniques to extract information from unstructured academic text. Applying domain-relevant thesauri or controlled vocabularies to scientific publications of various types, the fingerprint engine maps text to semantic 'fingerprints', collections of weighted key concepts. By identification and extraction of new concepts the Elsevier Fingerprint Engine can also enrich thesauri and generate new vocabularies.

The Elsevier Fingerprint Engine can be used as a stand-alone service, or as a back-end processing component of applications as it is used in Elsevier products such as Expert Lookup, Pure, or SciVal. Third parties can purchase the toolkit as a software product for the concept mining of large volumes of academic text, or can use its concept mining functionality on subscription basis through a RESTful web API.

This document provides a brief overview of the EFE through the RESTful web API, and contains instructions on how to use the RESTful web API.

1.1 Thesauri and controlled vocabularies

The Elsevier Fingerprint Engine integrates a range of thesauri to support applications pertaining to different subject areas, including a number of traditional popular ones like the Medical Subject Headings (MeSH), the National Agriculture Library's (NAL) thesaurus and Elsevier's Compendex thesaurus. To improve coverage we use the Fingerprint Engine to enrich existing thesauri (Cambridge Math thesaurus, STW thesaurus for Economics) and develop stand-alone vocabularies (e.g., for the humanities). Together, the thesauri and controlled vocabularies allow to annotate text with concepts and terms for specific scientific domains. Where a domain is described by a thesaurus, which in contrast to vocabularies structure concepts in a hierarchy of broader and narrower concepts, the term annotations contain identifiers that link to concepts in the hierarchy, as exemplified in Figure 1 for MeSH based indexing.

In its current standard configuration, the Elsevier Fingerprint Engine supports a set of thesauri and controlled vocabularies as listed in Appendix 1. In addition, key thesauri have been joined in a Unified Thesaurus. This meta-thesaurus supports the use case in which there is no need or desire to focus on specific thesauri.

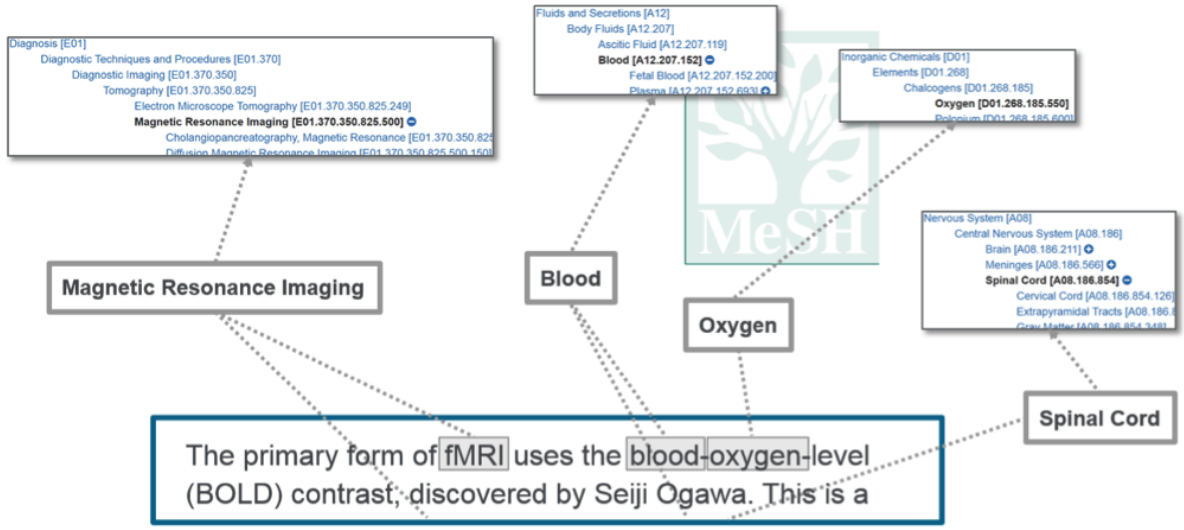

Figure 1: Terms recognized as belonging to MeSH concepts.

2 A closer look to the Elsevier Fingerprint Engine

2.1 Fingerprinting as a workflow

The Elsevier Fingerprint Engine identifies relevant technical concepts in a text based on a thesaurus or vocabulary.

The concept finding algorithm is sensitive to lexical and grammatical features - casing, word order, part-of-speech and others - when it must be - e.g., to distinguish Windows(®) from windows, the noun from the verb 'lead', etc. At the same time it ignores differences when they have no meaning - e.g., the differences between 'tumour' and 'tumor', between 'kidney failure' and 'failure of the kidney' etc.

In addition, concept finding takes into account the context of terms. It looks at their neighbors and will, e.g., not identify a "non-Hodgkin Lymphoma" as a Hodgkin Lymphoma or the ' tree of human ancestry' as a plant, but also at their wider environment and will, e.g., not interpret 'administration' as management in a text about a drug as a treatment for a disease.

Concepts found in documents are weighted according to their frequency, their occurrence in a text's title or text body or according to their occurrence in automatically detected subsections of a text's body.

So-called Named Entities like the names of people ('John O'Keefe') and places ('Philadelphia, Pennsylvania') are identified and disambiguated across thesauri and vocabularies and can be presented separated from fingerprints proper.

2.2 Natural Language Processing components

In order to capture all possible term instance (maximizing recall) while keeping erroneous annotations minimal (maximizing precision), the Fingerprint Engine relies on a processing pipeline of natural language processing components. The most important components are the following:

Language Detection: Identifies the language in which a text is written.

Tokenization: Splits text in tokens like words, punctuation marks and sentences.

Dehyphenation: Recognizes sentence-final hyphenations and reconstructs the original words, for instance replaces "dehyph- enation" with "dehyphenation".

Coordination Expansion: Detects abbreviated coordinations and reconstructs full forms. For instance, the phrase "intra- and extramural" is expanded to "intramural and extramural". Similarly, full noun phrases are reconstructed from compacted ones, e.g. "Hepatitis A and B" is expanded to "Hepatitis A and Hepatitis B".

Normalization: Produces normal forms converting plural to singular forms (children > child) and British to American spelling variants (gynaecology > gynecology).

Abbreviation Expansion: Detects and expands abbreviations that are defined in a text. For instance, if the phrase "Blood group (BG)" is detected all occurrences of "BG" in the same text are expanded to "Blood group".

Entity Recognition: Recognizes specific entities like email addresses, URLs, citations and chemicals using regular expressions. For example "b[a-z]+kinaseb" recognizes simple enzymes while "[A-Z][a-z]+ ([0-9]+)" recognizes simple citations.

Part-of-Speech Tagging: Tags tokens as linguistic parts of speech (verb, noun etc.) depending on their context. E.g., the word 'lead' will be tagged as verb in "This lead to the conclusion...", as a noun in "....where lead concentration was high".

Term Finder: Finds occurrences of the terms of a thesaurus or vocabulary in preanalyzed text (i.a. by the modules described above).

Term Annotation: Marks thesaurus terms identified in text with flags providing further information about them, most notably to exclude terms from concept assignment (see below) when disambiguation routines found its meaning in the given context to differ from its meaning in the applied thesaurus.

Idiom Removal: Excludes known idioms from concept assignment. For instance, "on the other hand" will not produce the concept "hand".

Fingerprint Creation: Assigns concepts to the remaining found terms and, based on a set of criteria (see above), assigns a weight to each concept.

3 Getting started with the API

3.1 General

Upon obtaining a subscription or test account from Elsevier, authentication for any call requires credentials to be provided using BASIC AUTHENTICATION. The API functionality can be subdivided in *thesaurus based indexing*, *thesaurus querying*, *thesaurus-free indexing* and *classification*.

The current API version ingests either plain text (for single text/document calls) or XML (for sectioned document or multiple document calls), and returns XML containing annotations. The payload provided with POST calls generally cannot exceed 25MB of text data.

The thesaurus-based indexing can be done by providing an individual text / document as payload, or providing multiple texts documents as payload in a single call (see *Bulk indexing*)

3.2 Thesaurus based indexing

Each thesaurus has its own end-point, which can be called with POST requests containing *plain text* or *XML* payload, and returns a set of annotations of different types:

- `ConceptAnnotation`, with most importantly the preferred name of the concept (`Name`) and its ID in the thesaurus (`ConceptID`)
- `TermAnnotation`, with most importantly the concept it belongs to (`ConceptID`), its own ID (`TermID`), the text span involved (`Text`), and character offset and end positions in the text (`TextOffset` and `TextEnd`)
- `POSTag`, with most importantly the part-of-speech (whether noun, verb, etc.)
- `Word`, `Sentence`, and `Token`, denoting word and non-word token information

To obtain concept and term annotations using a specific thesaurus, the API can be called in the following way:

`POST /Taco[version]/TacoService.svc/[thesaurus]`

Where *[version]* is the long (four digit) version number of the engine to use, and

[thesaurus] is a string indicating which thesaurus to index with. Currently supported thesauri strings are: `CpxTree`, `Chemistry`, `Geobase`, `MathVoc`, `MeSH`, `NAL`, `NASA`, `Gesis`, `Humanities`, `Economics`, `Unified`.

Request Headers

- Authorization – HTTP Basic authentication with provided credentials

Status Codes

- 200 OK – no error
- 400 Bad Request – bad request, because of malformed URI

Example request:

```
POST /Taco7900/TacoService.svc/NAL HTTP/1.1
Host: fingerprintengine.scivalcontent.com
Accept: text/plain

"About oceanography."
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/xml
```

```
<TextAnalysis>
  <Annotations>
    <Annotation i:type="ConceptAnnotation">
      <AFreq>1</AFreq>
      <ConceptID>55558</ConceptID>
      <Name>oceanography</Name>
      <Rank>1</Rank>
      <Thesaurus>NAL</Thesaurus>
    </Annotation>
    <Annotation i:type="TermAnnotation">
      <ConceptID>55558</ConceptID>
      <Flags xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
        <a:string>L701.558</a:string>
      </Flags>
      <MatchScore>0.9</MatchScore>
      <OrderingQuality>1</OrderingQuality>
      <TermID>T093163</TermID>
      <TermType>PrefLabel</TermType>
      <Text>oceanography</Text>
      <TextEnd>18</TextEnd>
      <TextOffset>6</TextOffset>
      <Thesaurus>NAL</Thesaurus>
      <Tokens xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
        <a:int>2</a:int>
      </Tokens>
    </Annotation>
    <Annotation i:type="POSTag">
      <End>1</End>
      <Offset>0</Offset>
      <Alternative i:nil="true"/>
      <Distance>0</Distance>
      <LexicalSource>0</LexicalSource>
      <SemTag>rb</SemTag>
      <Tag>rb</Tag>
    </Annotation>
    <Annotation i:type="POSTag">
      <End>3</End>
      <Offset>2</Offset>
      <Alternative i:nil="true"/>
```

(continues on next page)

(continued from previous page)

```
      <Distance>0</Distance>
      <LexicalSource>0</LexicalSource>
      <SemTag>nn</SemTag>
      <Tag>nn</Tag>
    </Annotation>
    <Annotation i:type="Word">
      <Flags>Empty</Flags>
      <Score>1</Score>
      <SourceID>0</SourceID>
      <Text>About</Text>
      <Token>0</Token>
    </Annotation>
    <Annotation i:type="Word">
      <Flags>Empty</Flags>
      <Score>1</Score>
      <SourceID>0</SourceID>
      <Text>oceanography</Text>
      <Token>2</Token>
    </Annotation>
    <Annotation i:type="Word">
      <Flags>Contained</Flags>
      <Score>0.9</Score>
      <SourceID>1</SourceID>
      <Text>about</Text>
      <Token>0</Token>
    </Annotation>
    <Annotation i:type="Word">
      <Flags>Contained</Flags>
      <Score>0.9</Score>
      <SourceID>1</SourceID>
      <Text>oceanography</Text>
      <Token>2</Token>
    </Annotation>
    <Annotation i:type="Sentence">
      <End>3</End>
      <Offset>0</Offset>
      <Duplicate>-1</Duplicate>
    </Annotation>
    <Annotation i:type="Token">
      <End>5</End>
      <Offset>0</Offset>
      <Capitalisation>Initial Caps Lowers SentenceInitial</Capitalisation>
      <Type>Alpha HasWhitespaceLeft HasWhitespaceRight HasBoundaryLeft</Type>
    </Annotation>
    <Annotation i:type="Token">
      <End>6</End>
      <Offset>5</Offset>
      <Capitalisation>None</Capitalisation>
      <Type>White NoWhitespaceLeft NoWhitespaceRight</Type>
    </Annotation>
    <Annotation i:type="Token">
      <End>18</End>
      <Offset>6</Offset>
      <Capitalisation>Lowers</Capitalisation>
      <Type>Alpha HasWhitespaceLeft HasWhitespaceRight</Type>
    </Annotation>
  </Annotations>
</TextAnalysis>
```

Request Headers

- Authorization – HTTP Basic authentication with provided credentials

Status Codes

- 200 OK – no error

3.2.1 Annotation Types

Most important fields in the output are the following:

1. `ConceptAnnotation/Afreq` : absolute frequency (count) of concept in the text
2. `ConceptAnnotation/ConceptID` : ID of concept in original source thesaurus/vocab
3. `ConceptAnnotation/Rank` : Rank of concept in case different section types are involved (title, abstract, etc.)
4. `ConceptAnnotation/Thesaurus` : thesaurus that was used for indexing
5. `TermAnnotation/ConceptID` : Concept the term belongs to
6. `TermAnnotation/TermID` : ID of term, as in the original thesaurus or controlled vocabulary if present, otherwise generated
7. `TermAnnotation/Thesaurus` : thesaurus that was used for indexing
8. `TermAnnotation/Tokens` : array element(s) of token annotation. 0 refers to first token, 1 refers to second token, n-1 refers to nth token
9. `Token/Offset` : between character position of token start (0 refers to position before 1st character)
10. `Token/End` : between character position of token end

A standard response contains all annotation types, and as a result can become quite large (tens of MB). At the same time, many applications only require Concept annotations. To reduce the response size and network traffic needed, each plain workflow has an `ConceptsOnly` version. For instance, whereas calling the plain text `MathVoc` workflow (`/Taco7900/TacoService.svc/MathVoc`) will return all annotation types, calling the `MathConceptsOnly` workflow (`/Taco7900/TacoService.svc/MathVocConceptsOnly`) only returns Concept annotations.

3.2.2 Thesauri & Vocabularies

The workflows (thesauri & vocabularies) that are currently available for indexing and and can be encoded in the URL (`/Taco[version]/TacoService.svc/[workflow]`) are the following:

- `CpxTree` : Elsevier's Compendex (COMPuterized ENgineering inDEX) thesaurus for Engineering & Materials Science, Chemistry.
- `Chemistry` : Composite thesaurus covering the Chemistry domain. Composed (on the fly) of MeSH's chemistry concepts (Semantic group Chemicals and Drugs = MeSH's core [D] branch + MeSH Supplementary concepts) and Elsevier's Compendex concepts (see below); contains chemicals and related concepts.
- `Economics` : Elsevier's FPE-generated vocabulary covering the Economics domain.
- `Embase` : Elsevier's EMTREE thesaurus support the bibliographic database EMBASE, covering the Life Sciences domain.

- `Geobase` : Elsevier's Earth Sciences thesaurus + GEMET - GEneral Multilingual Environmental Thesaurus, covering the Earth Sciences domain.
- `Gesis` : Gesis Thesaurus for the Social Sciences, covering the Social Sciences domain.
- `Humanities` : Elsevier's FPE-generated vocabulary covering the Humanities domain.
- `KeyPhrases` : Elsevier's FPE-generated vocabulary, multidisciplanary, complementary to other thesauri and vocabularies.
- `MathVoc` : University of Cambridge Connecting Mathematics Thesaurus + Elsevier's FPE-generated vocabulary, covering the Mathematics domain.
- `MeSH` : National Library of Medicine's MeSH Medical Subject Headings incl. MeSH Supplemental terms, covering the Medical Sciences domain.
- `NAL` : NALT National Agricultural Library Agricultural Thesaurus, covering the Agronomics domain.
- `NASA` : National Aeronautics and Space Administration (NASA) Thesaurus, covering the Physics domain.
- `Unified` : Virtual resource, combining concepts and terms of forementioned thesauri and vocabularies. Excludes KeyPhrase.

3.3 Sectioned indexing

If the input text or document is divided into sections, and it is important to be able to determine which concept occurred (how often) in what section, the Sectioned workflows can be used. Each plain text workflow has a `XmlSectioned` version.

Given a payload with XML, a root node, and multiple sub-nodes, annotations of type `SectionConceptAnnotation` will be generated for each section concept. The section in which the concept was found is denoted with the `<Section>` tag.

Example request:

```
POST /Taco7900/TacoService.svc/MeSHXmlSectioned HTTP/1.1
Host: fingerprintengine.scivalcontent.com
Accept: text/xml
```

```
<Xml>
  <Title>
    Photosynthesis
  </Title>
  <Definition>
    A process by which plants use energy from sunlight to produce glucose.
  </definition>
</Xml>
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/xml
```

```
<TextAnalysis xmlns="http://www.collexis.com/annotations/" xmlns:i="http://www.w3.
 ↪org/2001/XMLSchema-instance">
  <Annotations>
    <Annotation i:type="SectionConceptAnnotation">
      <AFreq>1</AFreq>
      <ConceptID>16761</ConceptID>
      <Name>Photosynthesis</Name>
      <Rank>1</Rank>
      <Thesaurus>MeSH</Thesaurus>
      <Section>title</Section>
    </Annotation>
    <Annotation i:type="SectionConceptAnnotation">
      <AFreq>1</AFreq>
      <ConceptID>16967</ConceptID>
      <Name>Plants</Name>
      <Rank>1</Rank>
      <Thesaurus>MeSH</Thesaurus>
      <Section>definition</Section>
    </Annotation>
    <Annotation i:type="SectionConceptAnnotation">
      <AFreq>1</AFreq>
      <ConceptID>20815</ConceptID>
      <Name>Sunlight</Name>
      <Rank>1</Rank>
      <Thesaurus>MeSH</Thesaurus>
      <Section>definition</Section>
    </Annotation>
    <Annotation i:type="SectionConceptAnnotation">
      <AFreq>1</AFreq>
      <ConceptID>9339</ConceptID>
      <Name>Glucose</Name>
      <Rank>1</Rank>
      <Thesaurus>MeSH</Thesaurus>
      <Section>definition</Section>
    </Annotation>
  </Annotations>
</TextAnalysis>
```

3.4 Bulk indexing

All endpoints listed so far expect the payload to be plain text to be indexed. It is possible to have multiple documents indexed in a single call. The maximum payload size to send in this way is 25MB.

Data provided to bulk indexing endpoints are encoded with XML, where the texts to be indexed are contained by `<Document>` nodes which each go with an `ID` attribute. The `Document` node can contain a single or multiple nodes called either `Title`, `Abstract`, or `Text`.

```
<BulkAnalysis>
  <Document ID="1">
    <Text>...</Text>
  </Document>
  <Document ID="2">
    <Text>...</Text>
  </Document>
</BulkAnalysis>
```

The response of bulk indexing requests follow a similar format:

```
<BulkAnalysis>
  <Document>
    <ID>1</ID>
```

(continues on next page)

(continued from previous page)

```
    <TextAnalysis>
      <Annotations>...</Annotations>
    </TeztAnalysis>
  </Document>
  <Document>
    <ID>2</ID>
    <TextAnalysis>
      <Annotations>...</Annotations>
    </TeztAnalysis>
  </Document>
</BulkAnalysis>
```

A bulk indexing request can be placed as follows:

POST /Taco[version]/Bulk.svc/[thesaurus]/Document/ID

Where *[version]* is the long (four digit) version number of the engine to use, and *[thesaurus]* is a string indicating which thesaurus to index with, **with Xml suffix**. Currently supported thesauri workflows are: `CpxTreeXml`, `ChemistryXml`, `GeobaseXml`, `MathVocXml`, `MeSHXml`, `NALXml`, `NASAXml`, `GesisXml`, `HumanitiesXml`, `EconomicsXml`, `UnifiedXml`.

Request Headers

- Authorization – HTTP Basic authentication with provided credentials

Status Codes

- 200 OK – no error
- 400 Bad Request – bad request, either because of malformed URI or incorrectly structured XML input.

Example request:

```
POST /Taco7900/Bulk.svc/UnifiedXmlConceptsOnly/Document/ID HTTP/1.1
Host: fingerprintengine.scivalcontent.com
Accept: text/xml
```

```
<BulkAnalysis>
  <Document ID="1">
    <Text>cell</Text>
  </Document>
  <Document ID="2">
    <Text>photosynthesis</Text>
  </Document>
</BulkAnalysis>
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/xml
```

```
<BulkAnalysis>
  <Document>
    <ID>1</ID>
    <TextAnalysis xmlns="http://www.collexis.com/annotations/" xmlns:i="http://www.w3.
 ↪org/2001/XMLSchema-instance">
      <Annotations>
        <Annotation i:type="ConceptAnnotation">
          <AFreq>1</AFreq>
```

(continues on next page)

(continued from previous page)

```
          <ConceptID>17332</ConceptID>
          <Name>Cells</Name>
          <Rank>1</Rank>
          <Thesaurus>Unified</Thesaurus>
        </Annotation>
      </Annotations>
    </TextAnalysis>
  </Document>
  <Document>
    <ID>2</ID>
    <TextAnalysis xmlns="http://www.collexis.com/annotations/" xmlns:i="http://www.w3.
↪org/2001/XMLSchema-instance">
      <Annotations>
        <Annotation i:type="ConceptAnnotation">
          <AFreq>1</AFreq>
          <ConceptID>16822</ConceptID>
          <Name>Photosynthesis</Name>
          <Rank>1</Rank>
          <Thesaurus>Unified</Thesaurus>
        </Annotation>
      </Annotations>
    </TextAnalysis>
  </Document>
</BulkAnalysis>
```

3.5 Thesaurus structure and elements

3.5.1 Concept Hierarchy

Vocabularies generally do not contain hierarchical structure (broader/narrower relationships), but most thesauri do. This structure can be obtained with the following API call:

POST /Taco[version]/Query.svc/Hierarchy/[thesaurus]

Where *[version]* is the long (four digit) version number of the engine to use, and *[thesaurus]* is a string indicating which thesaurus to index with. Currently supported thesauri strings are: `CpxTree`, `Chemistry`, `Geobase`, `Math`, `MeSH`, `NAL`, `NASA`, `Gesis`, `Humanities`, `Economics`, `Unified`.

Request Headers

- Authorization – HTTP Basic authentication with provided credentials

Status Codes

- 200 OK – no error
- 400 Bad Request – bad request, because of malformed URI

Example request:

```
POST /Taco7900/Query.svc/Hierarchy/NAL HTTP/1.1
Host: fingerprintengine.scivalcontent.com
Accept: <empty>
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/xml
```

```
<Result>
  <ConceptHierarchy ParentConceptID="2" ChildConceptID="6331"/>
  <ConceptHierarchy ParentConceptID="2" ChildConceptID="6335"/>
  <ConceptHierarchy ParentConceptID="2" ChildConceptID="6570"/>
  ...
  <ConceptHierarchy ParentConceptID="4" ChildConceptID="12044"/>
  <ConceptHierarchy ParentConceptID="4" ChildConceptID="12046"/>
  <ConceptHierarchy ParentConceptID="4" ChildConceptID="18665"/>
  <ConceptHierarchy ParentConceptID="4" ChildConceptID="20409"/>
  <ConceptHierarchy ParentConceptID="4" ChildConceptID="26546"/>
  <ConceptHierarchy ParentConceptID="4" ChildConceptID="34802"/>
  <ConceptHierarchy ParentConceptID="4" ChildConceptID="35623"/>
  <ConceptHierarchy ParentConceptID="4" ChildConceptID="37703"/>
  <ConceptHierarchy ParentConceptID="4" ChildConceptID="187364"/>
  <ConceptHierarchy ParentConceptID="7" ChildConceptID="9"/>
  <ConceptHierarchy ParentConceptID="7" ChildConceptID="10"/>
  <ConceptHierarchy ParentConceptID="7" ChildConceptID="11"/>
  <ConceptHierarchy ParentConceptID="8" ChildConceptID="7"/>
  <ConceptHierarchy ParentConceptID="8" ChildConceptID="259"/>
  ...
</Result>
```

As can be seen from the example, the output contains hierarchical stucture encoded by ConceptHierarchy elements with Parent – Child relations by ConceptIDs.

3.5.2 Concept List

Information on the concepts involved can then be retrieved with the following call:

`POST /Taco[version]/Query.svc/Concepts/[thesaurus]`

Where *[version]* is the long (four digit) version number of the engine to use, and *[thesaurus]* is a string indicating which thesaurus to index with. Currently supported thesauri strings are: `CpxTree`, `Chemistry`, `Geobase`, `Math`, `MeSH`, `NAL`, `NASA`, `Gesis`, `Humanities`, `Economics`, `Unified`. Multiple Concept IDs can be queried by providing a comma separated list of IDs.

Request Headers

- Authorization – HTTP Basic authentication with provided credentials

Status Codes

- 200 OK – no error
- 400 Bad Request – bad request, either because of malformed URI or incorrectly structured input.

Example request:

```
POST /Taco7900/Query.svc/Concepts/NAL HTTP/1.1
Host: fingerprintengine.scivalcontent.com
Accept: text/plain

7,9,10,11
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/xml
```

```
<Result>
  <Concept ID="7" Name="acids, bases, and salts">
    <Terms/>
  </Concept>
  <Concept ID="9" Name="acids">
    <Terms/>
  </Concept>
  <Concept ID="10" Name="chemical bases">
    <Terms/>
  </Concept>
  <Concept ID="11" Name="salts">
    <Terms/>
  </Concept>
  <Query>7,9,10,11</Query>
</Result>
```

Finally, the full concept list can be retrieved with the following query (warning: the response can be hundreds of MB depending on thesaurus size, and the response duration may exceed 2 minutes)

POST /Taco[version]/Query.svc/ConceptList/[thesaurus]

Where *[version]* is the long (four digit) version number of the engine to use, and *[thesaurus]* is a string indicating which thesaurus to index with. Currently supported thesauri strings are: `CpxTree`, `Chemistry`, `Geobase`, `Math`, `MeSH`, `NAL`, `NASA`, `Gesis`, `Humanities`, `Economics`, `Unified`. The XML response that will be returned XML contains all concepts in the thesaurus and their child concepts.

Request Headers

- Authorization – HTTP Basic authentication with provided credentials

Status Codes

- 200 OK – no error
- 400 Bad Request – bad request, because of malformed URI

Example request:

```
POST /Taco7900/Query.svc/ConceptList/NAL HTTP/1.1
Host: fingerprintengine.scivalcontent.com
Accept: <empty>
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/xml
```

```
<Result>
  <Concept ID="2" Name="positive sense, single-stranded RNA viruses" Idf="11.
↪4391929544611" IgnoredConcept="N" SuspiciousConcept="N">
    <Terms>
      <Term>single-stranded RNA viruses positive sense</Term>
      <Term>positive sense single-stranded RNA viruses</Term>
      <Term>positive-sense RNA viruses</Term>
      <Term>(+)ssRNA viruses</Term>
```

(continues on next page)

(continued from previous page)

```
    </Terms>
    <ChildConcepts>
      <ChildConcept>15882</ChildConcept>
      <ChildConcept>21188</ChildConcept>
      ...
      <ChildConcept>6689</ChildConcept>
    </ChildConcepts>
    <ParentConcepts>
      <ParentConcept>276283</ParentConcept>
    </ParentConcepts>
  </Concept>
  <Concept ID="4" Name="negative sense, single-stranded RNA viruses" Idf="11.
↪7223218648034" IgnoredConcept="N" SuspiciousConcept="N">
    <Terms>
      <Term>negative sense single-stranded RNA viruses</Term>
      <Term>(-)ssRNA viruses</Term>
      <Term>single-stranded RNA viruses negative sense</Term>
    </Terms>
    <ChildConcepts>
      <ChildConcept>12046</ChildConcept>
      <ChildConcept>187364</ChildConcept>
      <ChildConcept>18665</ChildConcept>
      <ChildConcept>34802</ChildConcept>
      <ChildConcept>35623</ChildConcept>
      <ChildConcept>26546</ChildConcept>
      <ChildConcept>20409</ChildConcept>
      <ChildConcept>12044</ChildConcept>
      <ChildConcept>37703</ChildConcept>
    </ChildConcepts>
    <ParentConcepts>
      <ParentConcept>276283</ParentConcept>
    </ParentConcepts>
  </Concept>
</Result>
```

3.5.3 Ignore Concepts

For each thesaurus, a number of concepts are either particularly generic or ambiguous. Depending on the purpose of the concept annotations, it may be desirable to filter them out post-indexing by the client.

The concepts marked as candidates to possibly ignore can then be retrieved with the following call:

GET /Taco[version]/Query.svc/IgnoreConcepts/[thesaurus]

Where *[version]* is the long (four digit) version number of the engine to use, and *[thesaurus]* is a string indicating which thesaurus to index with.

Request Headers

- Authorization – HTTP Basic authentication with provided credentials

Status Codes

- 200 OK – no error

Example request:

```
GET /Taco7900/Query.svc/IgnoreConcepts/NAL HTTP/1.1
Host: fingerprintengine.scivalcontent.com
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/xml
```

```
<Result>
    <Concept ID="10"/>
    <Concept ID="110"/>
    <Concept ID="123"/>
    <Concept ID="183"/>
    <Concept ID="186"/>
    <Concept ID="192"/>
    <Concept ID="197"/>
    <Query/>
</Result>
```

3.6 Domain classification

The Elsevier Fingerprint Engine exposes a text classifier that ingests plain text, classifies it into the domains / thesauri supported, and returns a ranking that reveals the most relevant domain(s) for the text input.

3.6.1 Endpoint

POST /Taco[version]/TacoService.svc/DomainClassifier

Where *[version]* is the long (four digit) version number of the engine to use. The XML response that will be returned XML contains a ranking of all available thesauri/domains.

Request Headers

- Authorization – HTTP Basic authentication with provided credentials

Status Codes

- 200 OK – no error
- 400 Bad Request – bad request, because of malformed URI

Example request:

```
POST /Taco7900/TacoService.svc/DomainClassifier HTTP/1.1
Host: fingerprintengine.scivalcontent.com
Accept: text/plain

"Proteins are very large molecules - macro-biopolymers - made from monomers called␣
 ↪amino acids."
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/xml
```

```
<TextAnalysis xmlns="http://www.collexis.com/annotations/" xmlns:i="http://www.w3.org/
 ↪2001/XMLSchema-instance">
  <Annotations>
```

(continues on next page)

(continued from previous page)

```
    <Annotation i:type="Category">
      <Name>CHM</Name>
      <Rank>0.886719</Rank>
    </Annotation>
    <Annotation i:type="Category">
      <Name>PHY</Name>
      <Rank>0.0957031</Rank>
    </Annotation>
    <Annotation i:type="Category">
      <Name>HUM</Name>
      <Rank>0.00976564</Rank>
    </Annotation>
    <Annotation i:type="Category">
      <Name>ECO</Name>
      <Rank>0.00390627</Rank>
    </Annotation>
    <Annotation i:type="Category">
      <Name>MSH</Name>
      <Rank>1.95313E-08</Rank>
    </Annotation>
  </Annotations>
</TextAnalysis>
```

The domain classifier ranks all available categories according to descending goodness of fit: the first assigned category is the best fit, the second assigned category the one but best, et cetera. The scores are the result of the softmax function, which means that taken altogether they sum up to 1. Thresholding can be applied to identify those classes that stand out: in the example discussed, Chemistry (CHM) is the top ranked class with a rank score that is substantially higher than those of the classes that follow. Note that extremely small values can be shown with scientific notation, as shown with the last category listed (`MSH : 1.95313E-08`).

3.6.2 Domain abbreviations

The domain classifier returns annotations that map as follows to thesauri/vocabularies and scientific domains.

```
1.  CHM     Chemistry (Chemistry)
2.  CPX     CpxTree (Engineering & Materials Sciences)
3.  ECO     Economics (Economics)
4.  GEO     Geobase (Earth Sciences)
5.  HUM     Humanities (Humanities)
6.  MAT     MathVoc (Mathematics)
7.  MSH     Mesh (Medical Sciences)
8.  NAL     NAL (Agronomics)
9.  PHY     NASA (Physics)
10. SOC     Gesis (Social Sciences)
```

3.6.3 Multi-disciplinarity

In case the top-ranked categories suggest that the text input involves multiple disciplines, an additional top-ranked label will be inserted:

```
MD      Multi-disciplanary
```

This label can either be ignored or be used to flag clearly multidisciplanary text. An example of such output could look as follows:

```
<Annotation i:type="Category">
    <Name>MD</Name>
    <Rank>0.99414114</Rank>
</Annotation>
<Annotation i:type="Category">
    <Name>MSH</Name>
    <Rank>0.501953</Rank>
</Annotation>
<Annotation i:type="Category">
    <Name>CPX</Name>
    <Rank>0.494141</Rank>
</Annotation>
<Annotation i:type="Category">
    <Name>CHM</Name>
    <Rank>0.00195314</Rank>
</Annotation>
```

4 Frequently Asked Questions

4.1 General

4.1.1 Are there restrictions when using the API? Like limitation of # of calls per day?

There are no restrictions when using the API.

4.1.2 Can the API be used to analyze Scopus data?

Yes. Scopus data can be analyzed. The Elsevier Fingerprint Engine has been optimized to process scientific text.

4.1.3 What formats does the fingerprinting engine accept?

Plain text or XML.

4.1.4 Can I use my own vocabulary or thesaurus?

The API supports a standard set of thesauri and controlled vocabularies only. If there is a need to use a custom thesaurus or vocabulary and make that available at the API, this can be requested. Only English language is currently supported. The turn around time for Importing, Testing, and Rolling out to API is on average 3 to 4 weeks.

4.1.5 Can I classify with my own scheme of categories?

The API supports categorization for Scientific Domains only. A new classification / categorization model can be developed on request if needed.

HTTP Routing Table

/Taco[version]

```
GET /Taco[version]/Query.svc/IgnoreConcepts/[thesaurus],
      15
POST /Taco[version]/Bulk.svc/[thesaurus]/Document/ID,
      11
POST /Taco[version]/Query.svc/ConceptList/[thesaurus],
      14
POST /Taco[version]/Query.svc/Concepts/[thesaurus],
      13
POST /Taco[version]/Query.svc/Hierarchy/[thesaurus],
      12
POST /Taco[version]/TacoService.svc/DomainClassifier,
      16
POST /Taco[version]/TacoService.svc/[thesaurus],
      5
```