

Part 1. Short manual of Bilbo

1. Usage

move to the Bilbo root directory, bilbo or in any directory.
python src/bilbo/Main.py

2. Configuration

configuration files are in KB/config/
lexicon for feature extraction, tag reorganization rules, tei data format convert rules, etc.
for more information, see configuration part in doc/README_EN.txt

3. Modification (For a detailed information, check page 6)

3-1. xml output format

File::buildReferences

3-2. add lexicon or shape features for CRF

Rule.py

Extract::_printdataWapiti <- modify "features" list (features are classified with their type)

add the feature in KB/config/features.txt

modify KB/config/wapiti/pattern_ref

****IMPORTANT****

If you added shape or lexicon features, you should change the feature list in *Extract::_printdataWapiti*. Then add template of unigram in pattern_ref.

For example, if you added 14th group of features in the feature list

in *Extract::_printdataWapiti*, activate U139:%x[-1,14], U140:%x[0,14], U141:%x[1,14] by deleting '#' at the very front. If you again add 15th group of features, add U149:%x[-1,15], U150:%x[0,15], U151:%x[1,15].

3-3. add lexicon features from a proper nouns list

Extract::_init__

Extract_crf::extract

check Name, Place, Properlist objects

add the feature in KB/config/features.txt

modify KB/config/wapiti/pattern_ref

3-4. doi extraction

File::buildReferences

Identifier.py

4. Data

4-1. All annotated data

KB/data/corpus1/XML_annotated2/

KB/data/corpus2/alldata_added_corrected/

4-2. 10-fold cross validation data, randomly selected

KB/data/corpus1/cross_validation/01/train/

KB/data/corpus1/cross_validation/01/test/

KB/data/corpus2/cross_validation/01/train/

KB/data/corpus2/cross_validation/01/test/

4-3. 10-fold cross validation data for corpus2, randomly selecting notes including <bibl> only

KB/data/corpus2/cross_validation/01/onlybibl/train/

KB/data/corpus2/cross_validation/01/onlybibl/test/

Part 2. <hi> tag treatment

Background

As original xml contains a lot of <hi> tags, which are often complicate and sometimes not correctly written, we should especially deal with the conflict among <hi> tag and annotated tags by Bilbo. The objective is exchanging tags so that <hi> tag includes text only. It is not easy because users sometimes do errors when they modify character appearance in their articles (<hi> concerns character appearance). Even if this is not very often, it's not easy to add some rules to detect and solve this problem because Bilbo also mistakes and it is not easy to automatically detect if the conflict comes from user or Bilbo. Anyway we start the correction from the obvious error of Bilbo and try to make some detailed rules for every error cases.

Operation

Input – an annotated reference (or note)

Output – reference (or note) with correctly arranged <hi> tags

In the input string, find <hi> tag and verify if it contains other tags annotated by Bilbo. If yes, move and (or) delete tags according to the following cases.

For a detected <hi> tag

Pre-treatment

If there is only one type of tag but the tags have not been joined because they are name tags -> group them, cause they are part of an author wrapped by <hi>

```
...<forename>L.</forename> <hi rendition="#T5"><surname>de</surname>
<surname>Laborde</surname></hi>...
```

->

```
... <forename>L.</forename> <hi rendition="#T5"><surname>de Laborde</surname></hi> ...
```

case 1

if there is only one tag and it is title, move it

```
... <title_m><hi rend="italic">Dynamic of Destruction</hi></title_m><hi rend="italic"> </hi><hi rend="italic">:
<title_m>Culture and Mass Killing in the First World War</title_m></hi> ...
```

After <title_m><hi rend="italic">Dynamic of Destruction</hi></title_m><hi rend="italic"> </hi><title_m><hi rend="italic">: Culture and Mass Killing in the First World War</hi></title_m>

case 2

if there is only one tag and no characters between pairs

```
... <forename><hi rend="st">V. J. </hi></forename><hi rend="st"><surname><hi
rendition="#T30">Harward</hi></surname></hi><hi rend="st">, </hi> ...
```

After <forename><hi rend="st">V. J. </hi></forename><surname><hi rend="st"><hi rendition="#T30">Harward</hi></hi></surname><hi rend="st">, </hi>

case 2-1

just one another token, accept that token as same tag

```
... <hi font-variant="small-caps">van <forename>Uytven</forename> </hi><surname>Raymond</surname><hi
font-variant="small-caps">, « </hi><title_a>Wenceslas I van Bohemen</title_a> », ...
```

After ... <forename><hi font-variant="small-caps">van Uytven </hi></forename><surname>Raymond</surname><hi font-variant="small-caps">, « </hi><title_a>Wenceslas I van Bohemen</title_a> » ...

case 3

if there are more than one tag but only one title, move title and delete the other tags

```
... <title_a><hi rendition="#T11">Le </hi><hi rendition="#T3">c</hi></title_a><hi  
rendition="#T11"><title_a>onfesseur du roi</title_a>. <nolabel>Les directeurs de conscience sous la monarchie  
française</nolabel></hi>, ...
```

After <title_a><hi rendition="#T11">Le </hi><hi rendition="#T3">c</hi></title_a><title_a><hi
rendition="#T11">onfesseur du roi. Les directeurs de conscience sous la monarchie française</hi></title_a>

case 4

if all tags are title tags, but different, take the first title tag for all

```
... <hi rendition="#T11"><title_m>Time Sanctified</title_m>. <title_a>The Book of Hours in Medieval Art and  
Life</title_a></hi>, ...
```

After <title_m><hi rendition="#T11">Time Sanctified. The Book of Hours in Medieval Art and
Life</hi></title_m>

case 4-1

mix of different titles and other tags, take the last title tag for all

```
... <hi rendition="#T2"><title_a>Fraternité universelle et intérêt national</title_a> (<date>1713-1795</date>).  
<title_m>Les cosmopolitiques du droit des gens</title_m></hi>, ...
```

After <title_m><hi rendition="#T2">Fraternité universelle et intérêt national (1713-1795). Les cosmopolitiques
du droit des gens</hi></title_m>

case 5

if all tags are name tags and there are other words, maybe annotation error, delete

```
... <hi rendition="#T16">Étude sur le style de la Continuation du Perceval par <forename>Gerbert</forename>  
et du Roman <surname>de</surname> la Violette par Gerbert de <surname>Montreuil</surname></hi>, ...
```

After <hi rendition="#T16">Étude sur le style de la Continuation du Perceval par Gerbert et du Roman de la
Violette par Gerbert de Montreuil</hi>

fully annotated name tags

case 5-1

if <hi font-variant="small-caps"> or <hi rend="bold"> : person name, CUT

```
<bibl><hi rend="bold"><surname>Caillois</surname> <forename>R.</forename></hi>, <title_m><hi  
rend="italic">Les jeux et les hommes</hi></title_m>, <place>Paris</place>,  
<title_s>Idées</title_s>/<publisher>Gallimard</publisher>, <date>1967</date>.</bibl>
```

After <bibl><hi rend="bold"><surname>Caillois</surname></hi> <hi
rend="bold"><forename>R.</forename></hi>, <title_m><hi rend="italic">Les jeux et les
hommes</hi></title_m>, <place>Paris</place>, <title_s>Idées</title_s>/<publisher>Gallimard</publisher>,
<date>1967</date>.</bibl>

case 5-2

if there is no title in the input reference (note), delete all tags and wrapping with <title_m>

```
<note n="24" place="foot"><p rend="footnote"><hi rendition="#T4"><forename>Décade</forename>  
<surname>égyptienne</surname></hi>, <date>1799</date>, <abbr>t</abbr>. <biblscope_v>II</biblscope_v>,  
<abbr>p.</abbr> <biblscope_pp>93</biblscope_pp>.</p></note>
```

After <note n="24" place="foot"><p rend="footnote"><title_m><hi rendition="#T4">Décade égyptienne</hi></title_m>, <date>1799</date>, <abbr>t</abbr>. <biblscope_v>II</biblscope_v>, <abbr>p.</abbr> <biblscope_pp>93</biblscope_pp>.</p></note>

case 5-3

there is title in the input reference (note), then check it can be really name

<note n="6" place="foot"><p rend="footnote"><hi rendition="#T32" xml:lang="en"><forename>Robert</forename> <surname>de</surname> <surname>Boron</surname></hi>, <hi rendition="#T47" xml:lang="en"><place>Joseph</place> d'<title_t>Arimathie: A Critical Edition of the Verse and Prose Versions</title_t></hi>, <abbr>éd</abbr>. <forename>R.</forename> <hi rendition="#T32" xml:lang="en"><surname>O'</surname> <surname>Gorman</surname></hi>, <place>Toronto</place>, <publisher>University of Toronto Press</publisher>, <date>1995</date>.</p></note>

After <note n="6" place="foot"><p rend="footnote"><forename><hi rendition="#T32" xml:lang="en">Robert</hi></forename> <surname><hi rendition="#T32" xml:lang="en">de</hi></surname> <surname><hi rendition="#T32" xml:lang="en">Boron</hi></surname>, <hi rendition="#T47" xml:lang="en"><place>Joseph</place> d'<title_t>Arimathie: A Critical Edition of the Verse and Prose Versions</title_t></hi>, <abbr>éd</abbr>. <forename>R.</forename> <hi rendition="#T32" xml:lang="en"><surname>O'</surname> <surname>Gorman</surname></hi>, <place>Toronto</place>, <publisher>University of Toronto Press</publisher>, <date>1995</date>.</p></note>

case 5-4

else, delete all tags and wrapping with <title_m>

<note n="24" place="foot"><p rend="footnote"><hi rendition="#T16"><forename>Two</forename> <surname>Old</surname> <forename>French</forename> <surname>Gauvain</surname> <forename>Romances</forename>: </hi><hi rendition="#T23">"</hi><title_a><hi rendition="#T16">Le chevalier à l'épée</hi></title_a> ...

After <note n="24" place="foot"><p rend="footnote"><title_m><hi rendition="#T16">Two Old French Gauvain Romances: </hi></title_m><hi rendition="#T23">"</hi><title_a><hi rendition="#T16">Le chevalier à l'épée</hi></title_a>

case 6

if all tags are removable (defined in the list 'canDelete'), delete ALL

<note n="28" place="foot"><p rend="footnote"><forename>I.</forename> <surname><hi rendition="#T30">Arseneau</hi></surname>, « <title_a>Ce roman</title_a>... », <hi rendition="#T23"><bookindicator>in</bookindicator> <nolabel>fine</nolabel></hi>.</p></note>

After <note n="28" place="foot"><p rend="footnote"><forename>I.</forename> <surname><hi rendition="#T30">Arseneau</hi></surname>, « <title_a>Ce roman</title_a>... », <hi rendition="#T23">in fine</hi>.</p></note>

case 7

fully mixed, but most of case is removable. consider some special sub cases.

case 7-1

if there is no title in the input, delete all tags and wrap with <title_m> **Bilbo error**

<note n="14" place="foot"><p rend="footnote"><forename>Cf.</forename> <surname>Jean-Yves</surname> <surname>Leloup</surname>, <hi rendition="#T4"><place>Désert</place>, <publisher>déserts</publisher></hi>, <publisher>Albin Michel</publisher>, <date>1996</date>. Dans <hi rendition="#T4">Naissance du désert</hi> (Balland, 1992), Chantal Dagron et Mohamed Kacimi se sont mis à la recherche des significations et des imaginaires du désert dans les différentes traditions méditerranéennes.</p></note>

After <note n="14" place="foot"><p rend="footnote"><forename>Cf.</forename> <surname>Jean-Yves</surname> <surname>Leloup</surname>, <title_m><hi rendition="#T4">Désert, déserts</hi></title_m>,</p></note>

<publisher>Albin Michel</publisher>, <date>1996</date>. Dans <hi rendition="#T4">Naissance du désert</hi> (Balland, 1992), Chantal Dagron et Mohamed Kacimi se sont mis à la recherche des significations et des imaginaires du désert dans les différentes traditions méditerranéennes. </p></note>

case 7-2

<place>, <publisher>, <abbr> <---- keep them

... <title_a><hi rend="subtitle1"><hi rendition="#T22"> siècle</hi></hi></title_a><hi rend="subtitle1">,
<place>Orléans</place>, <publisher>Paradigme</publisher>,
<abbr>coll</abbr>. </hi><title_a>Medievalia</title_a> », <date><hi rend="subtitle1">2000</hi></date></p></note>

After <title_a><hi rend="subtitle1"><hi rendition="#T22"> siècle</hi></hi></title_a><place><hi rend="subtitle1">, Orléans</hi></place>, <publisher><hi rend="subtitle1">Paradigme</hi></publisher>,
<abbr><hi rend="subtitle1">coll. </hi></abbr><title_a>Medievalia</title_a> », <date><hi rend="subtitle1">2000</hi></date></p></note>

case 7-3

other cases. for the moment, treat as sub case 1

... voir : <forename>S.</forename> <surname>B<hi rendition="#T6">oynton</hi></surname>, <hi rendition="#T1"><publisher>Shaping a Monastic Identity...</publisher>, <w>op. cit.</w>. </hi>;
<forename>G.</forename> <surname>B<hi rendition="#T6">aroffio</hi></surname>, « <title_a>San Benedetto Po-Polirone : una tradizione cluniacense in Italia</title_a> », <hi rendition="#T1"><forename>Vox</forename> <surname>Antiqua</surname></hi>,
<biblscope_i>1</biblscope_i> (<date>2010</date>), <abbr>p.</abbr> <biblscope_pp>121-172</biblscope_pp>. </p></note>

After ... voir : <forename>S.</forename> <surname>B<hi rendition="#T6">oynton</hi></surname>,
<title_m><hi rendition="#T1">Shaping a Monastic Identity..., op. cit. </hi></title_m>;
<forename>G.</forename> <surname>B<hi rendition="#T6">aroffio</hi></surname>, « <title_a>San Benedetto Po-Polirone : una tradizione cluniacense in Italia</title_a> », <title_m><hi rendition="#T1">Vox Antiqua</hi></title_m>, <biblscope_i>1</biblscope_i> (<date>2010</date>),
<abbr>p.</abbr> <biblscope_pp>121-172</biblscope_pp>. </p></note>

Part 3. Advanced Comprehension of Code for Modification of BILBO

1. Features

Background

The quality of CRF model greatly depends on features describing input token characteristics. Therefore a feature modification is important for both research and practical issues. Here we present how to modify features in the code.

Classes and methods to modify

If you want to add features, first modify one of the following files.

Rule.py

- Class that reorganizes tokens and extracts features according to predefined rules.
- It is concerned with early data processing when saving data in Reference objects.
- In CleanCorpus1 or CleanCorpus2, we already created Reference objects by separating tokens with whitespaces. A Reference object is a list of Word objects, which contains these separated tokens with annotated tags and attributes. The original features are simply the manually annotated tag attributes that had been extracted in CleanCorpus1 or CleanCorpus2.
- Features extracted in Rule.py are processed according to the following order:
 - o Lexicons defined in KB/config/lexique.txt – in *self._checkLexique*
 - o Check initial expression – in *self._checkLexique*
 - o Check url expression – in *self._initCheck* in *self._checkLexique*
 - o Check numbers, dash and capital chars – in *self._refCheck* in *self._checkLexique*
 - o Check guillemots and quotes – in *self._featureCheck*
- Now if you want to detect your own new features, either add detection in *self._featureCheck* or create a new method such as *self._featureCheck* then call it after the call of *self._featureCheck* in the method reorganizing.
- To understand class structures, see “doc/conception/class_diagram_bilbo.png” and “doc/documentation/[class_name].html”.

lexique.txt

- As above explication, some lexicons are defined in KB/config/lexique.txt. If you want add other lexicon features, just modify this file.
- There are two types of lexicons features as explained README_EN.txt: [including] and [matching].
- [including] means check if the corresponding word is INCLUDED in the input token: e.g. *[January]*,
- [matching] means check if the corresponding word is exactly matching the input token : e.g. *pp.*

Once you add features by modifying the above files, you need to modify the following files.

features.txt

- It contains information about valid features (# features) and labels to be ignored (#nonLabels).
- Even if features are defined in the above files (Rule.py, lexique.txt), we can again filter them using this features.txt. It's to make experiments easy. You first extract all features via Rule.py or lexique.txt, and then choose necessary features via the features.txt file.
- So, we accept features only written in KB/config/features.txt under the heading # features

Extract.py

- Class that reorganizes again features and labels for final training and test data for CRF. It also contains data printing methods.
- You need to set correctly the data printing for Wapiti because its data format is particular.
- Modify *self._printdataWapiti* method.
- Wapiti input data features should be grouped according to feature types. For example, since the features ALLCAP, 'FIRSTCAP', ALLSMALL, and 'NONIMPCAP' are related to capital expression of characters, and these features are exclusive each other, we group them.
- We set this grouping information in the list, 'features' as follows. So modify the list features.

- **features =**

```

[['ALLNUMBERS', 'NUMBERS'], #1
['DASH'], #2
['ALLCAP', 'ALLSMALL', 'FIRSTCAP', 'NONIMPCAP'], #3
['BIBL_START', 'BIBL_IN', 'BIBL_END'], #4
['INITIAL'], #5
['WEBLINK'], #6
['ITALIC'], #7
['POSSESSOR'], #8
['POSSPAGE'], #9
['POSSMONTH'], #10
['SURNAMELIST'], #11
['FORENAMELIST'], #12
['PLACELIST'], #13
['JOURNALLIST']] #14
#There are 14 groups and 20 features

```
- You should modify also the list **tmp_features** indicating default value of features when the corresponding features do not exist. As the number of groups is 14, the length of this list should be 14 also.
- **tmp_features** = ['NONUMBERS', 'NODASH', 'NONIMPCAP', 'NULL', 'NOINITIAL', 'NOWEBLINK', 'NOITALIC', 'NOEDITOR', 'NOPAGE', 'NOMONTH', 'NOSURLIST', 'NOFORELIST', 'NOPLACELIST', 'NOJOURLIST']

pattern_ref

- Finally we modify the file KB/config/wapiti/pattern_ref. It's Wapiti pattern setting file.
- Detailed information of this file is in KB/config/wapiti/explication_pattern_ref.
- You just need to **DO (delete #) or UNDO (write #) lines according to the number of groups you defined above**. As you have 14 groups, you UNDO the lines related to 15th and more groups like the following example.

```

#U149:%x[-1,15]
#U150:%x[0,15]
#U151:%x[1,15]

```
- If you added 15th group, delete # at starting of line to accept these features.