Enrichment of Renaissance texts with proper names

Denis Maurel, Nathalie Friburger, Iris Eshkol-Taravella

Enrichment of Renaissance texts with proper names

Denis Maurel[1], Nathalie Friburger[1], Iris Eshkol-Taravella[2]

[1]*Université François-Rabelais de Tours, Laboratoire d'informatique, EA 6300*

[2]*Université d'Orléans, Laboratoire ligérien de linguistique, UMR 7270*

denis.maurel@univ-tours.fr, *nathalie.friburger@univ-tours.fr*, *iris.eshkol@univ-orleans.fr*

Abstract

The Renom project proposes to enrich Renaissance texts by proper names. These texts present two new challenges: great diversity due to various spellings of words; numerous XML-TEI tags to save the exact format of original edition. The task consisted to add Named Entity tags to this format tagging with generally the left context and sometimes the right context of a name. To do that, we improved the free and open source program CasSys to parse texts with Unitex graph cascades and we built dictionaries and specific cascades. The slot error rate was 6.1%.

Proper Names and maps. were to allow navigating into. So, this paper deals with Named Entity Recognition in Renaissance texts.

Keywords

Named entities; Renaissance texts; graph cascades; CasSys, Humanities and tourism.

1 Motivation

From more than ten years ago, the Center of Higher Education of the Renaissance (CESR) proposes on the Web the Humanist Electronic Libraries (BVH)[1]: a great number of Renaissance books, from Rabelais, Ronsard and so on as scanned and transcribed books. The transcription, defined in the TEI format, follows the same presentation than the scan one: paragraph, line breaks, abbreviations, hyphen, lettering, and so on. Figure 1 presents extracts of the Website: one paragraph (transcription and scanned text) of the novel *Gargantua* from Rabelais, transcribed in TEI-format bellow (--- denotes paragraph and <lb/> denotes line break).

```
<p>
<lb/><hi rend="larger">E</hi>N ceste mesmes saison Fayoles
<lb/>quart roy de Numidie envoya
<lb/>du pays de Africque a Grand-
 <lb rend="hyphen"/>gousier une jument la plus enorme &amp;amp; la
<lb/>plus grande que feut oncques veue, &amp;amp;
<lb/>la plus monstreuse, Comme assez scavez,
<lb/>que Afrique aporte tousjours quelque
<lb/>chose de noveau.
</p>
```

[1] http://www.bvh.univ-tours.fr/

Rabelais spoke about Tours Region (in France) where the giants Grandgousier, Gargantua and Pantagruel move into. So we planned with the *Renom* project[2] to develop 'literary tourism' with links between names and books: the website proposes to navigate in the novel using the proper names and to see where the imaginary or antique places were `located`… Figure 2 shows the imaginary Theleme Abbacy between the towns Chinon and Azay-le-Rideau, near Tours. Tourists are incited to visit the castles of these two towns and the Rabelais Museum near these `places`…

EN ceste mesmes saison Fayoles
quart roy de Numidie envoya
du pays de Africque a Grand-
gousier une jument la plus enorme & la
plus grande que feut oncques veue, &
la plus monstreuse, Comme assez scavez,
que Afrique aporte tousjours quelque
chose de noveau.

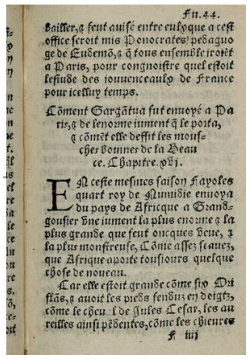

Figure 1: extract from the BVH website

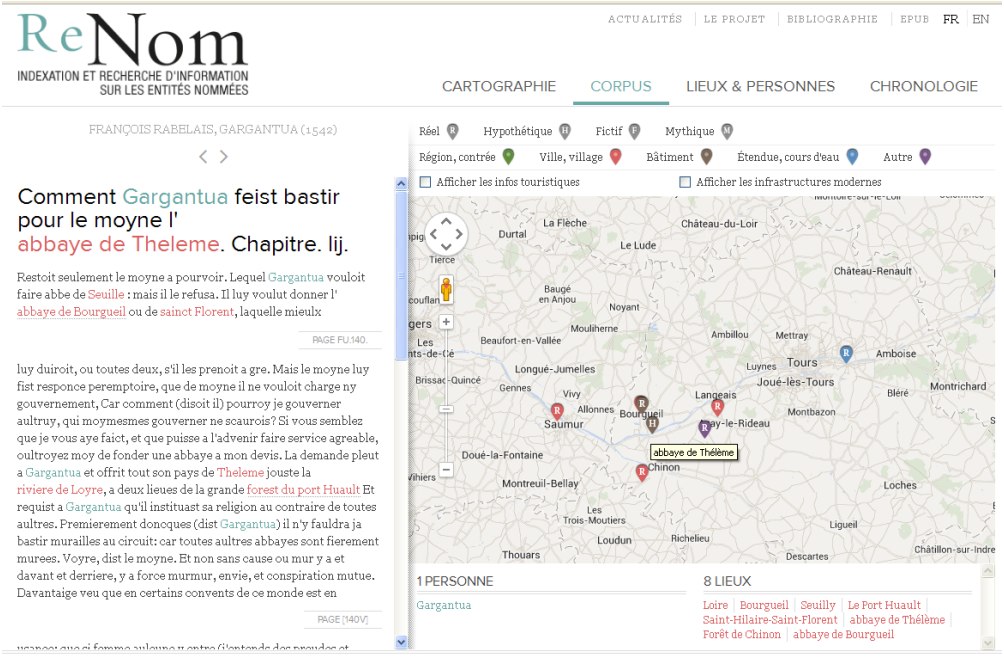

Figure 2: extract from the Renom website

The aim of the scientific project was to enrich the texts with tags indicating names of persons, personages and locations with supervised Natural Language Processing (NLP) techniques. This task is well defined in NLP as Named Entity Recognition (NER) from two decades. The challenge is the TEI format with a lot of hyphens and the great variability in the spelling of proper names (and in the orthography of words on the whole). For instance, in the extract of Figure 1, the name *Grandgousier* is written *Grand-<lb rend="hyphen"/>gousier* and the name *Africa* has two orthographies, *Africque* and *Afrique*.

From MUC conferences (*Message Understanding Conferences*), the NER includes person names, location names, organization names, dates, percentages, currency (Chinchor, 1997)

[2] http://renom.univ-tours.fr/

and sometimes titles, hours, occupations, etc. A state of the art of NER can be found in (Nadeau, Sekine, 2009); the main idea is to use *internal and external evidence* (MacDonald, 1996), i.e. the local context. For instance, in the sequence *Hugues Thierry Salel l'ainé, seigneur de Seuille*, the first name *Hugues Thierry* indicates that *Hugues Thierry Salel* is a person name (*internal evidence*) and the title *seigneur* (*lord*) proofs that Seuille is a toponym (*external evidence*).

Three approaches are possible, machine learning or symbolic rules and sometimes hybrid approaches. Machine learning techniques need training corpus, not available for this task. For this reason we used an approach based on symbolic rules (Ait-Mokhtar et Chanod, 1997; Hobbs *et al.*, 1997). To facilitate the cooperation between computer scientists, linguists and Renaissance experts, we chose Unitex platform[3] (Paumier, 2003) for its friendly interface (with use of graphs) and its free license. With Unitex, we can define cascade rule systems (CasSys menu), with all properties of Unitex graphs. Cascades (Abney, 1991) are used in many NLP applications, as chunking (Abney, 1996), syntactic analysis (Kokkinakis, Kokkinakis, 1999), morphological analysis (Alegria et al., 2001) and so on. Our system is inspired from (Friburger, Maurel, 2004).

The NLP community is interested in ancient languages and ancient states of modern language. See for instance (Denooz, Rosmordus, 2009). In old French, orthography is not fixed and a lot of name variants exist. For middle French (just before Renaissance French), specific tools and dictionaries (Souvay, 2004) or lemmatizers (Souvay, 2007) has been developed. In the same way, we used specific dictionaries and cascades, built cooperatively with Renaissance experts. The goal of Renom project is to help experts to tag precisely texts and to complete dictionaries. These dictionaries contain proper names and their variants linked with unique keys (proposed by Renom and validated by expert) and locations linked with unique keys to the Geonames[4] database. This pointer is used by the website to post the map. When too tiny locations or ancient locations were not found in Geonames, experts added new entries to Geonames.

Corpus presentation

The corpus contains 11 books:

–*Discours fantastiques* (edition of 1566), Justin Tonnelier;

–*Courtisan* (edition of 1538), Baldassare de Castiglione;

–*Voyage de Tours* (edition of 1560) and *Élégie sur les troubles d'Amboise* (edition of 1563), Pierre de Ronsard;

–*Gargantua* (edition of 1542), *Pantagruel* (edition of 1542), le *Tiers Livre* (editions of 1546 and 1552), le *Quart Livre* (editions of 1548 and 1552) and *Brève déclaration* (edition of 1552), François Rabelais.

The format is very particular: as we explained at section 1, it respects the whole layout of the original edition, line feed, footer, initial letter and so on. Sometimes, the transcriber added some corrections. For instance:

- Initial letter (*Pantagruel*)

<lb/><hi rend="larger">P</hi>Antagruel quelque jour pour se

[3] http://www-igm.univ-mlv.fr/~unitex/

[4] http://www.geonames.org/

- Transfer of the end of the first line at the end of the second one, after parenthesis, but transcribed on three lines[5]. Latin citation.

<item>Les hoseaulx, alias les bottes de patien
<lb rend="hyphen"/><hi rend="bottom">(ce.</hi></item>
<item><foreign xml:lang="lat">Formicarium artium</foreign>.</item>

- Footer (Dace truncated by the page number - 188)

<lb rend="hyphen"/>bek Norwerge, Sweden, Rich, Da-
<lb/>
<lb/>
<lb/>
<pb n="188" xml:id="_Page_-188"/>
<lb/>
<lb/><fw place="top-left" type="pageNum">[94v]</fw>
<lb rend="hyphen"/>ce, Gotthie, Engroneland, les Estre-

- Correction (addition of apostrophe)

<item>Les aultres a sainct Jean <choice><orig>dangery</orig><reg>d'angery</reg></choice>.</item>

The named entity brackets have to contain all the format brackets. For instance the last example becomes:

<item>Les aultres a ***<placeName>****sainct Jean*
<choice><orig>dangery</orig><reg>d'angery</reg></choice> ***</placeName>****.</item>*

2 Typology used

The CESR used TEI format for transcribed texts, so it was obvious to adopt also the TEI typology. We used four types: geography (*geogName*), places (*placeName*), organizations (*orgName*) and persons (*persName*).

2.1 Geography and places

Geographic names were divided in two subtypes: first, geonyms (mountains, plains, plateaus, grottos…) and, second, hydronyms (oceans, seas, rivers, lakes, ponds…). When exist, the geographical precisions were included in tags, with specific internal tags.

<geogName type="geo" key="#loc_montsinai"><geogFeat>mont</geogFeat> Sinaï</geogName>

<geogName type="hydro" key="#loc_loire"><geogFeat>rivière</geogFeat> de Loyre</geogName>

Place names were sometimes also subtyped (cities, countries, estates and buildings).

<placeName type="city" key="#loc_seuilly">Seuille</placeName>

<placeName type="country" key="#loc_france">France</placeName>

<placeName type="building" key="#loc_lapommardiere">mestayrie de la Pomardiere</placeName>

Two locations were sometimes imbricated.

<placeName type="building">Palais de <placeName type="city" key="#loc_poitiers">Poictiers</placeName></placeName>

<placeName key="#loc_guevede">gue de <geogName type="hydro" key="#loc_vede">Vede</geogName></placeName>

<geogName key="#loc_ilescanaries">isles de <placeName key="#loc_canaries">Canarre</placeName></geogName>

[5] The edited text is: *Les hoseaulx, alias les bottes de patien*
Formicarium artium. (ce.

2.2 Organizations

Organizations were divided in thee subtypes: peoples, estates and communities. The CESR choose to not link organizations with keys.

<orgName type="domaine">Royaulme de <placeName type="pays" key="#loc_france">France</placeName></orgName>

Organizations were sometimes imbricated.

<orgName type ="domaine">Royaulme des <orgName type = "peuple">Dipsodes</orgName></orgName>

When it was difficult to choose between placeName and orgName, we had inserted the two tags.

<placeName type="building" key="#loch_coingnaufondabbaye"><orgName type="community">*abbaye de <placeName type="city" key="#loch_coingnaufond">Coingnaufond</placeName>****</orgName></placeName>***

2.3 Persons

The simplest examples were just persName tags (with their keys).

<persName key="#pers_aristote">Aristote</persName>

If exist, we added internal tags with first names (*foreName*), surnames (*surName*) and particles (*nameLink*).

<persName key="#pers_francoisconnan"><forename>François</forename> <nameLink>de</nameLink> <surname>Connan</surname></persName>

Finally, these tags were extended with titles or civilities (*roleName*) that are subtyped: nobiliary role, religious role, function or occupation, honor. When the title included a place name, it was also tagged: the *lord of Essars* is a person, but *Essars* is a place:

*le <persName key="#pers_seigneurdesessars">****<roleName type="nobiliary">****seigneur****</roleName><placeName key="#loc_desessars">****des Essars****</placeName>****</persName>, & quelques*

We added sometimes precisions: nicknames or role in the family (elder son below).

<persName key="#pers_huguesthierrysalel"><forename>Hugues</forename> <forename>Thierry</forename> <surname>Salel</surname> ***<genName>****l'ainé****</genName>****, <roleName type="nobiliary">seigneur de <placeName type="ville" key="#loc_seuilly">Seuille</placeName></roleName></persName>*

The text contained ambiguities. If possible, the expert will choose the good interpretation. For instance below, *saint Martin de Candes* may be a church or a person:

<persName key="#pers_sainctmartindecandessaintmartin"><placeName key="#loc_sainctmartindecandessaintmartin">*sainct <lb/>Martin de <placeName type="city" key="#loc_candessaintmartin">Candes</placeName>****</placeName></persName>***

3 Dictionaries

As we said below, we often need for NER to recognize the context of named entities. So we built a variant orthography dictionary, studying contexts and using old first name list.

For instance, in the Renaissance, the word captain was written *capitaine*, *capiteine* or *cappitaine*. We chose the synchronic entry as lemma and we added features to use it for the NER (see sections 5.3 and 5.4):

capitaine,.N+Military:ms
capiteine,capitaine.N+Military:ms
cappitaine,capitaine.N+Military:ms

The second line contains five informations: form (*capiteine*), lemma (*capitaine*), part of speech (*N*), feature (*Military*) and morphology (*ms*).

We transformed three CESR lists of names: persons, organizations, locations in Unitex dictionaries, which were improved after each book parsing.

In the three name dictionaries, a word has its key for lemma. This key is used to link different orthographies and to link also locations to Geonames:

ancenis,loc_ancenis.N+id=loc:ms
ancenys,loc_ancenis.N+id=loc:ms

In previous works, CESR experts choose to use explicit keys (as *loc_ancenis* for the toponym *Ancenis*). We had to use the same system.

Table 1 presents the number of dictionary entries at the end of Renom project

Persons	1 145
Locations	987
Organizations	57
Other words	2 622

Table 1: Number of dictionary entries

4 Improvements of Unitex platform

As we said below, we choose Unitex platform to facilitate the cooperation between computer scientists, linguists and Renaissance experts. Unitex is open-source and free (LGPL license). With Unitex, one can parse texts with his own dictionaries (see section 3) and write linguistic rules as graph with a very friendly interface; it is also possible to build cascades of graph with the CasSys menu.

A graph cascade is a succession of graph parsing: the first graph parses the text, the second graph parses the text modified by the first graph and so on.

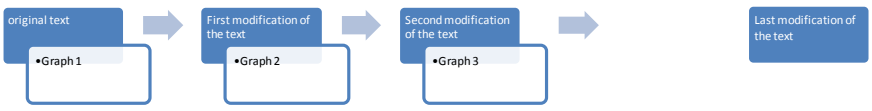

Figure 3: Graph cascade principle

The Renom project had need three improvements of CasSys (see below): graph iteration until fixed point, use of Unitex morphological dictionaries and a more convivial file for the output of the cascade.

In Unitex system, graphs parse text possibly merging new sequences, replacing others, using variables, moving sequences, inserting information from dictionaries.

4.1 Graph iteration

We added to CasSys graph iteration until fixed point: the iterative graph parses the text, then it parses the text result and so on until the parsing does not modify the resulting text (this is the fixed point).

We used iterative graphs above all for building keys of imbricated types. See section 5.5.

4.2 Unitex morphological dictionaries

A Unitex graph can extract information from dictionaries: lemmas, feature or morphology. These dictionaries are named `morphological dictionaries`.

We added the possibility to include in a cascade this kind of graph. We defined our three name dictionaries (section 3) as `morphological dictionaries` and, when a name is in one of these dictionaries, we linked it to its key. See again section 5.5.

4.3 File output of a cascade

The major idea to parse text with graph cascade is to consider a tagged text as a *multiword expression* (*MWE*) because the other graphs of the cascade cannot parse inside. A sequence of characters is recognized by the Unitex system as a MWE if it is enclosed with curly brackets.

For instance, the XML tag *<lb rend="hyphen"/>* will be interpreted as a MWE if curly brackets are added before and after the tag:

{<lb rend="hyphen"/>,.BaliseXML+DFlb+hyphen}

That is done by the graph of Figure 4.

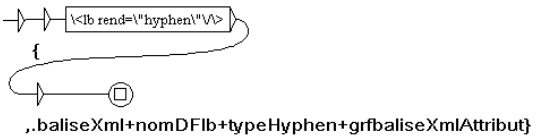

Figure 4 : A graph recognizing XML tag *lb*[6]

So other graphs of the cascade can parse this MWE with Unitex expressions as *<BaliseXML>*, *<lb>* or *<hyphen>*, depending on the necessary degree of precision.

A disadvantage is that the resulting text is difficult to be read. To overcome this we implemented a specific CasSys XML format:

<csc><form><lb rend="hyphen"/></form>
<code>BaliseXML</code><code>DFlb</code><code>hyphen</code></csc>

So, our cascades always are in couple: the first one parses the text and the second one transforms the specific CasSys XML format in the required format.

The first graph of the cascade transforms all the XML tags in MWEs.

5 Method

There are two groups of texts. The first group of texts contained tags for proper name (persName, geogName and placeName), added manually by experts; our work on these texts was to add internal tags (geoFeat, foreName, lastName…), to search the key in the dictionaries and to extend names to named entities with roleName (Lord, Abbey…), genName (elder son…) and so on. The second group of texts was annotated only for formatting. So we have to recognize names in these texts, before doing the same work as the first group of texts.

So we organized our work in four steps (see Figure 5). The few pre-tagged texts were parsed only from third step.

1. Preprocessing to rebuild truncated names at the end of line or page;
2. Dictionaries lookup and use of context rules to tag names;

[6] As the other, this graph add also a feature with it name, to debug the cascade.

3. Consultation of dictionaries and application of internal and expanded rules, as presented just before (*firstName* versus *genName* and so on).
4. Extraction of names that are not in dictionaries.

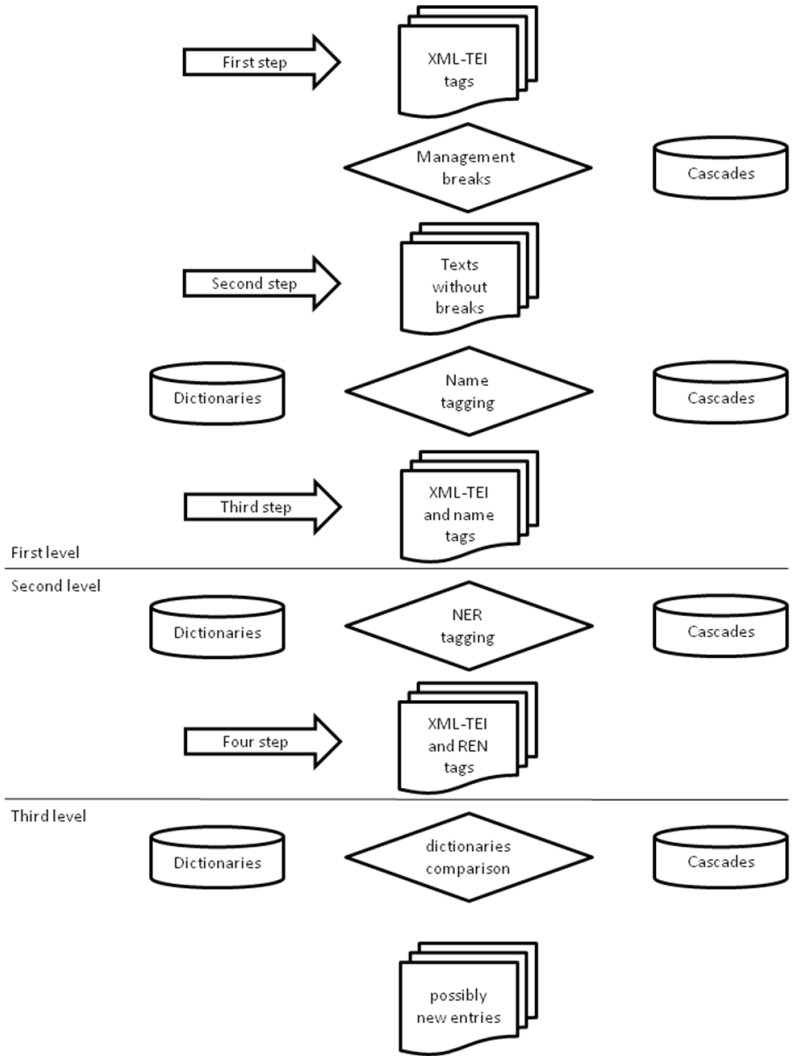

Figure 5: Four steps

5.1 Hyphens

As we said, there are a lot of hyphens in the text. A first graph recognizes XML tags as multiword expression (MWE) with the part of speech (POS) *baliseXML*. The other graphs of the first cascade rebuild words. For instance, the graph of Figure 6 recognizes a letter larger than other: the *hi* tag cut the word and the graph builds entire word. The two *hi* tags becomes a new MWE with POS *largerSup* and a new attribute memorizes the size of the hyphen (here, *value="1"*).

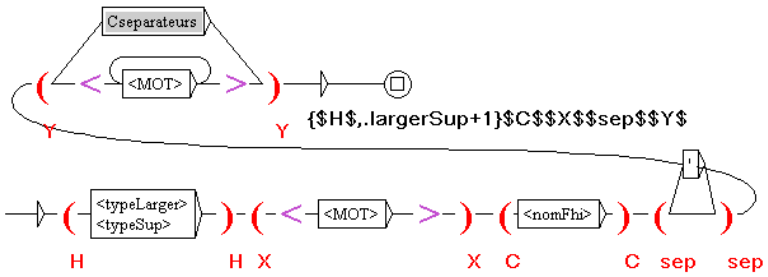

Figure 6 : Graph to rebuild words with the first letter larger.

For instance:

<hi rend="larger">E</hi>N ceste mesme heure

becomes

***{**<hi rend="larger" **value="1"**></hi>**,.largerSup}**EN ceste mesme heure*

5.2 Abbreviations

The graph of Figure 7 recognizes the XML tags *choice*, *abbr* and *expan*, and builds a MWE[7] with POS *abbreviation*.

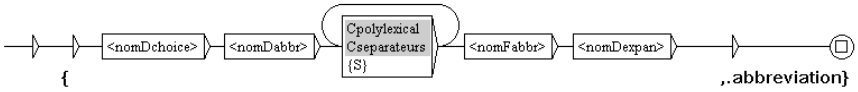

Figure 7 : Graph to hide abbreviations

For instance:

<choice><abbr>PAN.</abbr><expan>PANURGE</expan></choice>

becomes

***{**<choice><abbr>PAN.</abbr><expan>**,.abbreviation}**PANURGE</expan></choice>*

5.3 Names recognized from dictionaries

The second step begins with a dictionary lookup to tag names that are in dictionaries. Some names are ambiguous, so we use the context to disambiguate person from location or organization. The graph of Figure 8 tags persName in military context.

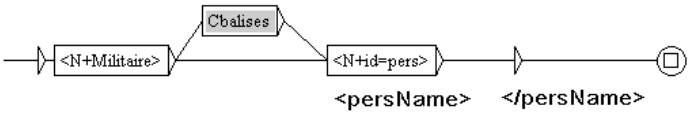

Figure 8 : Graph to tag persname from dictionaries in military context

For instance:

<lb/> du capitaine Engoulevent, pour descou

becomes

*<lb/> du capitaine **<persName>**Engoulevent**</persName>**, pour descou*

[7] This MWE hides the original word *PAN* to the parsing. Here, this name is ambiguous to the Greek mythological god Pan.

5.4 Names recognized only from context

When we recognized names from dictionaries, we used the same contexts to tag names that are not in the dictionaries, if the first letter is capitalized. The graph of Figure 8 tags *persName* in military context.

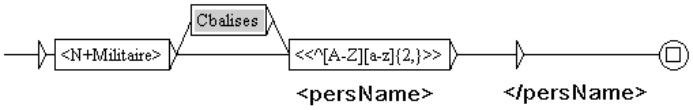

Figure 9 : Graph to tag persname with first letter capitalized in military context

For instance:

<lb/> du chevalereux capitaine Moses

becomes

<lb/> du chevalereux capitaine ***<persName>Moses</persName>***

5.5 Keys

When names are identified, we search key in dictionaries, if the entry exists. The graph of Figure 10 returns these keys.

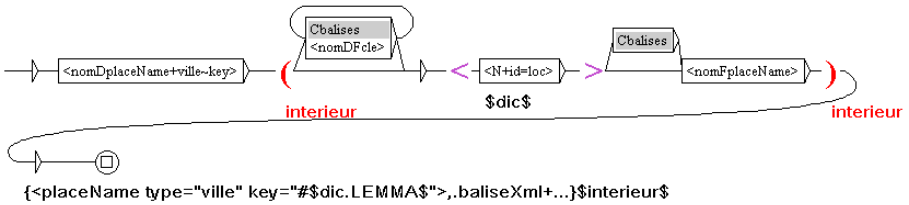

Figure 10 : Graph that search key in dictionaries.

For instance:

mestaiers de <placeName>Seuille</placeName> & de <placeName>Synays</placeName>.

becomes

mestaiers de <placeName ***key="#loc_seuilly"****>Seuille</placeName> & de <placeName* ***key="#loc_cinais"****>Synays</placeName>.*

If the name is not found in dictionaries, other graphs build a possible key by concatenation. The attribute *dic="no"* indicates to the expert that the name (with this orthography) was not found in the dictionaries. The expert adds it, with another key if it is a variant of existing entry or with these key if it is a real new entry.

To build key is not trivial, because of imbricated names. For instance we have to add three keys for the name *château du gué de Vede*: one for the proper noun Vede (key extracted from dictionaries), one for the ford of Vede and one for the castle of the ford of Vede:

<placeName ***key="#loc_chasteauduguedevede" dic="no"****>chasteau du <placeName* ***key="#loc_guedevede" dic="no"****>Gue de <geogName* ***key="#loc_vede"****>Vede</geogName></placeName></placeName>*

The graph that builds keys of imbricated names is iterative. It calls four subgraphs, one per given types. Figure 11 presents this graph and the subgraph that inserts key in a placeName.

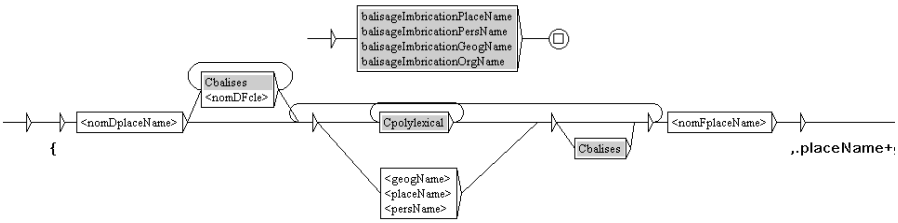

Figure 11 : Iterative graphs for imbricated names

5.6 Tags inside names

We introduced inside of *persName* tags forenames and surnames; and inside of *geogName* tags *geogFeat* tags. We used a Renaissance forename dictionary. Some difficulties: person with more than one forename (*hugues thierry salel*) or with a multiword surname (*Jan Trivolse Guallo*) or with a particle (*Ulrich Thierry du Gallet*). The graph of Figure 12 tags one forename and one surname (tags are in subgraphs). Then the names and their inside tags are considered as MWE.

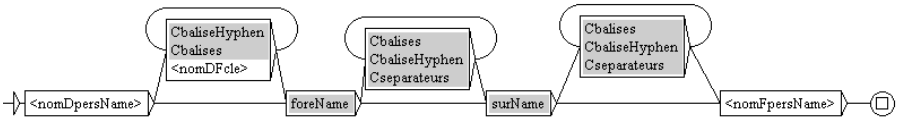

Figure 12 : One of graphs to add *forename/surname* tags

5.7 Extended tags

We also extended persNames to named entities with roleName (Lord, Abbey, teacher…), genName (elder son…) or addName (nicknames). We added new tags on the left or on the right and we moved PersName tags. The graph of Figure 13 taggs the left of a named entity when it is preceeded by a roleName. The key does not change.

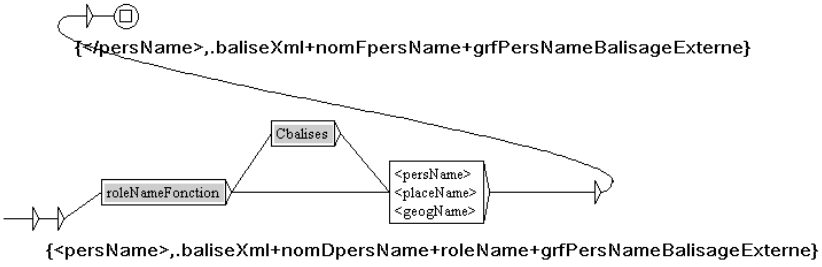

Figure 13 : Graph to tag a roleName at the left of a name

For instance, Epistemon is a personal teacher (*précepteur*):

<lb>ton precepteur <persName key="#pers_epistemon" dic="no">Epistemon<persName> don't

becomes

<lb>ton <persName key="#pers_epistemon" dic="no"><roleName type="function">precepteur<roleName> Epistemon<persName>

5.8 Dictionary completion

Finally the two last cascades build a new file for dictionary completion: they erase the text, except the names that are not in dictionary or are in dictionary with another feature.

For instance, the last example:

<lb>ton precepteur <persName key="#pers_epistemon" dic="no">Epistemon<persName> don't

becomes

Epistemon *#pers_epistemon*

The entire list of names with the feature dic="no" was transmitted to the experts to improve the dictionaries.

6 Evaluation

To evaluate our work, we parsed the two books of Pierre de Ronsard from our corpus (*Voyage de Tours* and *Élégie sur les troubles d'Amboise*). We constructed our cascades studying the other books of the corpus, principally the François Rabelais' books. We computed a weighted variant of the *slot error rate* (SER) (Makhoul et al., 1999) used in French evaluation campaign. SER distinguished between three types of errors:

1. Insertion (I - weight 1): we tagged words that are not names.
2. Deletion (D - weight 1): we failed to tag a name.
3. Tags with border errors: bad type (T - weight 0.5), tag outside or inside the proper name (E - weight 0.5) or both (TE - weight 1).

If #R is the sum of the entities of the reference texts, the SER is computed by:

$$SER = \frac{\#I + \#D + 0{,}5 * \#T + 0{,}5 * \#E + \#TE}{\#R}$$

With these counts, if #S is the sum of the detected entities, we can also compute precision and recall of our work:

$Precision = \frac{\#S-\#I}{\#S}$ and $Recall = \frac{\#S-\#I}{\#R}$

The tagged texts are entirely supervised, so border errors are less important. But we can also compute the type precision (correct recognition of types) and the limit precision (the boundaries of a named entity):

$Type\ precision = \frac{\#S-\#I-\#T-\#TE}{\#S}$ and $Limits\ precision = \frac{\#S-\#I-\#E-\#TE}{\#S}$

Table 2 presents the results.

The experts of CESR want to read the whole corpus before publication on the website. The SER of 6.1% is a real improvement of their work. The significant number of deletion corresponds to a lot of names without context and out of dictionaries. The experts completed them.

Conclusion

We presented in this paper NER in XML-TEI encoded Renaissance texts. The format of the corpus and the important variation of vocabulary need specific treatments from contemporary texts. We used dictionaries and rule-based cascades and we obtained 6.1% of SER. Our system will be settled in their production line of transcribed texts.

The most important texts of Rabelais are on line in the website Renom with a search engine for names. It uses key to link variants and to map locations.

Acknowledgment

This work is supported by Région Centre research program. Authors thank the CESR, particularly Sandrine Breuil, Marie-Luce Demonet, Jorge Fins and Marie Olivron.

#I	#D	#T	#E	#TE	#S	#R
5	19	3	3	0	136	150

SER	6,1%
Precision	96,3%
Recall	87,3%
Type precision	94,1%
Limits precision	94,1%

Table 2: Evaluation

References

Abney S. Parsing By Chunks. In *Principle-Based Parsing*, pp. 257-278, Kluwer Academic Publishers. 1991.

Abney S. Partial Parsing via Finite-State Cascades. *Workshop on Robust Parsing*, 8th *European Summer School in Logic, Language and Information*, Prague, Czech Republic, 8-15. 1996.

Ait-Mokhtar S., Chanod J.-P. « Incremental Finite-State Parsing », *Applied Natural Language Processing*, p. 72-79. 1997.

Alegria I., Aranzabe M. J., Ezeiza N., Ezeiza A., Urizar R. Using Finite State Technology in Natural Language Processing of Basque, CIAA 2001.

Chinchor N. Muc-7 Named Entity Task Definition. 1997.

Denooz J., Rosmorduc S. *Langues Anciennes*, TAL 50:2. 2009.

Friburger N., Maurel D. Finite-state transducer cascade to extract named entities in texts, *Theoretical Computer Science*, 313:94-104. 2004.

Hobbs J., Appelt D., Bear J., Israel D., Kameyama M., Stickel M., Tyson M. « FASTUS: A cascaded finite-state transducer for extracting information from natural-language text »,*Finite-State Language Processing*, MIT Press, p. 383-406. 1997.

Kokkinakis D., Kokkinakis S. J. A Cascaded Finite-State Parser for Syntactic Analysis of Swedish, *EACL'99*. 1999.

MacDonald D. (1996), Internal and external evidence in the identification and semantic categorisation of Proper Names, *Corpus Processing for Lexical Acquisition*, 21-39, Massachussetts Institute of Technology.

Makhoul J., Kubala J., Schwartz R., Weischedel R. Performance measures for information extraction, in Proceedings of *DARPA Broadcast News Workshop*. 1999.

Nadeau N., Sekine S. *A survey of named entity recognition and classification*, Satoshi Sekine and Elisabete Ranchhod, ed., John Benjamins publishing company, 3-28. 2009.

Paumier S. (2003), *De la Reconnaissance de Formes Linguistiques à l'Analyse Syntaxique*, Thèse de Doctorat en Informatique, Université de Marne-la-Vallée.

Souvay G. Vers un Dictionnaire électronique du Moyen Français, Actes du Colloque Euralex 2004, European Association for Lexicography congress, vol. 2 p.671-678. Lorient, France, 6-10 juillet. 2004.

Souvay G. LGeRM : un outil d'aide à lemmatisation du français médiéval, 18th International Conference on Historical Linguistics ICHL 2007, Université du Québec À Montréal. Canada. 6-11 août. 2007.