Code Sprint: Track A

Extraction of Bibliographical Data and Citations from PDF Applying GROBID

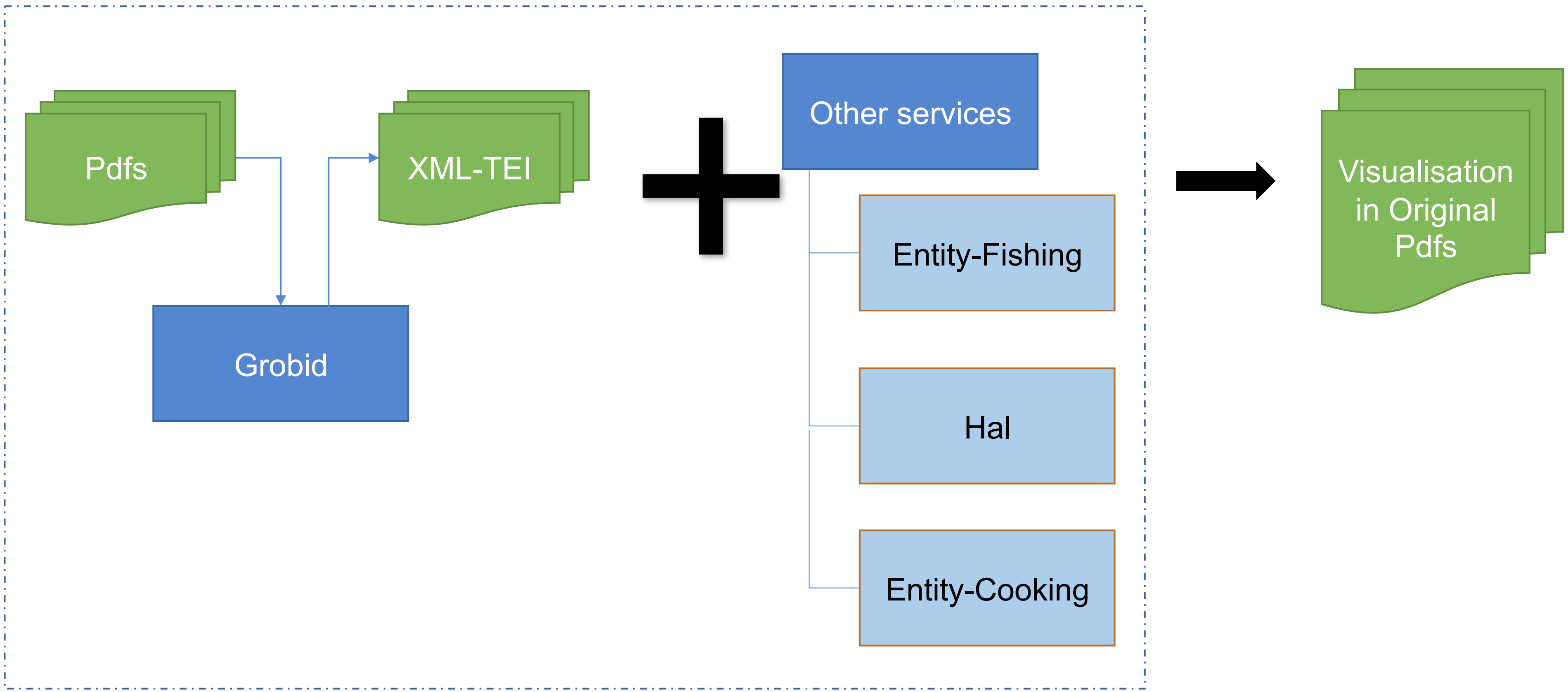
What is Grobid?

- GROBID (or Grobid) stands for GeneRation Of BIbliographic Data.
- Machine learning library for extracting, parsing and re-structuring raw documents such as PDF into structured TEI-encoded documents.
  - Focus on technical and scientific publications.
- Functionalities:
  - Extraction and parsing of header, references, and full text from Pdf articles;
  - Parsing of names (e.g. person title, fornames, middlename), dates, affiliations;
  - Manages 55 final labels from traditional publication metadata to full text structures:
    - Ex. of traditional publication metadata: title, author first/last/middlenames, affiliation types, detailed address, journal, volume, issue, pages, etc
    - Ex. of full text structures: section title, paragraph, reference markers, head/foot notes, figure headers, etc.

4 Tasks for this Code Sprint

- **Extraction** of citation data using Grobid as a tool;
- **Visualisation** of extracted information on PDF files using Grobid as a library;
- **Visualisation** of extracted information collected from external services on PDF files;
- **Development** of enhanced and usable PDF viewers using extracted information from more services as input *)

General Ideas



0. Preparation

- PDF documents have been already prepared
  - 50 files in 5 languages
  - https://github.com/DESIR-CodeSprint/trackA-kickoff/tree/master/data/pdf
- Grobid's documentation :
  - http://grobid.readthedocs.io
- Install, build, and run Grobid
- Possibilities of using Grobid:
  - Web service mode (simple and efficient way)
  - Batch mode
  - Java API

Get the Grobid Source

- *Download* and *unzip* the latest stable release version (0.5.1) :

  $ wget https://github.com/kermitt2/grobid/archive/0.5.1.zip
- (OR) *Clone* the development version :

  $ git clone https://github.com/kermitt2/grobid.git
- (OR) Use the *docker* container: (http://grobid.readthedocs.io/en/latest/Grobid-docker/)

Build the Grobid

- Using gradle

  $ ./gradlew clean install

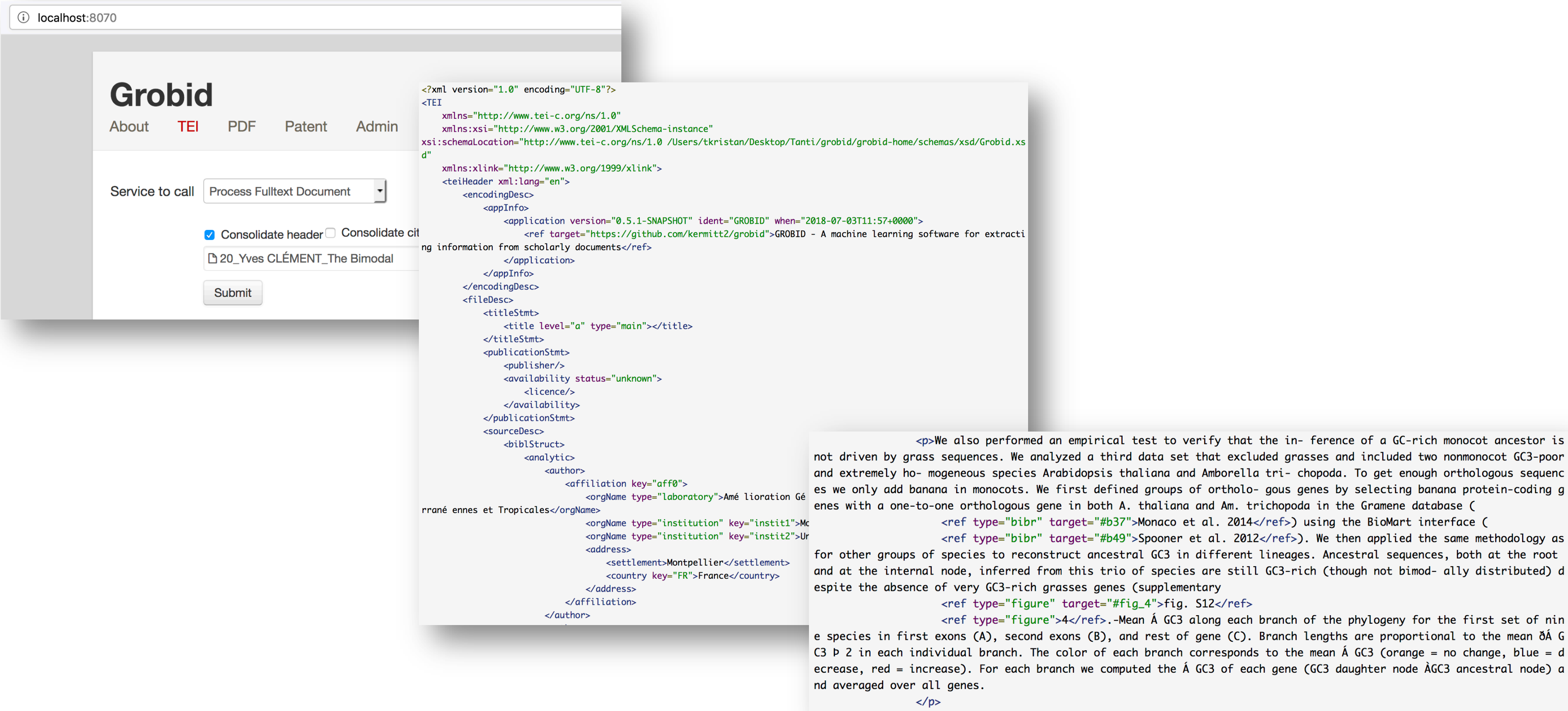
```
grobid
├── grobid-core
├── grobid-home
├── grobid-service
├── grobid-trainer
```

```
grobid-home
├── build
├── config
├── language-detection
├── lexicon
├── lib
├── models
├── pdf2xml
├── schemas
└── tmp
```

Run and Use the Grobid

- Start the server with Gradle
  `$ ./gradlew run`
- **8070** is a default port
  - Restful API is under http://localhost:8070
  - For service check: http://localhost:8070/api/version or http://localhost:8070/api/isalive
  - For starting the server on a *different port* or for changing the absolute path : `/grobid/grobid-service/config/config.yaml`
- **grobid-home** directory (/grobid/grobid-home) contains all the models and static resources required to run Grobid

Task 1: Extraction of Citation Data



TEI format

- [TEI header]
- [Elements in TEI documents]

TEI Header

- File description **<fileDesc>**
  - Bibliographical description of the computer file
- Encoding description **<encodingDesc>**
  - Relationship between an electronic text and its sources
- Text profile **<profileDesc>**
  - Classificatory and contextual information about the text
- Container element **<xenoData>**
  - Inclusion of metadata from non-TEI schemes
- Revision history **<revisionDesc>**
  - History of changes

```
<teiHeader>
 <fileDesc>
  <titleStmt>
   <title>
<!-- title of the resource -->
   </title>
  </titleStmt>
  <editionStmt>
   <p>
<!-- information about the edition of the
              resource  -->
   </p>
  </editionStmt>
  <extent>
<!-- description of the size of the resource -->
  </extent>
  <publicationStmt>
   <p>
<!-- information about the distribution
                 of the resource -->
   </p>
  </publicationStmt>
  <seriesStmt>
   <p>
<!-- information about any series to which
                the resource belongs  -->
   </p>
  </seriesStmt>
  <notesStmt>
   <note>
<!-- notes on other aspects of the resource -->
   </note>
  </notesStmt>
  <sourceDesc>
   <p>
<!-- information about the source from which
                 the resource was derived  -->
   </p>
  </sourceDesc>
 </fileDesc>
</teiHeader>
```

Elements in TEI Documents

- Paragraphs
- Treatment of punctuation
- Highlighting and quotation
- Editorial changes
- Names, numbers, dates, abbreviations, and addresses
- Links and cross-references
- Lists
- Notes, annotation, and indexing
- Graphics and non-textual components
- References


```xml
<?xml version="1.0" encoding="UTF-8"?>
<TEI
    xmlns="http://www.tei-c.org/ns/1.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.tei-c.org/ns/1.0 /home/lopez/grobid/grobid-home/schemas/xsd/Grobid.xsd"
    xmlns:xlink="http://www.w3.org/1999/xlink">
    <teiHeader xml:lang="en">
        <encodingDesc>
            <appInfo>
                <application version="0.5.1-SNAPSHOT" ident="GROBID" when="2018-06-04T15:00+0000">
                    <ref target="https://github.com/kermitt2/grobid">GROBID - A machine learning software for extracting
                        information from scholarly documents</ref>
                </application>
            </appInfo>
        </encodingDesc>
        <fileDesc>
            <titleStmt>
                <title level="a" type="main">G Gaifman-Locality ▸ Locality of Queries Gazetteers</title>
            </titleStmt>
            <publicationStmt>
                <publisher/>
                <availability status="unknown">
                    <licence/>
                </availability>
            </publicationStmt>
            <sourceDesc>
                <biblStruct>
                    <analytic>
                        <author>
                            <persName
                                xmlns="http://www.tei-c.org/ns/1.0">
                                <forename type="first">Linda</forename>
                                <forename type="middle">L</forename>
                                <surname>Hill</surname>
                            </persName>
                            <affiliation key="aff0">
                                <orgName type="institution">University of California-Santa Barbara</orgName>
                                <address>
                                    <settlement>Santa Barbara</settlement>
                                    <region>CA</region>
                                    <country key="US">USA</country>
                                </address>
                            </affiliation>
                        </author>
                        <title level="a" type="main">G Gaifman-Locality ▸ Locality of Queries Gazetteers</title>
                    </analytic>
                    <monogr>
                        <imprint>
                            <date/>
                        </imprint>
                    </monogr>
                </biblStruct>
            </sourceDesc>
        </fileDesc>
        <profileDesc>
            <abstract/>
        </profileDesc>
    </teiHeader>
    <text xml:lang="en"></text>
</TEI>
```

Web Service in Grobid

- Extract the header of a PDF document

  $ curl -v --form input=@./File1.pdf localhost:8070/api/processHeaderDocument
- Process full text document
  - curl -v --form input =@./File1.pdf localhost:8070/api/processFulltextDocument
- Extract all the bibliographical references and convert it into TEI XML format
  - curl -v --form input =@./File1.pdf localhost:8070/api/processReferences


```xml
<?xml version="1.0" encoding="UTF-8"?>
<TEI
    xmlns="http://www.tei-c.org/ns/1.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.tei-c.org/ns/1.0 /Users/tkristan/Desktop/Tanti/grobid/grobid-home/schemas/xsd/Grobid.xsd"
    xmlns:xlink="http://www.w3.org/1999/xlink">
    <teiHeader xml:lang="en">
        <encodingDesc>
            <appInfo>
                <application version="0.5.1-SNAPSHOT" ident="GROBID" when="2018-07-03T14:36+0000">
                    <ref target="https://github.com/kermitt2/grobid">GROBID - A machine learning software for extracting
                        information from scholarly documents</ref>
                </application>
            </appInfo>
        </encodingDesc>
        <fileDesc>
            <titleStmt>
                <title level="a" type="main">The influence of catch trials on the consolidation of motor memory in force
                    field adaptation tasks</title>
            </titleStmt>
            <publicationStmt>
                <publisher>Frontiers Media SA</publisher>
                <availability status="unknown">
                    <p>Copyright Frontiers Media SA</p>
                </availability>
                <date type="published" when="2013-07-25">published: 25 July 2013</date>
            </publicationStmt>
            <sourceDesc>
                <biblStruct>
                    <analytic>
                        <author>
                            <persName
                                xmlns="http://www.tei-c.org/ns/1.0">
                                <forename type="first">Anne</forename>
                                <surname>Focke</surname>
                            </persName>
                        </author>
                        <author>
                            <persName
                                xmlns="http://www.tei-c.org/ns/1.0">
                                <forename type="first">Christian</forename>
                                <surname>Stockinger</surname>
                            </persName>
                        </author>
                        <author>
                            <persName
                                xmlns="http://www.tei-c.org/ns/1.0">
                                <forename type="first">Christina</forename>
                                <surname>Diepold</surname>
                            </persName>
                        </author>
                        <author>
                            <persName
                                xmlns="http://www.tei-c.org/ns/1.0">
                                <forename type="first">Marco</forename>
                                <surname>Taubert</surname>
                            </persName>
                        </author>
                        <author>
```

Task 2: Visualisation of Information Extracted by Grobid in Original Pdf

DeepType: Multilingual Entity Linking by Neural Type System Evolution

Jonathan Raiman
OpenAI
San Francisco, California
raiman@openai.com

Olivier Raiman
Agilience
Paris, France
or@agilience.com

Abstract

The wealth of structured (e.g. Wikidata) and unstructured data about the world available today presents an incredible opportunity for tomorrow's Artificial Intelligence. So far, integration of these two different modalities is a difficult process, involving many decisions concerning how best to represent the information so that it will be captured or useful, and hand-labeling large amounts of data. DeepType overcomes this challenge by explicitly integrating symbolic information into the reasoning process of a neural network with a type system. First we construct a type system, and second, we use it to constrain the outputs of a neural network to respect the symbolic structure. We achieve this by reformulating the design problem into a mixed integer problem: create a type system and subsequently train a neural network with it. In this reformulation discrete variables select which parent-child relations from an ontology are types within the type system, while continuous variables control a classifier fit to the type system. The original problem cannot be solved exactly, so we propose a 2-step algorithm: 1) heuristic search or stochastic optimization over discrete variables that define a type system informed by an Oracle and a Learnability heuristic, 2) gradient descent to fit classifier parameters. We apply DeepType to the problem of Entity Linking on three standard datasets (i.e. WikiDisamb30, CoNLL (YAGO), TAC KBP 2010) and find that it outperforms all existing solutions by a wide margin,

2017), a loss function that trades off specificity for accuracy by incorporating hypo/hypernymy relations (Deng et al. 2012), using NER types to constrain the behavior of an Entity Linking system (Ling, Singh, and Weld 2015), or more recently integrating explicit type constraints within a decoder's grammar for neural semantic parsing (Krishnamurthy, Dasigi, and Gardner 2017). However, current approaches face several difficulties:

- Selection of the right symbolic information based on the utility or information gain for a target task.
- Design of the representation for symbolic information (hierarchy, grammar, constraints).
- Hand-labelling large amounts of data.

DeepType overcomes these difficulties by explicitly integrating symbolic information into the reasoning process of a neural network with a type system that is automatically designed without human effort for a target task. We achieve this by reformulating the design problem into a mixed integer problem: create a type system by selecting roots and edges from an ontology that serve as types in a type system, and subsequently train a neural network with it. The original problem cannot be solved exactly, so we propose a 2-step algorithm:

Task 3: Integration with external services (Abstract / Title)

DeepType: Multilingual Entity Linking by Neural Type System Evolution

DeepType: Multilingual Entity Linking by Neural Type System Evolution

OpenAI
San Francisco, California
raiman@openai.com

Agilience
Paris, France
or@agilience.com

Abstract

The wealth of structured (e.g. Wikidata) and unstructured data about the world available today presents an incredible opportunity for tomorrow's Artificial Intelligence. So far, integration of these two different modalities is a difficult process, involving many decisions concerning how best to represent the information so that it will be captured or useful, and hand-labeling large amounts of data. DeepType overcomes this challenge by explicitly integrating symbolic information into the reasoning process of a neural network with a type system. First we construct a type system, and second, we use it to constrain the outputs of a neural network to respect the symbolic structure. We achieve this by reformulating the design problem into a mixed integer problem: create a type system and subsequently train a neural network with it. In this reformulation discrete variables select which parent-child relations from an ontology are types within the type system, while continuous variables control a classifier fit to the type system. The original problem cannot be solved exactly, so we propose a 2-step algorithm: 1) heuristic search or stochastic optimization over discrete variables that define a type system informed by an Oracle and a Learnability heuristic, 2) gradi-

informed by an Oracle and ... ent descent to fit classifier ... the problem of Entity Lin... WikiDisamb30, CoNLL (... that it outperforms all ex...

2017), a loss function that trades off specificity for accuracy by incorporating hypo/hypernymy relations (Deng et al. 2012), using NER types to constrain the behavior of an Entity Linking system (Ling, Singh, and Weld 2015), or ...ating explicit type constraints within a ... for neural semantic parsing (Krishna... Gardner 2017). However, current ap... ...l difficulties:

...ight symbolic information based on the ...ion gain for a target task.

...esentation for symbolic information (hi... constraints).

...ge amounts of data.

...mes these difficulties by explicitly in... information into the reasoning process ... with a type system that is automati... ...out human effort for a target task. We ...ormulating the design problem into a ...em: create a type system by selecting ... an ontology that serve as types in a type ...ently train a neural network with it. The ...not be solved exactly, so we propose a

Task 3: Integration with external services (Authors / Affiliations)

DeepType: Multilingual Entity Linking by Neural Type System Evolution

HAL id: 123456

Jonathan Raiman
OpenAI
San Francisco, California
raiman@openai.com

Olivier Raiman
Agilience
Paris, France
or@agilience.com

HAL id: 12345

Abstract

The wealth of structured (e.g. Wikidata) and unstructured data about the world available today presents an incredible opportunity for tomorrow's Artificial Intelligence. So far, integration of these two different modalities is a difficult process, involving many decisions concerning how best to represent the information so that it will be captured or useful, and hand-labeling large amounts of data. DeepType overcomes this challenge by explicitly integrating symbolic information into the reasoning process of a neural network with a type system. First we construct a type system, and second, we use it to constrain the outputs of a neural network to respect the symbolic structure. We achieve this by reformulating the design problem into a mixed integer problem: create a type system and subsequently train a neural network with it. In this reformulation discrete variables select which parent-child relations from an ontology are types within the type system, while continuous variables control a classifier fit to the type system. The original problem cannot be solved exactly, so we propose a 2-step algorithm: 1) heuristic search or stochastic optimization over discrete variables that define a type system informed by an Oracle and a Learnability heuristic, 2) gradient descent to fit classifier parameters. We apply DeepType to the problem of Entity Linking on three standard datasets (i.e. WikiDisamb30, CoNLL (YAGO), TAC KBP 2010) and find that it outperforms all existing solutions by a wide margin,

2017), a loss function that trades off specificity for accuracy by incorporating hypo/hypernymy relations (Deng et al. 2012), using NER types to constrain the behavior of an Entity Linking system (Ling, Singh, and Weld 2015), or more recently integrating explicit type constraints within a decoder's grammar for neural semantic parsing (Krishnamurthy, Dasigi, and Gardner 2017). However, current approaches face several difficulties:

- Selection of the right symbolic information based on the utility or information gain for a target task.
- Design of the representation for symbolic information (hierarchy, grammar, constraints).
- Hand-labelling large amounts of data.

DeepType overcomes these difficulties by explicitly integrating symbolic information into the reasoning process of a neural network with a type system that is automatically designed without human effort for a target task. We achieve this by reformulating the design problem into a mixed integer problem: create a type system by selecting roots and edges from an ontology that serve as types in a type system, and subsequently train a neural network with it. The original problem cannot be solved exactly, so we propose a 2-step algorithm:

Extraction of Information from External Services

Examples of information collected from Entity-Fishing (http://cloud.science-miner.com/nerd/service/)

- Text disambiguation
- Term disambiguation
- Features collection from wikidata


```
{
  "text": "The text to be processed.",
  "shortText": "term1 term2 ...",
  "termVector": [
    {
      "term": "term1",
      "score": 0.3
    },
    {
      "term": "term2",
      "score": 0.1
    }
  ],
  "language": {
    "lang": "en"
  },
  "entities": [],
  "mentions": [
    "ner",
    "wikipedia"
  ],
  "nbest": 0,
  "sentence": false,
  "customisation": "generic",
  "processSentence": []
}
```

Sources and Bibliography

- https://github.com/kermitt2/grobid
- http://grobid.readthedocs.io/en/latest/Grobid-service/
- https://github.com/istex/grobid-istex
- http://www.tei-c.org/Guidelines/P5/
- http://cloud.science-miner.com/nerd/service/