



The Luxembourg BabelNet Workshop

2 March 2016: Session 3

# Tech session



Disambiguating text with Babelfy.  
The Babelfy API

Claudio Delli Bovi



# Outline

**Multilingual disambiguation with Babelfy**

**Using Babelfy**

**How to query Babelfy programmatically:**

**HTTP and Java APIs**

**The Babelfy Java API: Download and set up**

**The Babelfy Java API: Main classes**

**Usage example**



# Outline

Multilingual disambiguation with Babelfy

Using Babelfy

How to query Babelfy programmatically:

HTTP and Java APIs

The Babelfy Java API: Download and set up

The Babelfy Java API: Main classes

Usage example

Technical part!



# Multilingual disambiguation with Babelfy



**Babelfy** is a joint approach to multilingual word sense disambiguation and entity linking powered by BabelNet

- It leverages the BabelNet network and represents the semantic interpretations of an ambiguous sentence using a graph.
- Then it extracts the *densest subgraph* (=most coherent interpretation)!

# Multilingual disambiguation with Babelfy



**Babelfy** is a joint approach to multilingual word sense disambiguation and entity linking powered by BabelNet

- It leverages the BabelNet network and represents the semantic interpretations of an ambiguous sentence using a graph.
- Then it extracts the *densest subgraph* (=most coherent interpretation)!

Gory details here:

A. Moro, A. Raganato, R. Navigli. **Entity Linking meets Word Sense Disambiguation: a Unified Approach**. Transactions of the Association for Computational Linguistics (TACL), 2, pp. 231-244, 2014.

# Using Babelfy

LOG IN REGISTER



Babelfy

Text to babelfy...

Enable partial matches:

ENGLISH

BABELFY!



ABOUT  
PUBLICATIONS  
DOWNLOADS  
API GUIDE

Babelfy is an output of the [MultiJEDI ERC Starting Grant](#) No. 259234. Concept and application by [Andrea Moro](#) and [Roberto Navigli](#). Babelfy and its API are licensed under a [Creative Commons Attribution-Non Commercial-Share Alike 3.0 License](#). For any commercial use, please [contact us](#).



# Using Babelfy



Babelfy

Nintendo announces new details on Mario Kart 8.

Enable partial matches:

ENGLISH

BABELFY

[expanded view](#) | [compact view](#)

Nintendo

announces

new

details

on

Mario Kart 8



**Nintendo**

Nintendo is a Japanese multinational consumer electronic...

**announces**

Make known; make an announcement



**new**

Not of long duration; having just (or relatively recently) come into being or...

**details**

A small part that can be considered separately from the whole



**Mario Kart 8**

Mario Kart 8 is a 2014 kart racing game and the eighth major installment in the...

**Kart**

Kart racing or karting is a variant of open-wheel motorsport with small, open, four-...

# Using Babelfy



Babelfy

Thomas and Mario are strikers playing in Munich.

Enable partial matches:

ENGLISH

BABELFY!

[expanded view](#) | [compact view](#)

Thomas

and

Mario

are

strikers

playing

in

Munich



**Thomas**

Thomas Müller is a German footballer who plays for Bayern Munich and the



**Mario**

Mario Gómez García is a German footballer who plays as a striker for



**strikers**

A forward on a soccer team

**playing**

Shoot or hit in a particular manner



**Munich**

Fußball-Club Bayern München e.V., commonly known as FC Bayern München,



# Using Babelfy



Babelfy

BabelNet is both a dizionario enciclopedico multilingüe und a reseau semantique.

Enable partial matches:

AGNOSTIC

BABELFY

expanded view | compact view

BabelNet

is

both a dizionario enciclopedico

multilingüe

und a reseau semantique



**BabelNet**

BabelNet is a multilingual lexicalized semantic network and ontology.

is

Form or compose



**dizionario**

A reference book containing an alphabetical list of words with informati...

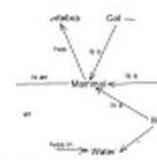
**dizionario enciclopedico**

An encyclopedic dictionary typically includes a large number of short...



**multilingüe**

The ability to speak two languages colloquially



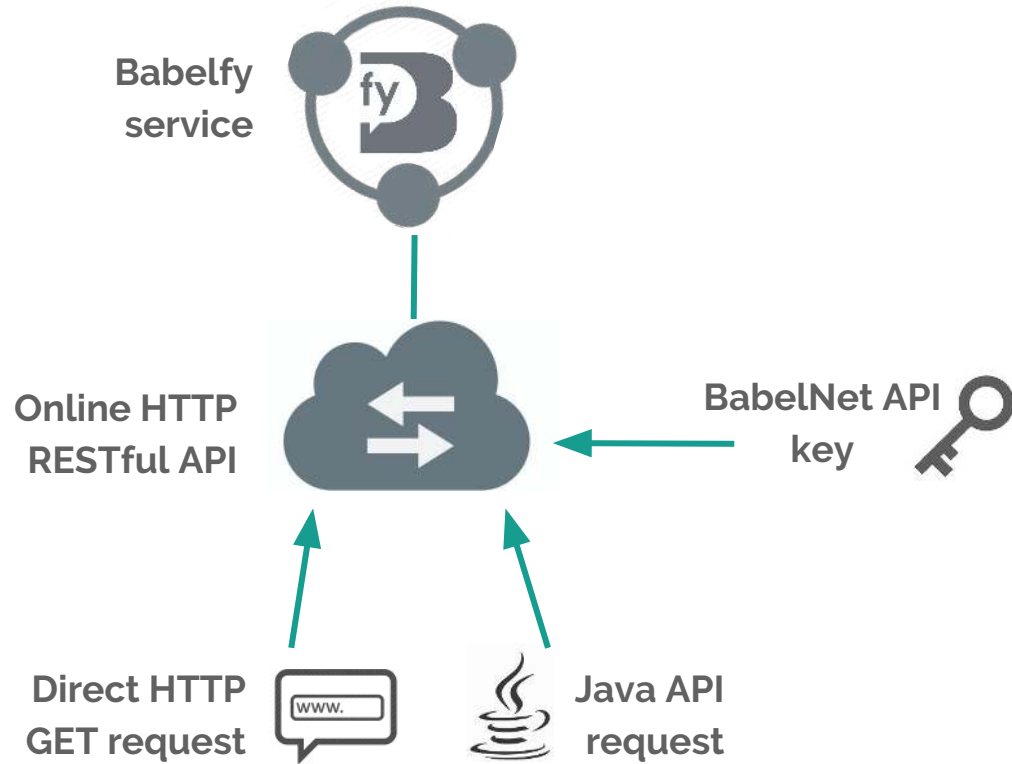
**reseau**

A net or mesh foundation for lace

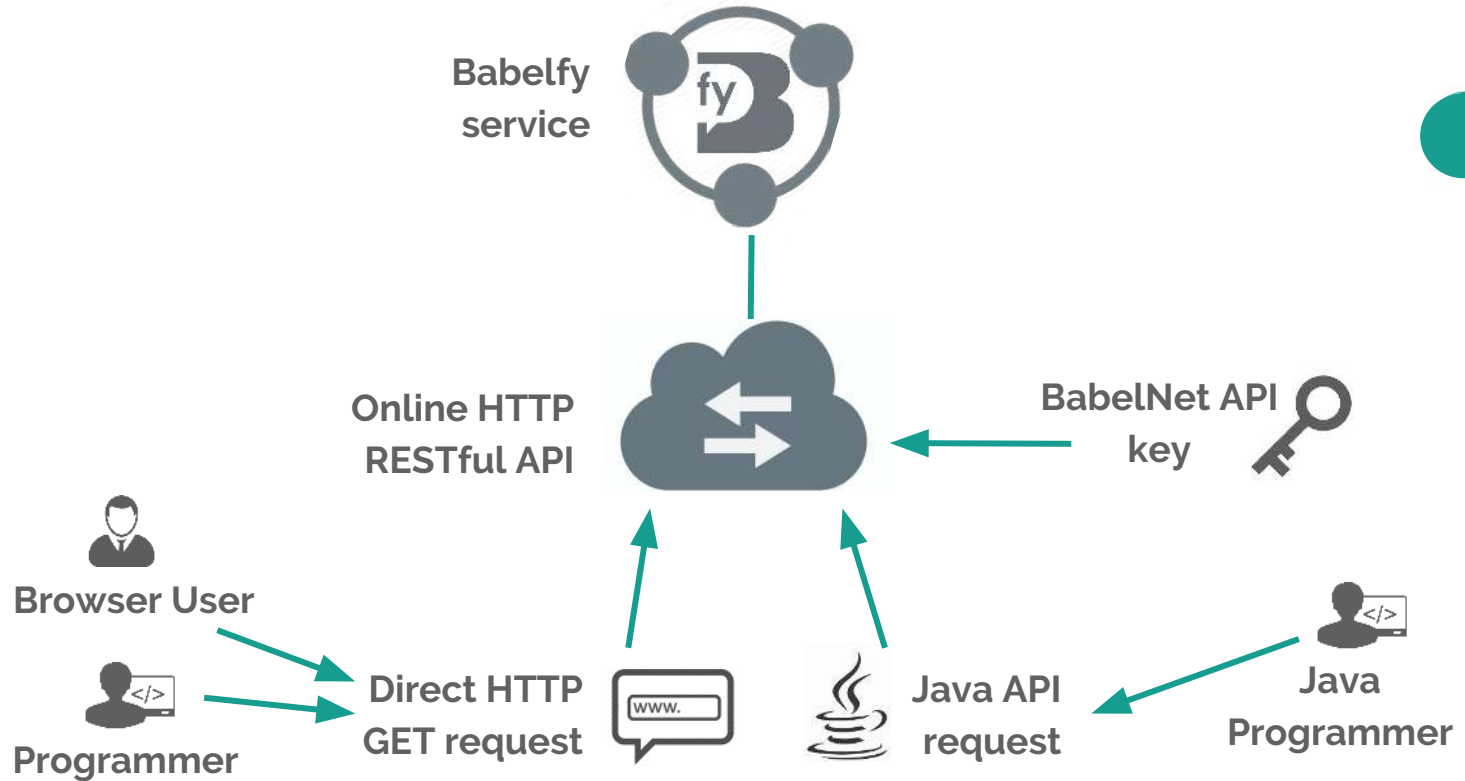
**reseau semantique**

A semantic network, or frame network, is a network which represents semanti...

# Using Babelfy... programmatically



# Using Babelfy... programmatically



# Using Babelfy... programmatically

The BabelNet and Babelfy APIs use the very same key.

If you already registered an account on BabelNet, **no** need to register again: just log in with the same credentials!

Otherwise:

[babelnet.org/register](https://babelnet.org/register)



# Using Babelfy... programmatically

The BabelNet and Babelfy APIs use the very same key.

If you already registered an account on BabelNet, **no** need to register again: just log in with the same credentials!

Otherwise:

[babelnet.org/register](https://babelnet.org/register)

The Babelfy API also relies on **Babelcoins** to track user requests:

**1** Babelcoin = **1** query to BabelNet or Babelfy

Base account: **1000** Babelcoins per day



# The HTTP and Java APIs



# The HTTP and Java APIs

Like BabelNet, Babelfy can be queried programmatically via an HTTP RESTful interface that returns JSON.

You just have to append a **key** parameter to the HTTP request.



# The HTTP and Java APIs

Like BabelNet, Babelfy can be queried programmatically via an HTTP RESTful interface that returns JSON.

You just have to append a **key** parameter to the HTTP request.

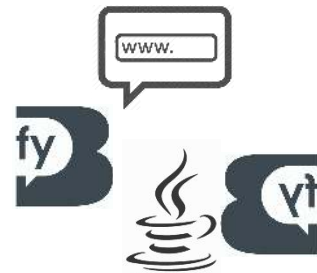
The Babelfy Java API provides a **Java binding** to the online HTTP RESTful service with classes, types and methods to query Babelfy for disambiguation from inside a Java program.

## Only requirement:

Standard installation of **Java JDK** (version  $\geq 1.7$ )

## Detailed Javadoc:

[babelfy.org/javadoc](http://babelfy.org/javadoc)





Technical part ahead!



# Downloading and installing instructions





## Babelfy RESTful API

 [Babelfy RESTful Java API](#) version 1.0 (April 2015 - Size: 2M)

*The [Babelfy Java API](#) is an extension of our online [HTTP RESTful service](#). It provides classes to work with Babelfy. If you would rather use the raw HTTP API, please read the [HTTP guide](#).*

The **legacy API v0.9** has been shutdown as of June 1, 2015.

[ABOUT](#) [PUBLICATIONS](#) [DOWNLOADS](#) [API GUIDE](#)





## Babelify RESTful API

[ABOUT](#) [PUBLICATIONS](#) [DOWNLOADS](#) [API GUIDE](#)

 [Babelify RESTful Java API](#) version 1.0 (April 2015 - Size: 2M)

The [Babelify Java API](#) is an extension of our online [HTTP RESTful service](#). It provides classes to work with Babelify. If you would rather use the raw HTTP API, please read the [HTTP guide](#).

The **legacy API v0.9** has been shutdown as of June 1, 2015.

Java API 





Babelfy RESTful Java API version 1.0 (April 2015 - Size: 2M)

The *Babelfy Java API* is an extension of our online [HTTP RESTful service](#). It provides classes to work with Babelfy. If you would rather use the raw HTTP API, please read the [HTTP guide](#).

The legacy API v0.9 has been shutdown as of June 1, 2015.

Java API



**Download** and **unpack** the package: **BabelfyAPI-1.0.zip**

You will find the following:

babelfy-online-1.0.jar

config

docs

lib

run-babelfydemo.sh

README

CHANGELOG

LICENSE

run-babelfydemo.bat



 Babelfy RESTful Java API version 1.0 (April 2015 - Size: 2M)

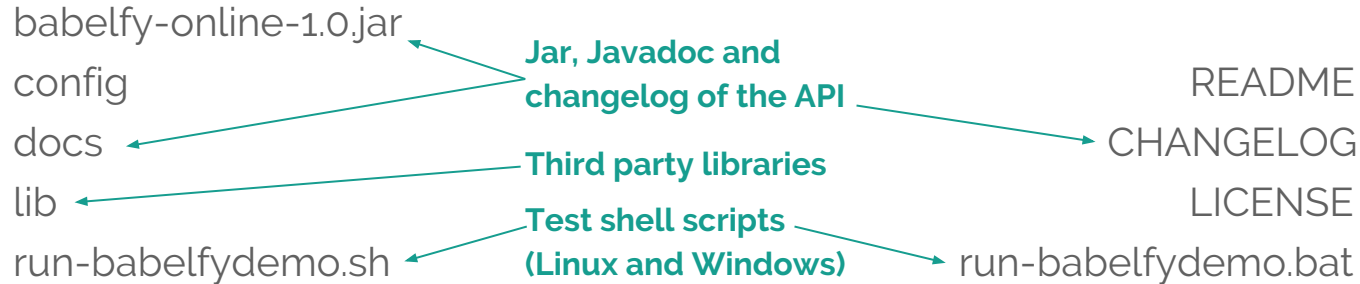
The *Babelfy Java API* is an extension of our online *HTTP RESTful service*. It provides classes to work with Babelfy. If you would rather use the raw HTTP API, please read the *HTTP guide*.

The legacy API v0.9 has been shutdown as of June 1, 2015.



**Download** and **unpack** the package: **BabelfyAPI-1.0.zip**

You will find the following:





Babelfy RESTful Java API version 1.0 (April 2015 - Size: 2M)

The *Babelfy Java API* is an extension of our online *HTTP RESTful service*. It provides classes to work with Babelfy. If you would rather use the raw HTTP API, please read the *HTTP guide*.

The legacy API v0.9 has been shutdown as of June 1, 2015.

Java API



**Download** and **unpack** the package: **BabelfyAPI-1.0.zip**

You will find the following:

babelfy-online-1.0.jar

config

docs

lib

run-babelfydemo.sh

**README file**

**configuration files**

**License of the API**

README

CHANGELOG

LICENSE

run-babelfydemo.bat



# Downloading and installing instructions

Same easy steps to set up and test the API:





# Downloading and installing instructions

Same easy steps to set up and test the API:

1. Specify a **valid key** in the "babelfy.key" property inside the configuration file `config/babelfy.var.properties`



# Downloading and installing instructions

Same easy steps to set up and test the API:

1. Specify a **valid key** in the "babelfy.key" property inside the configuration file `config/babelfy.var.properties`
2. Test the API with the corresponding shell script:

```
run-babelfydemo.sh
```

```
run-babelfydemo.bat
```



Linux



Windows



# Configuring the API on Eclipse/Netbeans

Assuming you have your Java (or Scala) project in the workspace of your favourite IDE under `projectDir/`:

1. Copy (or link) the `config/` directory from the API folder into `projectDir/`;



# Configuring the API on Eclipse/Netbeans

Assuming you have your Java (or Scala) project in the workspace of your favourite IDE under `projectDir/`:

1. Copy (or link) the `config/` directory from the API folder into `projectDir/`;
2. Include the third-party libraries (`lib/*.jar`) and the API itself (`babelfy-online-1.0.jar`) in the project build classpath;



# Configuring the API on Eclipse/Netbeans

Assuming you have your Java (or Scala) project in the workspace of your favourite IDE under `projectDir/`:

1. Copy (or link) the `config/` directory from the API folder into `projectDir/`;
2. Include the third-party libraries (`lib/*.jar`) and the API itself (`babelfy-online-1.0.jar`) in the project build classpath;



Find the project in the package explorer view → Project → Properties → Java build path → Libraries → Add external JARs



Find the project in the left tree view → Properties → Categories → Libraries → compile → Add JAR/Folder



# Configuring the API on Eclipse/Netbeans

Assuming you have your Java (or Scala) project in the workspace of your favourite IDE under `projectDir/`:

1. Copy (or link) the `config/` directory from the API folder into `projectDir/`;
2. Include the third-party libraries (`lib/*.jar`) and the API itself (`babelfy-online-1.0.jar`) in the project build classpath;
3. Include the `config/` directory in the project build classpath;



# Configuring the API on Eclipse/Netbeans

Assuming you have your Java (or Scala) project in the workspace of your favourite IDE under `projectDir/`:

1. Copy (or link) the `config/` directory from the API folder into `projectDir/`;
2. Include the third-party libraries (`lib/*.jar`) and the API itself (`babelfy-online-1.0.jar`) in the project build classpath;
3. Include the `config/` directory in the project build classpath;



Find the project in the package explorer view → Project → Properties → Java build path → Source → Add Folder



Find the project in the left tree view → Properties → Categories → Libraries → compile → Add JAR/Folder (same as before)



# The Java API: main classes





# The Java API: main classes

## **Babelfy**

The **Babelfy** class is used as entry point to access all disambiguation functions available in Babelfy. It extends the **IBabelfy** interface.



# The Java API: main classes

## **Babelfy**

The **Babelfy** class is used as entry point to access all disambiguation functions available in Babelfy. It extends the **IBabelfy** interface.

## **SemanticAnnotation**

The **SemanticAnnotation** class models Babelfy's response objects, i.e. token-based disambiguation results (fragment of text + disambiguation).



# The Java API: main classes

## **Babelfy**

The **Babelfy** class is used as entry point to access all disambiguation functions available in Babelfy. It extends the **IBabelfy** interface.

## **SemanticAnnotation**

The **SemanticAnnotation** class models Babelfy's response objects, i.e. token-based disambiguation results (fragment of text + disambiguation).

## **BabelfyToken**

A **BabelfyToken** is a token unit that can be used to build *custom* input sentences for Babelfy. Each **BabelfyToken** stores information about its language and may be associated with constraints (**BabelfyConstraints**)



# The Java API: Babelfy

The `Babelfy` class is used as entry point to access all the disambiguation functions available in Babelfy. You can create a `Babelfy` object by simply calling its default constructor:

```
Babelfy bfy = new Babelfy();
```



# The Java API: Babelfy

The `Babelfy` class is used as entry point to access all the disambiguation functions available in Babelfy. You can create a `Babelfy` object by simply calling its default constructor:

```
Babelfy bfy = new Babelfy();
```

Babelfy's disambiguation setting can be modified in various ways. When you create a `Babelfy` object you can specify different behaviors using the `BabelfyParameters` class as input for the constructor:

```
Babelfy bfy = new Babelfy(BabelfyParameters bp);
```



# The Java API: BabelfyParameters

The `BabelfyParameters` class provides a set of dedicated methods to specify disambiguation parameters for the Babelfy call:



# The Java API: BabelifyParameters

The `BabelifyParameters` class provides a set of dedicated methods to specify disambiguation parameters for the Babelify call:

- **`setAnnotationResource`**: allows the user to restrict the disambiguated entries to only WordNet or Wikipedia;



# The Java API: BabelifyParameters

The `BabelifyParameters` class provides a set of dedicated methods to specify disambiguation parameters for the Babelify call:

- **`setAnnotationResource`**: allows the user to restrict the disambiguated entries to only WordNet or Wikipedia;
- **`setAnnotationType`**: allows the user to restrict disambiguation to only named entities or only word senses;





# The Java API: BabelifyParameters

The `BabelifyParameters` class provides a set of dedicated methods to specify disambiguation parameters for the Babelify call:

- **setAnnotationResource**: allows the user to restrict the disambiguated entries to only WordNet or Wikipedia;
- **setAnnotationType**: allows the user to restrict disambiguation to only named entities or only word senses;
- **setDensestSubgraph**: enables or disables the *densest subgraph* heuristic;



# The Java API: BabelfyParameters

The `BabelfyParameters` class provides a set of dedicated methods to specify disambiguation parameters for the Babelfy call:

- **setAnnotationResource**: allows the user to restrict the disambiguated entries to only WordNet or Wikipedia;
- **setAnnotationType**: allows the user to restrict disambiguation to only named entities or only word senses;
- **setDensestSubgraph**: enables or disables the *densest subgraph* heuristic;
- **setMatchingType**: selects the candidates extraction strategy;



# The Java API: BabelifyParameters

The `BabelifyParameters` class provides a set of dedicated methods to specify disambiguation parameters for the Babelify call:

- **setAnnotationResource**: allows the user to restrict the disambiguated entries to only WordNet or Wikipedia;
- **setAnnotationType**: allows the user to restrict disambiguation to only named entities or only word senses;
- **setDensestSubgraph**: enables or disables the *densest subgraph* heuristic;
- **setMatchingType**: selects the candidates extraction strategy;
- **setMCS**: enables or disables the *most common sense* back-off;



# The Java API: BabelifyParameters

The `BabelifyParameters` class provides a set of dedicated methods to specify disambiguation parameters for the Babelify call:

- **setAnnotationResource**: allows the user to restrict the disambiguated entries to only WordNet or Wikipedia;
- **setAnnotationType**: allows the user to restrict disambiguation to only named entities or only word senses;
- **setDensestSubgraph**: enables or disables the *densest subgraph* heuristic;
- **setMatchingType**: selects the candidates extraction strategy;
- **setMCS**: enables or disables the *most common sense* back-off;
- **setPosTaggingOptions**: sets options for the POS-tagging phase;



# The Java API: BabelifyParameters

The `BabelifyParameters` class provides a set of dedicated methods to specify disambiguation parameters for the Babelify call:

- **setAnnotationResource**: allows the user to restrict the disambiguated entries to only WordNet or Wikipedia;
- **setAnnotationType**: allows the user to restrict disambiguation to only named entities or only word senses;
- **setDensestSubgraph**: enables or disables the *densest subgraph* heuristic;
- **setMatchingType**: selects the candidates extraction strategy;
- **setMCS**: enables or disables the *most common sense* back-off;
- **setPosTaggingOptions**: sets options for the POS-tagging phase;
- **setScoredCandidates**: defines whether to return just the top ranked candidate or all candidates for a fragment of text;



# The Java API: BabelifyParameters

The `BabelifyParameters` class provides a set of dedicated methods to specify disambiguation parameters for the Babelify call:

- **setAnnotationResource**: allows the user to restrict the disambiguated entries to only WordNet or Wikipedia;
- **setAnnotationType**: allows the user to restrict disambiguation to only named entities or only word senses;
- **setDensestSubgraph**: enables or disables the *densest subgraph* heuristic;
- **setMatchingType**: selects the candidates extraction strategy;
- **setMCS**: enables or disables the *most common sense* back-off;
- **setPosTaggingOptions**: sets options for the POS-tagging phase;
- **setScoredCandidates**: defines whether to return just the top ranked candidate or all candidates for a fragment of text;
- **setThreshold**: sets the disambiguation confidence threshold;
- ...



# The Java API: BabelifyParameters

`setMatchingType` selects the candidates extraction strategy:

```
public enum MatchingType
{
    /**
     * Only exact matches are considered for disambiguation
     */
    EXACT_MATCHING,

    /**
     * Both exact and partial matches (e.g. Thomas for Thomas Muller)
     * are considered for disambiguation
     */
    PARTIAL_MATCHING
}
```



# The Java API: BabelifyParameters

`setPosTaggingOptions` sets options for the POS-tagging phase:

```
public enum PosTaggingOptions
{
    /**
     * Standard POS tagging process.
     */
    STANDARD,

    /**
     * Interprets all adjectives as nouns.
     */
    NOMINALIZE_ADJECTIVES,

    /**
     * Interprets input fragment words as nouns.
     */
    INPUT_FRAGMENTS_AS_NOUNS,

    /**
     * Tokenize the input string by splitting all characters as single tokens (all tagged as
     * nouns, so that we can disambiguate nouns). This should be used for languages and texts in
     * which there is no standard word separator such as spaces and punctuation marks.
     */
    CHAR_BASED_TOKENIZATION_ALL_NOUN,
}
```





# The Java API: BabelfyParameters

The `BabelfyParameters` class provides a set of dedicated methods to specify disambiguation parameters for the Babelfy call.

Create a `BabelParameters` object

Use the public methods of `BabelParameters` to specify the preferred setting

```
BabelfyParameters bp = new BabelfyParameters();
bp.setAnnotationResource(SemanticAnnotationResource.BN);
bp.setMCS(MCS.ON_WITH_STOPWORDS);
bp.setScoredCandidates(ScoredCandidates.ALL);
Babelfy bfy = new Babelfy(bp);
```

Initialize a `Babelfy` object with the `BabelParameters` object as input



# The Java API: BabelfyToken

The `BabelfyToken` class enables you to provide to Babelfy with a custom-tokenized text, specifying each token individually.



# The Java API: BabelfyToken

The `BabelfyToken` class enables you to provide to Babelfy with a custom-tokenized text, specifying each token individually.

**Why would I need to do it?**



# The Java API: BabelfyToken

The `BabelfyToken` class enables you to provide to Babelfy with a custom-tokenized text, specifying each token individually.

## Why would I need to do it?

Each `BabelfyToken` has its own *word*, *lemma*, *POS tag* and *language*, allowing the user to generate an arbitrary text with multiple languages at the same time.

BabelNet is both a dizionario enciclopedico



multilingüe und ein reseau semantique



# The Java API: BabelfyToken

The `BabelfyToken` class enables you to provide to Babelfy with a custom-tokenized text, specifying each token individually.

```
List<String> myEnText = Arrays.asList("java", "bytecode");
List<String> myFrText = Arrays.asList("programme", "informatique");
List<BabelfyToken> tokenizedInput = new ArrayList<>();

//building tokens for the English text
for (String word : myEnText)
    tokenizedInput.add(new BabelfyToken(word, Language.EN));

//add an EOS token to separate the different texts
tokenizedInput.add(BabelfyToken.EOS);

//building tokens for the French text
for (String word : myFrText)
    tokenizedInput.add(new BabelfyToken(word, Language.FR));
```



# The Java API: BabelifyToken

First we add **English** tokens  
"java" and "bytecode"

Add a **separator** (EOS) to tell  
Babelify not to mix tokens in  
different languages

```
List<String> myEnText = Arrays.asList("java", "bytecode");
List<String> myFrText = Arrays.asList("programme", "informatique");
List<BabelifyToken> tokenizedInput = new ArrayList<>();

//building tokens for the English text
for (String word : myEnText)
    tokenizedInput.add(new BabelifyToken(word, Language.EN));

//add an EOS token to separate the different texts
tokenizedInput.add(BabelifyToken.EOS);

//building tokens for the French text
for (String word : myFrText)
    tokenizedInput.add(new BabelifyToken(word, Language.FR));
```

Then we add **French** tokens "programme" and "informatique"



# The Java API: IBabelfy

The `IBabelfy` interface (implemented by the `Babelfy` class) exposes various overloads of the main `babelfy` call.



# The Java API: IBabelfy

The `IBabelfy` interface (implemented by the `Babelfy` class) exposes various overloads of the main `babelfy` call.

The basic ones are:

```
List<SemanticAnnotation> babelfy(String, Language)
```

```
List<SemanticAnnotation> babelfy(List<? extends  
BabelfyToken>, Language)
```





# The Java API: IBabelfy

The `IBabelfy` interface (implemented by the `Babelfy` class) exposes various overloads of the main `babelfy` call.

The basic ones are:

```
List<SemanticAnnotation> babelfy(String, Language)
```

```
List<SemanticAnnotation> babelfy(List<? extends BabelfyToken>, Language)
```

Input text (either raw or  
tokenized)



# The Java API: IBabelfy

The `IBabelfy` interface (implemented by the `Babelfy` class) exposes various overloads of the main `babelfy` call.

The basic ones are:

```
List<SemanticAnnotation> babelfy(String, Language)
```

```
List<SemanticAnnotation> babelfy(List<? extends  
BabelfyToken>, Language)
```

Language of the input text (or  
language-agnostic setting)



# The Java API: SemanticAnnotation

The `SemanticAnnotation` class represents a *disambiguated fragment of text* (either a word or a multi-word expression). It stores information about the original fragment, the attached `BaseSynset`, and the disambiguation process.



# The Java API: SemanticAnnotation

The `SemanticAnnotation` class represents a *disambiguated fragment of text* (either a word or a multi-word expression). It stores information about the original fragment, the attached `BabelSynset`, and the disambiguation process:

- **`getBabelSynsetID/getBabelNetURL`**: returns the `BabelSynset` associated with the fragment as `BabelSynsetID` object/URL;
- **`getDBpediaURL`**: returns a link to the DBpedia entry associated with the selected `BabelSynset` (if any);

Disambiguation result (meaning associated to that particular fragment)



# The Java API: SemanticAnnotation

The `SemanticAnnotation` class represents a *disambiguated fragment of text* (either a word or a multi-word expression). It stores information about the original fragment, the attached `BabelSynset`, and the disambiguation process:

- **`getBabelSynsetID/getBabelNetURL`**: returns the `BabelSynset` associated with the fragment as `BabelSynsetID` object/URL;
- **`getDBpediaURL`**: returns a link to the DBpedia entry associated with the selected `BabelSynset` (if any);
- **`getCharOffsetFragment`**: returns the char-based offset of the annotation (when the input text is given as a `String`);
- **`getTokenOffsetFragment`**: returns the token-based offset of the annotation (when the input text is given as a `List<BabelfyToken>`);

Information about the disambiguated  
fragment in the input text



# The Java API: SemanticAnnotation

The `SemanticAnnotation` class represents a *disambiguated fragment of text* (either a word or a multi-word expression). It stores information about the original fragment, the attached `BabelSynset`, and the disambiguation process:

- **`getBabelSynsetID/getBabelNetURL`**: returns the `BabelSynset` associated with the fragment as `BabelSynsetID` object/URL;
- **`getDBpediaURL`**: returns a link to the DBpedia entry associated with the selected `BabelSynset` (if any);
- **`getCharOffsetFragment`**: returns the char-based offset of the annotation (when the input text is given as a `String`);
- **`getTokenOffsetFragment`**: returns the token-based offset of the annotation (when the input text is given as a `List<BabelfyToken>`);
- **`getSource`**: returns the method used to select that particular `BabelSynset` (Babelfy itself or the back-off strategy);

Disambiguation method



# The Java API: SemanticAnnotation

```
//bfyAnnotations is the result of Babelify.babelify() call
for (SemanticAnnotation annotation : bfyAnnotations)
{
    //splitting the input text using the CharOffsetFragment start and end anchors
    String frag = inputText.substring(annotation.getCharOffsetFragment().getStart(),
        annotation.getCharOffsetFragment().getEnd() + 1);
    System.out.println(frag + "\t" + annotation.getBabelSynsetID());
    System.out.println("\t" + annotation.getBabelNetURL());
    System.out.println("\t" + annotation.getDBpediaURL());
    System.out.println("\t" + annotation.getSource());
}
```



# The Java API: SemanticAnnotation

Retrieve the corresponding input fragment from the CharOffset

```
//bfyAnnotations is the result of Babelify.babelify() call
for (SemanticAnnotation annotation : bfyAnnotations)
{
    //splitting the input text using the CharOffsetFragment start and end anchors
    String frag = inputText.substring(annotation.getCharOffsetFragment().getStart(),
        annotation.getCharOffsetFragment().getEnd() + 1);
    System.out.println(frag + "\t" + annotation.getBabelSynsetID());
    System.out.println("\t" + annotation.getBabelNetURL());
    System.out.println("\t" + annotation.getDBpediaURL());
    System.out.println("\t" + annotation.getSource());
}
```

Print information about the associated BabelSynset and the disambiguation method





# The Java API: BabelfyConstraints

When you already have some information on the input text, the Babelfy API allows you to define *constraints* for the disambiguation process via the `BabelfyConstraints` class.



# The Java API: BabelfyConstraints

When you already have some information on the input text, the Babelfy API allows you to define *constraints* for the disambiguation process via the `BabelfyConstraints` class.

You can do it in two ways:

1. by specifying `SemanticAnnotations` for particular text fragments you already know how to disambiguate;

```
boolean addAnnotatedFragments(SemanticAnnotation... )
```



# The Java API: BabelifyConstraints

When you already have some information on the input text, the Babelify API allows you to define *constraints* for the disambiguation process via the `BabelifyConstraints` class.

You can do it in two ways:

1. by specifying `SemanticAnnotations` for particular text fragments you already know how to disambiguate;
2. by specifying *which* fragments of the input text you want to disambiguate.

```
boolean addFragmentToDisambiguate(TokenOffsetFragment... )
```

```
boolean addFragmentToDisambiguate(CharOffsetFragment... )
```



# The Java API: BabelfyConstraints

BabelfyConstraints works similarly to BabelfyParameters. You just have to create a BabelfyConstraints object, add your constraints using its public interface, and then pass it as input parameter for the Babelfy call:

```
Babelfy bfy = new Babelfy();
String inputText = "BabelNet is both a multilingual encyclopedic dictionary and a semantic network";
BabelfyConstraints constraints = new BabelfyConstraints();
SemanticAnnotation a = new SemanticAnnotation(new TokenOffsetFragment(0, 0), "bn:03083790n",
    "http://dbpedia.org/resource/BabelNet", Source.OTHER);
constraints.addAnnotatedFragments(a);
Babelfy bfy = new Babelfy();
List<SemanticAnnotation> bfyAnnotations = bfy.babelfy(inputText, Language.EN, constraints);
```



# The Java API: BabelifyConstraints

Initializing a `BabelifyConstraints` object

Specifying a pre-annotated fragment (i.e. the first word of the sentence is assigned the `BabelSynset bn:03083790n`)

```
Babelify bfy = new Babelify();
String inputText = "BabelNet is both a multilingual encyclopedic dictionary and a semantic network";
BabelifyConstraints constraints = new BabelifyConstraints();
SemanticAnnotation a = new SemanticAnnotation(new TokenOffsetFragment(0, 0), "bn:03083790n",
    "http://dbpedia.org/resource/BabelNet", Source.OTHER);
constraints.addAnnotatedFragments(a);
Babelify bfy = new Babelify();
List<SemanticAnnotation> bfyAnnotations = bfy.babelify(inputText, Language.EN, constraints);
```

Adding the pre-annotated fragment to the `BabelifyConstraints` object

Passing the constraint as input argument for the method `Babelify#babelify`

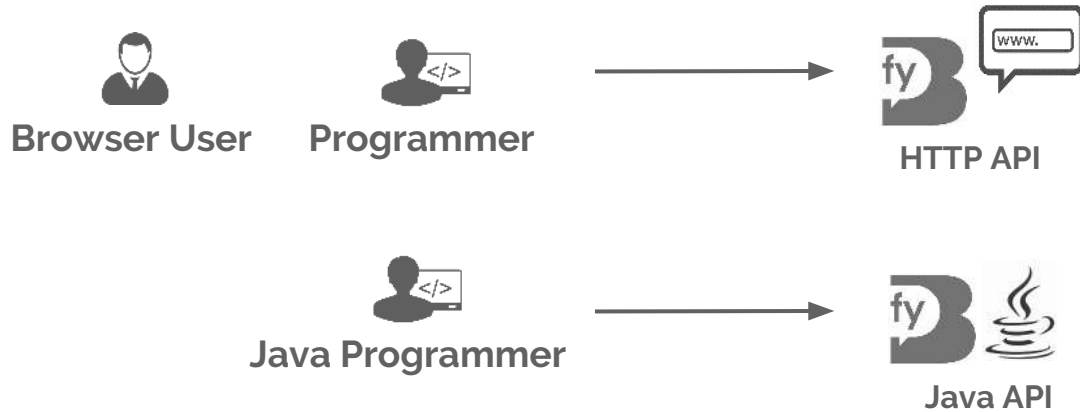


# Full usage example



# Full usage example

As in the previous session, we will look at this example from two perspectives:



# Full usage example



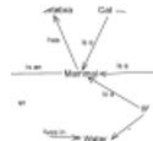
*"BabelNet is both a multilingual encyclopedic dictionary and a semantic network."*



0-0  
BabelNet  
**bn:03083790n**



5-6  
encyclopedic dictionary  
**bn:02290297n**



9-10  
semantic network  
**bn:02275757n**



# Full usage example

Basic call to the HTTP RESTful service:

**URL:** `https://babelfy.io/v1/disambiguate?  
text=text & lang=lang & key=key`

The required input parameters are the same of the Java API method `Babelfy#babelfy` (input text and language) + **the registration key**



# Full usage example

Basic call to the HTTP RESTful service:

**URL:** `https://babelfy.io/v1/disambiguate?  
text=text & lang=lang & key=key`

Call with disambiguation parameters:

**URL:** `https://babelfy.io/v1/disambiguate?  
text=text & lang=lang & annType=NAMED_ENTITIES & ...  
& match=PARTIAL_MATCHING & key=key`

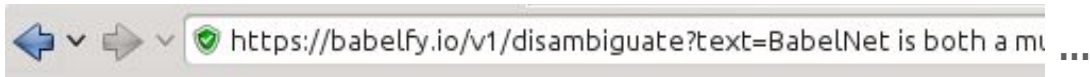
Disambiguation parameters specified in the same service call  
(complete list: <http://babelfy.org/guide#Disambiguateatext>)



HTTP API

# Full usage example

URL: `https://babelfy.io/v1/disambiguate?  
text=text & lang=lang & key=key`



```
{{"tokenFragment":{"start":0,"end":0},"charFragment":  
{"start":0,"end":7},"babelSynsetID":"bn:03083790n","DBpediaURL":"http://dbpedia.org/resource/BabelNet","BabelNetURL":  
L:"http://babelnet.org/rdf/s03083790n","score":1.0,"coherenceScore":0.6,"globalScore":0.09574468085106383,"source  
":"BABELFY"},{"tokenFragment":{"start":4,"end":4},"charFragment":  
{"start":19,"end":30},"babelSynsetID":"bn:00107021a","DBpediaURL":"","BabelNetURL":"http://babelnet.org/rdf/s00107  
021a","score":0.0,"coherenceScore":0.0,"globalScore":0.0,"source":"MCS"},{"tokenFragment":  
{"start":5,"end":5},"charFragment":  
{"start":32,"end":43},"babelSynsetID":"bn:00102202a","DBpediaURL":"","BabelNetURL":"http://babelnet.org/rdf/s00102  
202a","score":0.0,"coherenceScore":0.0,"globalScore":0.0,"source":"MCS"},{"tokenFragment":  
{"start":5,"end":6},"charFragment":  
{"start":32,"end":54},"babelSynsetID":"bn:02290297n","DBpediaURL":"http://dbpedia.org/resource/Encyclopedic_dictio  
nary","BabelNetURL":"http://babelnet.org/rdf/s02290297n","score":1.0,"coherenceScore":0.4,"globalScore":0.04255319  
14893617,"source":"BABELFY"},{"tokenFragment":{"start":6,"end":6},"charFragment":  
{"start":45,"end":54},"babelSynsetID":"bn:00026967n","DBpediaURL":"http://dbpedia.org/resource/Dictionary","BabelN  
etURL":"http://babelnet.org/rdf/s00026967n","score":0.8823529411764706,"coherenceScore":1.0,"globalScore":0.319148  
9361702128,"source":"BABELFY"},{"tokenFragment":{"start":9,"end":9},"charFragment":  
{"start":62,"end":69},"babelSynsetID":"bn:00110347a","DBpediaURL":"","BabelNetURL":"http://babelnet.org/rdf/s00110  
347a","score":1.0,"coherenceScore":0.2,"globalScore":0.010638297872340425,"source":"BABELFY"},{"tokenFragment":  
{"start":9,"end":10},"charFragment":  
{"start":62,"end":77},"babelSynsetID":"bn:02275757n","DBpediaURL":"http://dbpedia.org/resource/Semantic_network","  
BabelNetURL":"http://babelnet.org/rdf/s02275757n","score":1.0,"coherenceScore":0.6,"globalScore":0.127659574468085  
1,"source":"BABELFY"},{"tokenFragment":{"start":10,"end":10},"charFragment":  
{"start":71,"end":77},"babelSynsetID":"bn:00057379n","DBpediaURL":"","BabelNetURL":"http://babelnet.org/rdf/s00057  
379n","score":0.0,"coherenceScore":0.0,"globalScore":0.0,"source":"MCS"}]}
```



Browser User



HTTP API

# Full usage example

```
<script>
var service_url = 'https://babelfy.io/v1/disambiguate';
var text = 'BabelNet is both a multilingual encyclopedic dictionary and a semantic network';
var lang = 'EN';
var key =

var params = {
  'text' : text,
  'lang' : lang,
  'key' : key
};

$.getJSON(service_url + "?", params, function(response) {
  $.each(response, function(key, val) {
    // retrieving char fragment
    var charFragment = val['charFragment'];
    var cfStart = charFragment['start'];
    var cfEnd = charFragment['end'];

    var cfragment = "Start char fragment: " + cfStart
      + "<br/>" + "End char fragment: " + cfEnd;
    $('<div>', {html:cfragment}).appendTo(document.body);

    // retrieving annotation information
    var synsetId = val['babelSynsetId'];
    var id = "BabelNet Synset id: " + synsetId;
    $('<div>', {html:id}).appendTo(document.body);

    var synsetURL = val['BabelNetURL'];
    var url = "BabelNet URL: " + synsetURL;
    $('<div>', {html:url}).appendTo(document.body);

    var synsetSource = val['source'];
    var source = "Source: " + synsetSource;
    $('<div>', {html:source}).appendTo(document.body);

    var synsetCoherenceScore = val['coherenceScore'];
    var coherence = "Coherence Score: " + synsetCoherenceScore;
    $('<div>', {html:coherence}).appendTo(document.body);
  });
});
</script>
```

Input parameters here

Call to the service

Disambiguation  
output  
(and related  
information)



Programmer



HTTP API

# Full usage example

Start char fragment: 0  
End char fragment: 7  
BabelNet Synset id: bn:03083790n  
BabelNet URL: <http://babelnet.org/rdf/s03083790n>  
Source: BABELFY  
Coherence Score: 0.6

Start char fragment: 19  
End char fragment: 30  
BabelNet Synset id: bn:00107021a  
BabelNet URL: <http://babelnet.org/rdf/s00107021a>  
Source: MCS  
Coherence Score: 0

Start char fragment: 32  
End char fragment: 43  
BabelNet Synset id: bn:00102202a  
BabelNet URL: <http://babelnet.org/rdf/s00102202a>  
Source: MCS  
Coherence Score: 0

Start char fragment: 32  
End char fragment: 54  
BabelNet Synset id: bn:02290297n  
BabelNet URL: <http://babelnet.org/rdf/s02290297n>  
Source: BABELFY  
Coherence Score: 0.4

Start char fragment: 45  
End char fragment: 54  
BabelNet Synset id: bn:00026967n  
BabelNet URL: <http://babelnet.org/rdf/s00026967n>  
Source: BABELFY  
Coherence Score: 1

Start char fragment: 62  
End char fragment: 69  
BabelNet Synset id: bn:00110347a  
BabelNet URL: <http://babelnet.org/rdf/s00110347a>  
Source: BABELFY  
Coherence Score: 0.2

Start char fragment: 62  
End char fragment: 77  
BabelNet Synset id: bn:02275757n  
BabelNet URL: <http://babelnet.org/rdf/s02275757n>  
Source: BABELFY  
Coherence Score: 0.6

Start char fragment: 71  
End char fragment: 77  
BabelNet Synset id: bn:00057379n  
BabelNet URL: <http://babelnet.org/rdf/s00057379n>  
Source: MCS  
Coherence Score: 0



Programmer



HTTP API

# Full usage example



Start char fragment: 0  
End char fragment: 7  
BabelNet Synset id: bn:03083790n  
BabelNet URL: <http://babelnet.org/rdf/s03083790n>  
Source: BABELFY  
Coherence Score: 0.6

Start char fragment: 19  
End char fragment: 30  
BabelNet Synset id: bn:00107021a  
BabelNet URL: <http://babelnet.org/rdf/s00107021a>  
Source: MCS  
Coherence Score: 0

Start char fragment: 32  
End char fragment: 43  
BabelNet Synset id: bn:00102202a  
BabelNet URL: <http://babelnet.org/rdf/s00102202a>  
Source: MCS  
Coherence Score: 0

Start char fragment: 32  
End char fragment: 54  
BabelNet Synset id: bn:02290297n  
BabelNet URL: <http://babelnet.org/rdf/s02290297n>  
Source: BABELFY  
Coherence Score: 0.4

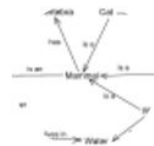
Start char fragment: 45  
End char fragment: 54  
BabelNet Synset id: bn:00026967n  
BabelNet URL: <http://babelnet.org/rdf/s00026967n>  
Source: BABELFY  
Coherence Score: 1

Start char fragment: 62  
End char fragment: 69  
BabelNet Synset id: bn:00110347a  
BabelNet URL: <http://babelnet.org/rdf/s00110347a>  
Source: BABELFY  
Coherence Score: 0.2

Start char fragment: 62  
End char fragment: 77  
BabelNet Synset id: bn:02275757n  
BabelNet URL: <http://babelnet.org/rdf/s02275757n>  
Source: BABELFY  
Coherence Score: 0.6

Start char fragment: 71  
End char fragment: 77  
BabelNet Synset id: bn:00057379n  
BabelNet URL: <http://babelnet.org/rdf/s00057379n>  
Source: MCS  
Coherence Score: 0

semantic  
network



encyclopedic  
dictionary



Programmer



HTTP API

# Full usage example

```
public class LuxTechSession2 {  
  
    public static void main(String[] args)  
    {  
        String inputText = "BabelNet is both a multilingual encyclopedic dictionary and a semantic network";  
  
        BabelfyConstraints constraints = new BabelfyConstraints();  
        SemanticAnnotation a = new SemanticAnnotation(new TokenOffsetFragment(0, 0), "bn:03083790n",  
            "http://dbpedia.org/resource/BabelNet", Source.OTHER);  
        constraints.addAnnotatedFragments(a);  
  
        BabelfyParameters bp = new BabelfyParameters();  
        bp.setAnnotationResource(SemanticAnnotationResource.BN);  
        bp.setMCS(MCS.ON_WITH_STOPWORDS);  
        bp.setScoredCandidates(ScoredCandidates.ALL);  
  
        Babelfy bfy = new Babelfy(bp);  
  
        List<SemanticAnnotation> bfyAnnotations = bfy.babelfy(inputText, Language.EN, constraints);  
  
        for (SemanticAnnotation annotation : bfyAnnotations)  
        {  
            String frag = inputText.substring(annotation.getCharOffsetFragment().getStart(),  
                annotation.getCharOffsetFragment().getEnd() + 1);  
            System.out.println(frag + "\t" + annotation.getBabelSynsetID());  
            System.out.println("\t" + annotation.getBabelNetURL());  
            System.out.println("\t" + annotation.getSource());  
            System.out.println("\tCoherence: "+annotation.getCoherenceScore());  
        }  
    }  
}
```



Programmer



Java API

# Full usage example

```
public class LuxTechSession2 {  
  
    public static void main(String[] args)  
    {  
        String inputText = "BabelNet is both a multilingual encyclopedic dictionary and a semantic network";  
        BabelfyConstraints constraints = new BabelfyConstraints();  
        SemanticAnnotation a = new SemanticAnnotation(new TokenOffsetFragment(0, 0), "bn:03083790n",  
            "http://dbpedia.org/resource/BabelNet", Source.OTHER);  
        constraints.addAnnotatedFragments(a);  
    }  
}
```

Input text (as String)

Defining a constraint: the first word of the input text is already annotated with a BabelSynset



Programmer



Java API



# Full usage example

```
public class LuxTechSession2 {  
  
    public static void main(String[] args)  
    {  
        String inputText = "BabelNet is both a multilingual encyclopedic dictionary and a semantic network";  
  
        BabelfyConstraints constraints = new BabelfyConstraints();  
        SemanticAnnotation a = new SemanticAnnotation(new TokenOffsetFragment(0, 0), "bn:03083790n",  
            "http://dbpedia.org/resource/BabelNet", Source.OTHER);  
        constraints.addAnnotatedFragments(a);  
  
        BabelfyParameters bp = new BabelfyParameters();  
        bp.setAnnotationResource(SemanticAnnotationResource.BN);  
        bp.setMCS(MCS.ON_WITH_STOPWORDS);  
        bp.setScoredCandidates(ScoredCandidates.ALL);  
  
        Babelfy bfy = new Babelfy(bp);  
  
    }  
}
```

Initialize a **Babelfy** object with the specified parameters

Specifying disambiguation parameters:

1. BabelNet as annotation resource
2. MCS back-off strategy on only with stop words
3. return all scored candidates



Programmer



Java API

# Full usage example

```
public class LuxTechSession2 {  
    public static void main(String[] args)  
    {  
        String inputText = "BabelNet is both a multilingual encyclopedic dictionary and a semantic network";
```

Call `Babelfy#babelfy` with the input text, the corresponding language and constraints

Print the resulting list of `SemanticAnnotations`

```
        List<SemanticAnnotation> bfyAnnotations = bfy.babelfy(inputText, Language.EN, constraints);  
        for (SemanticAnnotation annotation : bfyAnnotations)  
        {  
            String frag = inputText.substring(annotation.getCharOffsetFragment().getStart(),  
                annotation.getCharOffsetFragment().getEnd() + 1);  
            System.out.println(frag + "\t" + annotation.getBabelSynsetID());  
            System.out.println("\t" + annotation.getBabelNetURL());  
            System.out.println("\t" + annotation.getSource());  
            System.out.println("\tCoherence: " + annotation.getCoherenceScore());  
        }  
    }  
}
```



Programmer



Java API

# Full usage example

```
public class LuxTechSession2 {  
  
    public static void main(String[] args)  
    {  
        String inputText = "BabelNet is both a multilingual encyclopedic dictionary and a semantic network";  
  
        BabelfyConstraints constraints = new BabelfyConstraints();  
        SemanticAnnotation a = new SemanticAnnotation(new TokenOffsetFragment(0, 0), "bn:03083790n",  
            "http://dbpedia.org/resource/BabelNet", Source.OTHER);  
        constraints.addAnnotatedFragments(a);  
  
        BabelfyParameters bp = new BabelfyParameters();  
        bp.setAnnotationResource(SemanticAnnotationResource.BN);  
        bp.setMCS(MCS.ON_WITH_STOPWORDS);  
        bp.setScoredCandidates(ScoredCandidates.ALL);  
  
        Babelfy bfy = new Babelfy(bp);  
  
        List<SemanticAnnotation> bfyAnnotations = bfy.babelfy(inputText, Language.EN, constraints);  
  
        for (SemanticAnnotation annotation : bfyAnnotations)  
        {  
            String frag = inputText.substring(annotation.getCharOffsetFragment().getStart(),  
                annotation.getCharOffsetFragment().getEnd() + 1);  
            System.out.println(frag + "\t" + annotation.getBabelSynsetID());  
            System.out.println("\t" + annotation.getBabelNetURL());  
            System.out.println("\t" + annotation.getSource());  
            System.out.println("\tCoherence: "+annotation.getCoherenceScore());  
        }  
    }  
}
```



Programmer



Java API

# Full usage example

```
BabelNet      bn:03083790n
              http://babelnet.org/rdf/s03083790n
              http://dbpedia.org/resource/BabelNet
              BABELFY
multilingual  bn:00107021a
              http://babelnet.org/rdf/s00107021a
              null
              MCS
encyclopedic  bn:00102202a
              http://babelnet.org/rdf/s00102202a
              null
              MCS
encyclopedic dictionary bn:02290297n
              http://babelnet.org/rdf/s02290297n
              http://dbpedia.org/resource/Encyclopedic_dictionary
              BABELFY
dictionary    bn:00026967n
              http://babelnet.org/rdf/s00026967n
              http://dbpedia.org/resource/Dictionary
              BABELFY
semantic      bn:00110347a
              http://babelnet.org/rdf/s00110347a
              null
              BABELFY
semantic network      bn:02275757n
              http://babelnet.org/rdf/s02275757n
              http://dbpedia.org/resource/Semantic_network
              BABELFY
network       bn:00021488n
              http://babelnet.org/rdf/s00021488n
              http://dbpedia.org/resource/Computer_network
              BABELFY
```



Programmer

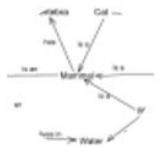


Java API

# Full usage example



BabelNet



semantic network

```
BabelNet      bn:03083790n
http://babelnet.org/rdf/s03083790n
http://dbpedia.org/resource/BabelNet
BABELFY
```

```
multilingual  bn:00107021a
http://babelnet.org/rdf/s00107021a
null
MCS
```

```
encyclopedic  bn:00102202a
http://babelnet.org/rdf/s00102202a
null
MCS
```

```
encyclopedic dictionary bn:02290297n
http://babelnet.org/rdf/s02290297n
http://dbpedia.org/resource/Encyclopedic_dictionary
BABELFY
```

```
dictionary    bn:00026967n
http://babelnet.org/rdf/s00026967n
http://dbpedia.org/resource/Dictionary
BABELFY
```

```
semantic      bn:00110347a
http://babelnet.org/rdf/s00110347a
null
BABELFY
```

```
semantic network bn:02275757n
http://babelnet.org/rdf/s02275757n
http://dbpedia.org/resource/Semantic_network
BABELFY
```

```
network       bn:00021488n
http://babelnet.org/rdf/s00021488n
http://dbpedia.org/resource/Computer_network
BABELFY
```

encyclopedic dictionary



Programmer



Java API

# Wrapping up



# Wrapping up

- Babelfy API shares the same structure of the BabelNet API:
  - **HTTP RESTful** service and corresponding **Java binding**
  - Internal credit mechanism (**Babelcoins**)

# Wrapping up

- Babelfy API shares the same structure of the BabelNet API:
  - **HTTP RESTful** service and corresponding **Java binding**
  - Internal credit mechanism (**Babelcoins**)
- The Java API defines a set of convenient classes and methods to query Babelfy for disambiguation:
  - Many different parameter settings (**BabelfyParameters**)
  - Disambiguation constraints (**BabelfyConstraints**)



# Wrapping up

- Babelfy API shares the same structure of the BabelNet API:
  - **HTTP RESTful** service and corresponding **Java binding**
  - Internal credit mechanism (**Babelcoins**)
- The Java API defines a set of convenient classes and methods to query Babelfy for disambiguation:
  - Many different parameter settings (**BabelfyParameters**)
  - Disambiguation constraints (**BabelfyConstraints**)
- Due to the multilingual nature of Babelfy, you can easily use the API to generate **custom-tokenized input text** (**BabelfyToken**) in multiple languages, and perform cross-lingual disambiguation.

Thanks

for your

attention

!

**Thanks**

An acknowledgment  
of appreciation



**attention**

The process whereby  
a person concentrates  
on some features of  
the environment to t...

fy

