LITL at CLEF eHealth2016: recognizing entities in French biomedical documents

Lydia-Mai Ho-Dac, Ludovic Tanguy, Céline Grauby, Aurore Heu Mby, Justine Malosse, Laura Rivière, Amélie Veltz-Mauclair, Marine Wauquier

▶ **To cite this version:**

Lydia-Mai Ho-Dac, Ludovic Tanguy, Céline Grauby, Aurore Heu Mby, Justine Malosse, et al.. LITL at CLEF eHealth2016: recognizing entities in French biomedical documents. CLEF eHealth 2016, Sep 2016, Évora, Portugal. hal-01365928

HAL Id: hal-01365928
https://hal.science/hal-01365928

Submitted on 13 Sep 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

LITL at CLEF eHealth2016: recognizing entities in French biomedical documents

Lydia-Mai Ho-Dac (1), Ludovic Tanguy (1), Céline Grauby (2), Aurore Heu Mby (2), Justine Malosse (2), Laura Rivière (2), Amélie Veltz-Mauclair (2), and Marine Wauquier (2)

1: CLLE-ERSS: CNRS & University of Toulouse, France
2: Master LITL, University of Toulouse, France
{hodac,tanguy}@univ-tlse2.fr

Abstract. This paper describes the participation of master's students (LITL programme, university of Toulouse) and their teachers to the CLEF eHealth 2016 campaign. Two runs were submitted for task 2 (multilingual information extraction) which consisted in the recognition and categorization of medical entities in French biomedical documents. The system used consists of a CRF classifier based on a number of different features (POS tagging, generic word lists and syntactic parsing). In addition, several patterns were used on the CRF's output in order to extract more complex entities. The best run achieved high precision (0.64–0.78) but lower recall (0.32–0.40), with an overall F1-measure of 0.43–0.53.

1 Introduction

This article describes the participation of the students of the LITL master to the CLEF eHealth 2016 Lab [5].

LITL (stands for Linguistique, Informatique, Technologies du Langage, i.e. *Linguistics, IT, Language technologies*) is a new master's program at the University of Toulouse, France. Mainly aimed at linguistics and humanities students, it comprises, for a major part, courses in natural language processing (NLP), computational linguistics and practical aspects of corpus analysis through programming and using various computer tools. An important part of this curriculum is project-oriented, and the first year students have to build a fully operational processing system for a precise NLP task. This year's project was the participation to the CLEF eHealth challenge, more precisely task 2: multilingual Information Extraction [10].

The teachers in charge of this project (members of the CLLE-ERSS laboratory) deemed that this task was ideal for pedagogical purposes:

- information extraction (and more precisely named entity recognition – NER) is a well-known, well-defined and central task in modern NLP;
- state of the art information extraction systems are based on machine learning techniques but can also make use of symbolic handcrafted rule-based approaches;

– supervised learning systems such as CRF classifiers can take advantage of different kind of linguistic resources, many of these are available for the biomedical domain;
– a collaborative task is an excellent exercise for students, as it motivates them and gives them a clear feedback on their work;
– the target language of CLEF eHEalth 2016 is French, the students' working language;
– the task's schedule was perfectly suited to the master's calendar.

Working as a team along the entire semester, the students were thus able (with help from their teachers) to submit two runs for the selected task, and got very satisfactory results for a first attempt.

This paper is organized as follows. Section 2 describes the tasks and get a closer look at the data. Section 3 gives a precise description of the different components of the system designed for the task, while the results are given and discussed in Section 4.

2 Task description

2.1 Overview

The LITL team participation only concerns the CLEF 2016 eHealth task 2 phase 1 addressing biomedical NER in French texts. The aim is to automatically detect and classify biomedical entities. Detection requires the system to provide start and end positions of each relevant entities. The classification step consists in associating each previously detected entity with one of the ten target categories corresponding to UMLS Semantic Group:

– Anatomy (`ANAT`),
– Chemical and Drugs (`CHEM`),
– Devices (`DEVI`),
– Disorders (`DISO`),
– Geographic Areas (`GEOG`),
– Living Beings (`LIVB`),
– Objects (`OBJC`),
– Phenomena (`PHEN`),
– Physiology (`PHYS`),
– Procedures (`PROC`).

Two text types are concerned with this task: drug inserts and biomedical research papers' titles and (sub)headings. Biomedical entities cover a wide range of linguistic expressions such as drug or city names, very common nouns (e.g. "*infirmière*" (*nurse*), "*études*" (*studies*)), technical specialized terms (e.g. "*thyroglobuline*", "*électronystagmographie*") or complex phrases (e.g. "*infectés par le virus de l'immunodéficience humaine*" (*infected with human immunodeficiency virus*), "*enfant âgé de plus de trois mois*" (*children over 3 months*)).

2.2 Data

The training data set corresponds to the QUAERO French Medical Corpus [11] previously used in CLEF 2015 eHealth task 1b [12] and made of two sub-corpora:

EMEA: 6 drug inserts written by the European Medicines Agency[1] for the general public ;

MEDLINE: 1665 titles and (sub)headings of biomedical academic papers indexed in the MEDLINE database[2].

Drug inserts are long texts (around 5,000 words/text on average) written in a less specialized language than MEDLINE's very short text segments (6 words/item on average), as illustrated in examples 1 from EMEA and 2 from MEDLINE, see below. MEDLINE texts are written with an uncommon syntax, considering that titles and (sub)headings usually correspond to noun phrases or non-finite sentences without final punctuation.

Example 1. *Il est utilisé en association avec d ' autres médicaments antiviraux dans le traitement des adultes et des enfants infectés par le virus de l ' immunodéficience humaine (VIH), le virus qui provoque le syndrome d ' immunodéficience acquise (SIDA).* `[EMEA/118_1]`

Example 2. *Hypersensibilité retardée dans les affections thyroïdiennes étudiée par le test de migration des leucocytes en présence de thyroglobuline humaine .* `[MEDLINE/4573749]`

Both sub-corpora have been manually annotated following similar guidelines [11]. Each annotation is encoded in the BRAT standoff annotation format[3] and includes for each entity an id, the relevant UMLS category, the offset position (start and end) and the textual content of the annotation. Example 3 gives the manual annotations associated with example 2 and Figure 1 its visualization via the BRAT annotation tool.

Example 3.
```
T1   DISO 0 25    Hypersensibilité retardée
T2   DISO 0 16    Hypersensibilité
T3   DISO 35 59   affections thyroïdiennes
T4   DISO 35 45   affections
T5   ANAT 46 59   thyroïdiennes
T6   PROC 75 107  test de migration des leucocytes
T7   PHYS 83 107  migration des leucocytes
T8   PHYS 83 92   migration
T9   ANAT 97 107  leucocytes
T10  CHEM 123 137 thyroglobuline
T11  LIVB 138 145 humaine
```

[1] http://opus.lingfil.uu.se/EMEA.php
[2] http://www.ncbi.nlm.nih.gov/pubmed/
[3] http://brat.nlplab.org/standoff.html

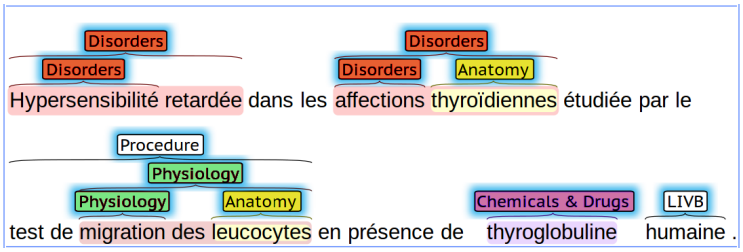

Fig. 1. Visualization of annotated entities through BRAT

As for the detection subtask, annotated entities can be single words (77%) or compounds (23%). Compound entities may be autonomous (e.g. "*prise en charge*"); nested such as in (3) where "*leucocytes*" is nested in "*migration des leucocytes*" which is in turn nested in "*test de migration des leucocytes*"; or discontinuous such as "*hépatites C*" and "*virus des hépatites C*" in (4).

Example 4. Prévalence des marqueurs des ***virus des hépatites*** *A , B ,* ***C*** *à La Réunion (Hôpital sud et prison de Saint Pierre).*
`[MEDLINE/10774493]`

```
T2   LIVB 29 34        virus
T5   DISO 39 48;57 58  hépatites C
T7   LIVB 29 48;57 58  virus des hépatites C
```

Table 1 gives an overview of the amount of annotated data per subset.

Table 1. Amount of annotated data per subset

Subset	**EMEA**	(%)	**MEDLINE**	(%)	**Total**	(%)
# of texts	6		1665		1671	
# of words	28215		10503		38718	
Annotated entities	4955		2977		7932	
Compounds entities	995	*20.1*	846	*28.5*	1841	*23.2*
Discontinuous entities	42	*0.8*	21	*0.7*	63	*0.8*

Because discontinuous entities are very rare, we decided to not detect such entities. Concerning nested entities which are more frequent, we implemented post-processing as described in Section 3.

As for classification subtask, entities are heterogeneously distributed among ten types. Half of the annotated entities are categorized as `DISO` (26%) or `CHEM` (25%). The rest spread mainly among `PROC` (16%), `ANAT` (11%) and `LIVB` (11%) while types `PHYS`, `OBJC`, `DEVI`, `GEOG` and `PHEN` are fairly rare. When compound, entities seem to be usually categorized as the first (head) component (e.g. "*virus des hépatites C*" is labeled `LIVB` as for "*virus*").
Preliminary linguistic observations of data reveal three main characteristics that

will constitute features for our system. First, the length of single-word annotated entities (in characters) seems longer than non-annotated words with an average of 9.5 characters against 5.5. Secondly, single-word entities show some morphological complexity with a use of recurrent productive affixes such as *–cyte*, *–thérapie*, *endocrino–*, *trachéo–*. Finally, annotated entities may be mainly characterized as technical vocabulary in the biomedical domain especially for main types (`DISO` and `CHEM`).

3 System description

In this section we present in details the system we designed for this task. Its cornerstone is a Conditional Random Field (CRF) classifier [6] that is used to train a model based on the manually tagged datasets. Although CRFs are easily applied thanks to readily available toolkits such as *CRF++*[4], they require a certain amount of preprocessing and are known to perform better when given additional information on the tokens to analyze. Also, they cannot by themselves provide all the solutions, especially when the target entities are nested and/or overlap, that's why several post-processing procedures were necessary as well.

3.1 Overview

As is mandatory with machine learning supervised techniques, we first had to build a model based on the training data. This was done according to the scheme presented in figure 2. As can be seen, this sequence starts with the raw text files from the training dataset, corrects their tokenization, performs POS tagging and syntactic parsing, then adds an additional set of features and finally provides, along with the annotated files, input for the CRF training phase. The result is a CRF model that can be used in the extraction phase.

The overall process used for extracting and categorizing entities from raw text files can be seen in figure 3. We can see that it applies roughly the same modules as for the training phase, but adds an additional post-processing step after the CRF classifier.

The stages are described in details in the following sections.

3.2 Preprocessing: correction, POS tagging and parsing

The first set of modules concerns the processing of text files. The main role of these modules is to provide a generic annotation layer for the tokens: POS tagging, lemmatization and dependency parsing, as these additional information are to be used by the CRF classifier.

For both phases, we used the Talismane toolkit [13] for POS tagging and dependency parsing, with the provided models for French and using the default

[4] `https://taku910.github.io/crfpp/`

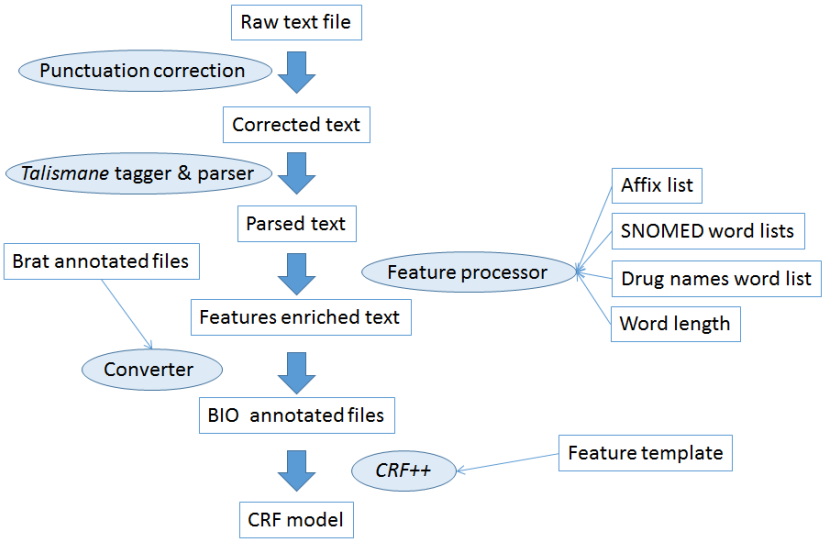

Fig. 2. System training phase

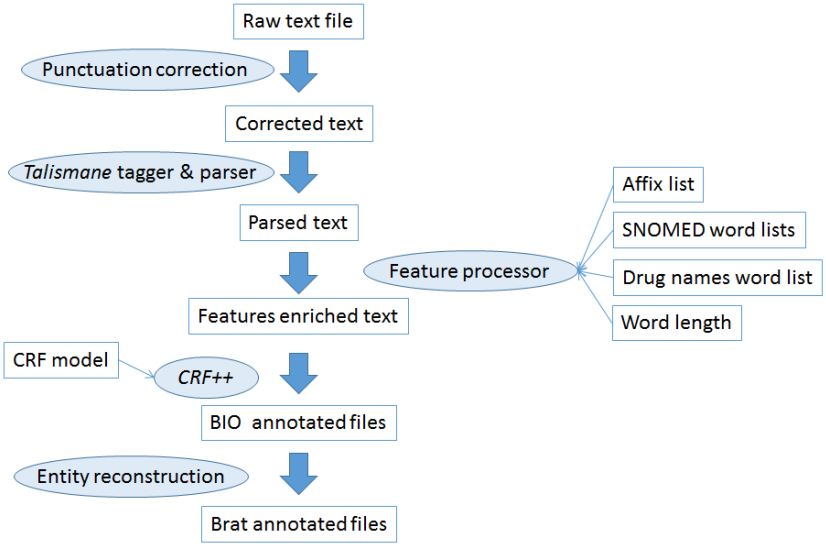

Fig. 3. Entities annotation phase

parameters. In addition to being a reliable tagger and dependency parser, Talismane provides the offset for each token, which is vital for dealing with the BRAT standoff annotation format.

The tokenization correction module has been developed in order to prevent many tagging errors due to the non-standard blank spaces that appear in both the training and test text files. The main problems were due to additional spaces before and/or after commas, hyphens, apostrophes, periods, digits, etc. For example, an extraneous space between "*l*" and the following apostrophe prevented the tagger and parser from identifying a determiner. This correction module is a simple Perl script using regular expressions to convert texts back to their supposedly original punctuation, while keeping the additional spaces in order to preserve the offsets for the annotated and extracted entities. For example, extract in (5) becomes (6) after correction:

Example 5. `Par␣conséquent␣,␣lorsque␣vous␣serez␣traité␣par␣TYSABRI␣,␣vous␣ne␣constaterez␣peut-être␣pas␣d␣'␣amélioration␣mais␣le␣traitement␣par␣TYSABRI␣pourra␣empêcher␣␣l␣'␣aggravation␣de␣votre␣maladie␣.␣[EMEA/425_6]`

Example 6. `Par␣conséquent,␣␣lorsque␣vous␣serez␣traité␣par␣TYSABRI,␣␣vous␣ne␣constaterez␣peut-être␣pas␣d'␣␣amélioration␣mais␣le␣traitement␣par␣TYSABRI␣pourra␣empêcher␣␣l'␣␣aggravation␣de␣votre␣maladie.␣␣`

The output of the parser follows a column-based format, in which each token is described on a separate line, as in the example (7) below (related to (5)), where columns indicate, in order: word form, POS tag, lemma, syntactic dependency, offset.

Example 7.

Par_conséquent	ADV	par_conséquent	mod	1398
,	PONCT	,	ponct	1413
lorsque	CS	lorsque	mod	1415
vous	CLS	vous	suj	1423
serez	V	être	aux_pass	1428
traité	VPP	traiter	sub	1434
par	P	par	p_obj	1441
TYSABRI	NPP	_	prep	1445
,	PONCT	,	ponct	1453
vous	CLS	vous	suj	1455
ne	ADV	ne	mod	1460
constaterez	V	constater	root	1463
peut-être	ADV	peut-être	mod	1475
pas	ADV	pas	mod	1485
d'	DET	de	det	1489
amélioration	NC	amélioration	obj	1493
mais	CC	mais	coord	1506
le	DET	le	det	1511
traitement	NC	traitement	suj	1514
par	P	par	dep	1525
TYSABRI	NPP	_	prep	1529
pourra	V	pouvoir	dep_coord	1537
empêcher	VINF	empêcher	obj	1544
l'	DET	l'	det	1553
aggravation	NC	aggravation	obj	1557
de	P	de	dep	1569
votre	DET	votre	det	1572
maladie	NC	maladie	prep	1578
.	PONCT	.	ponct	1586

As can be seen, unknown tokens (i.e. absent from the reference lexicon used by Talismane) have no corresponding lemma. We added correction procedure that reproduces the word form instead of the "_" placeholder. The POS tags indicate the token's nature (e.g. DETerminer, Verb, ADVerb, etc.) while the syntactic tag indicate its function in the sentence, more precisely the relation it has with its syntactic governor, such as object, modifier, etc. Details on both tagsets can be found in the Talismane documentation.

3.3 Adding features from external resources

In addition to the lemma, POS and syntactic relations, we added several features we deemed relevant with the task (see 2). These features are the following:

- does the token appear in a specific word lists?
- what is the token's length in number of characters?
- does the token begin with a recognizable prefix?
- does the token end with a recognizable suffix?

These features and the way they were calculated are described in the following paragraphs

Word lists

We used several word lists in order to extend the lexical coverage beyond what is present in the training datasets. We used two main sources: SNOMED and a list of commercial drug names.

SNOMED [14] is a well-known and used resource for biomedical NLP, as it contains extensive word lists. We selected the 4 most relevant top categories, and extracted all the single-word terms in each one.

As noted above (Section 2), manual observation of the training data (especially the EMEA subset) revealed that a large number of commercial names of drug were present and systematically tagged as CHEM entities. Given that SNOMED only contains scientific names of chemicals, we wanted to use an additional resource to tag these specific tokens, as performed by several named entities recognition systems [8]

We used the Vidal website[5], which is the reference compendium for pharmaceutical drugs in France. After downloading the list of raw drug names it provides, we developed an adhoc Perl script that identifies the main name through removal of extraneous information such as posology indications (i.e. transforming "ABSTRAL 100 μg cp subling" into "*Abstral*").

Table 2 gives the number of terms used and a few examples for each of the selected 5 categories.

Table 2. Word lists extracted from SNOMED resource and VIDAL database

Category	**Nb of terms**	**Examples**
SNOMED Chimie (`chem`)	3505	*toxine, magnésium, métoxyphénamine*
SNOMED Morphologie (`morph`)	1033	*microfracture, antéflexion, hypomyélinisation*
SNOMED Procédure (`proc`)	62	*cystométrogramme, électronystagmographie, thérapie*
SNOMED Agents physiques (`phys`)	378	*garrot, oxymètre, hémocytomètre*
VIDAL drug names (`chem`)	3873	*aspirine, nicopatch, ventoline*

In the end, each token is then associated with a tag (column) containing the SNOMED/VIDAL category (`chem`, `morph`, `proc` or `phys`) if its lemma appears in the corresponding word list, or *none* in other cases.

Token length

As noted above from an observation of the training data, word length can be a good indicator for biomedical terms. Given that the CRF classifier cannot deal with numerical feature, we used a qualitative measure of the length using the following scale: $\mathit{short} < 6 \leq \mathit{medium} < 10 \leq \mathit{long}$.

Affixes

Two additional features based on affixes were designed in order to help technical specialized vocabulary detection. It is well known that biomedical technical

[5] `http://www.vidal.fr`

terms are extensively coined through the use of standard suffixes and prefixes. Identifying these can be a useful addition to the necessarily limited reference word lists for entity recognition [8, 9]. We thus compiled lists of French prefixes and suffixes. We relied on existing lists (most of them seemingly compiled for the training of medicine students):

- *Lexique des racines, préfixes et suffixes des termes scientifiques et médicaux* by D. Pol, found at `http://www.didier-pol.net/`
- *Lexique des affixes (préfixes et suffixes)* by A. Abbara, found at `http://www.aly-abbara.com/litterature/medicale/affixes/a.html`
- *Terminologie médicale : préfixes et suffixes* by P. Cauwel, found at `https://sites.google.com/site/cauwelphilippe/Home/terminologie-medicale-prefixes-et-suffixes`

Once merged, these lists provided 394 prefixes and 126 suffixes.

For each token, the longest prefix (resp. suffix) in the list matching its lemma was used as an additional tag, and *none* when no match could be found.

3.4 From BRAT to BIO: selection of entities from the training set

Viewing the task of entity recognition and categorization as a tagging task requires the data (both training and testing) to be transformed into a compatible scheme. If the BRAT standoff annotation format is perfect for the visualization and declaration of these entities, it is not appropriate for an automated tagging process.

The BIO format is traditionally used for expressing token sequences such as chunks or named entities (cf. [3, 1, 2] for NER in biomedical texts). Its principle is the following: each token in a text can be at the `B`eginning, `I`nside or `O`utside an entity. Because entity categorization must be performed, the `B` (resp. `I`) letter is completed by the name of the entity's category. For example, to express the fact that "*Hypersensibilité retardée*" is a `DISO` we tag "*Hypersensbilité*" as `DISO_B` and "*retardée*" as `DISO_I`.

Thus defined, the task for the CRF classifier is to tag each token with one of the 21 possible value (there are 10 target entity categories, so 10 `X_B` and 10 `X_I`, but only one `O`). This task is thus formally similar to other NLP tagging tasks such as POS tagging, where a target value has to be decided according to the descriptive features of the token (and, in the case of CRF, to the features and values of the neighboring tokens).

An alternative scheme is known as BILOU and adds the possibility for a token to be the `L`ast token or to be the `U`nique one for this entity (cf. [4] for NER in biomedical domain). Although more precise, its main disadvantage is to increase the number of possible tags (41 in our case), making it harder for the classifier and generally requiring a larger amount of training data in order to be reliable (see [7] for a comparison between BIO and BILOU scores).

However, none of these schemes is able to faithfully represent nested, overlapping or discontinuous entities. As noted above, in section 2 (example 3), "*Hypersensibilité retardée*" contains two entities: "*Hypersensibilité*" and "*Hypersensibilité retardée*". Both are of the DISO category, and both begins with

the word "*Hypersensibilité*". However, only one of them can be represented using the BIO format. The same goes for more complex situations like the sequence "*test de migration des leucocytes*", still in example 3.

A decision had to be made when translating the annotated entities into the BIO format: do we keep the longest entities or the shortest? More precisely, do we consider "*leucocytes*" to be an ANAT entity by itself (thus receiving an ANAT_B tag) or part of the PHYS entity "*migration des leucocytes*" (with an PHYS_I tag)? This decision impacts the whole process, as what is decided for the training phase of the CRF will of course directly influence what is produced in the tagging phase.

This dilemma could be summarized as a choice between fewer but more complex entities versus more small ones. We decided to choose the latter, as we deemed it easier to "rebuild" complex entities from smaller ones than the other way around.

At this stage, each text is represented by a set of features available for each token, along with a BIO tag for the training data, as is the case in example (8) derived from (7):

Example 8.

```
Par_conséquent  ADV    par_conséquent mod       1398  long    none  none  none    O
,               PONCT  ,             ponct     1413  short   none  none  none    O
lorsque         CS     lorsque       mod       1415  medium  none  none  none    O
vous            CLS    vous          suj       1423  short   none  none  none    O
serez           V      être          aux_pass  1428  short   none  none  none    O
traité          VPP    traiter       sub       1434  medium  none  none  none    PROC_B
par             P      par           p_obj     1441  short   none  none  none    O
TYSABRI         NPP    TYSABRI       prep      1445  medium  none  none  chem    CHEM_B
,               PONCT  ,             ponct     1453  short   none  none  none    O
vous            CLS    vous          suj       1455  short   none  none  none    O
ne              ADV    ne            mod       1460  short   none  none  none    O
constaterez     V      constater     root      1463  long    con   none  none    O
peut-être       ADV    peut-être     mod       1475  medium  none  none  none    O
pas             ADV    pas           mod       1485  short   none  none  none    O
d'              DET    de            det       1489  short   none  none  none    O
amélioration    NC     amélioration  obj       1493  long    none  tion  none    O
mais            CC     mais          coord     1506  short   none  none  none    O
le              DET    le            det       1511  short   none  none  none    O
traitement      NC     traitement    suj       1514  medium  none  ment  none    PROC_B
par             P      par           dep       1525  short   none  none  none    O
TYSABRI         NPP    TYSABRI       prep      1529  medium  none  none  chem    CHEM_B
pourra          V      pouvoir       dep_coord 1537  medium  none  none  none    O
empêcher        VINF   empêcher      obj       1544  medium  none  none  none    O
l'              DET    l'            det       1553  short   none  none  none    O
aggravation     NC     aggravation   obj       1557  long    none  tion  none    O
de              P      de            dep       1569  short   none  none  none    O
votre           DET    votre         det       1572  short   none  none  none    O
maladie         NC     maladie       prep      1578  medium  none  none  none    DISO_B
.               PONCT  .             ponct     1586  short   none  none  none    O
```

3.5 CRF templates and parameters

As noted above, we chose to use a CRF classifier (CRF++) in order to predict the BIO tag for each token given the features described in the previous sections. Training the CRF requires to define a template, i.e. a selection of which pieces of information are given as an input to the model. The specificity of CRF systems is that they can take into account both the features associated to the target token and the ones of its neighbors. Thus, a template is used to define which feature(s) of which neighbor(s) are used.

In summary, after a number of trial runs using the training data, we opted for the following:

- word form of target token and of the tokens appearing in a size-2 window around it;
- POS tag of target token and of the tokens appearing in a size-2 window around it;
- length, prefix, suffix, and presence in word lists of the target token;
- BIO tag attributed to the previous token.

Moreover, the second run submitted uses the syntactic dependency tag of the target token, in addition to the previous features. This decision followed a first experiment to evaluate the contribution of each different features. The system evaluation was conducted by using the original train/devel split provided for the training data. Table 3 gives some results of a lesion studies, i.e. indicating for each feature how their removal from the template affects the task system's performances. F-scores were calculated by using the BRATeval.jar script provided by the organizers (exact matching option on true). As table 3 shows, syntactic feature contribution depends on the syntactic characteristics of texts: when texts have uncommon syntax as for MEDLINE items (see 2), using the syntactic information may worsen results. All other features have a positive contribution to the model for both target subsets, and thus has been kept in the system.

Table 3. Features contribution: F1 variation when feature is removed

F1-score	**EMEA**	**MEDLINE**
All features	0.5429	0.3821
- Syntactic relations	-0.0003	+0.0046
- Token length	+0.0109	-0.0038
- Affixes	-0.0085	-0.0227
- SNOMED and drugs lexicon	-0.0116	-0.0340

We considered separating the training subsets (EMEA and MEDLINE) and creating different models, but experiments in this direction were inconclusive, so we decided to use all the training data as a whole and to use the same model for both test subsets.

3.6 From BIO to BRAT: identifying complex named entity – NE

The final stage of the tagging phase is the translation from the BIO format back to the target BRAT standoff annotation format. Beyond simple format conversion, the main problem here is the reconstruction of entities, assuming the choices made in the training phase and addressing the limitations of the BIO format.

This stage is performed in two steps. The first one is a simple identification of entities, blindly following the BIO format. It means that, for each token tagged as `XXX_B` we extract one (and only one) entity, comprising the initial tokens along with all following tokens tagged as `YYY_I`. For example, we extract "*hyperplasie focale modulaire*" as a single `DISO` entity from the following output:

Example 9.
```
hyperplasie (NC)  DISO_B
focale      (ADJ) DISO_I
modulaire   (ADJ) DISO_I
```

In case of inconsistency between the categories (i.e. if `XXX` is different from `YYY`), we attribute the initial (`XXX`) category to the NEs.

As explained above, the choices made initially imply that at this stage we only get non-nested entities, although most of the time they are parts of larger entities. For example, given the following sequence in the CRF output:

Example 10.
```
tuberculose  (NC)  DISO_B
médiastinale (ADJ) ANAT_B
```

the first step extracts two separate entities: "*tuberculose*" as a `DISO` and "*médiastinale*" as a `ANAT`. In order to extract some of the more complex entities, we designed a few extraction patterns, based on the POS tags. The first pattern looks for a noun tagged as B, followed by an adjective tagged as B, i.e. the exact situation found in the example 10. Its output is the concatenation of both tokens, tagged with the category of the head noun. This means that this patterns produces "*tuberculose médiastinale*" as a `DISO`.

Similar patterns have been designed for longer sequences of adjectives: `NC (B) ADJ (B) ADJ (B or I)`, and for nominal compounds such as `NC (B) de/du/des/au/aux NC (B)` and `NC (B) P DET NC (B)`.

All extracted entities are collected in the BRAT standoff annotation format as noted above. The offsets are immediately available thanks to Talismane output format and the content of the entity is a simple concatenation of their tokens' word forms.

4 Results and discussion

In this last section we present and discuss the results obtained by our system on the task's test data.

4.1 Main results

As indicated in Table 4, our F1 scores are above the average (of submitted runs) for entity recognition in EMEA and slightly under it for MEDLINE with, for all subsets, a high precision and a low recall. The non-exact matching scoring procedure means that the offsets of the entities can partially overlap instead of being perfectly aligned. Run1 (without syntactic feature) and run2 (with

Table 4. LITL team scores on test data. For both target data and for each run, number of true positives (TP), fase positives (FP) and false negatives (FN), along with recall, precision and F1 scores, with exact and non-exact matching. Average scores were computed on all submitted runs.

EMEA	exact matching						no exact matching					
	TP	FP	FN	Precision	Recall	F1	TP	FP	FN	Precision	Recall	F1
LITL-run1	879	242	1325	0.7841	0.3988	**0.5287**	990	131	1069	0.8831	0.4808	0.6226
LITL-run2	867	264	1337	0.7666	0.3934	0.5199	995	136	1052	0.8798	0.4861	**0.6262**
Average scores				0.5250	0.4114	0.4350				0.6377	0.5141	0.5423
MEDLINE	exact matching						no exact matching					
	TP	FP	FN	Precision	Recall	F1	TP	FP	FN	Precision	Recall	F1
LITL-run1	998	556	2105	0.6422	0.3216	**0.4286**	1247	307	1531	0.8024	0.4489	**0.5757**
LITL-run2	989	561	2114	0.6381	0.3187	0.4251	1237	313	1544	0.7981	0.4448	0.5712
Average scores				0.5030	0.4264	0.4455				0.6387	0.5707	0.5859

syntactic feature) scores show little differences as expected, although run 1 gets higher scores for both subsets with exact matching. The order is partly different without exact matching where run2 gest slightly higher results for EMEA only. It should be noted that matching option only concerns entity delimitation and not disagreement on entity classification.

A closer look at the scores per category shows that we get the highest performance on EMEA texts for `LIVB` entities recognition with a 0.86 F1 (0.93 P and 0.8 R, 268 entities), followed by `PROC` (0.7 F1, 269 entities) and `CHEM` (0.67 F1, 885 entities). The detection of `DISO` entities, which cover the largest part of biomedical entities (988 entities in MEDLINE and 342 in EMEA), is better in MEDLINE texts (0.66 F1) than in EMEA texts (0.57 F1).

4.2 Test data set overview and errors observation

To better understand our system weaknesses, we propose a quick observation of the test dataset, then an overview of the false negatives (FN) and false positives (FP). First of all, an overview of test data reveals a difference from the training data that may explain some of our FN. Indeed, as indicated in table 5, while the proportion of compounds is approximatively the same than in training data set,

Table 5. Quantitative overview of test data set

Subset	**EMEA**	(%)	**MEDLINE**	(%)	**Total**	(%)
# of texts	4		833		1671	
# of words	12042		10871		22913	
Annotated entities	2070		3150		5220	
Compound entities	499	*24.1*	886	*28.1*	1385	*26.5*
Discontinuous entities	42	*2.0*	23	*0.7*	65	*1.2*

the proportion of discontinuous entities – which were not taken into account by our system – is double.

If we take a closer look at FP and FN, we may distinguish several cases (all subsequent examples come from the EMEA subset). First, FPs may be due to a disagreement on entity category such as the `CHEM` *polydextrose* or *diacétate de glycérol* which our system both classified as `DISO`. A comparison between run1 and the gold standard shows 187 cases of such miscategorization (60 in EMEA and 127 in MEDLINE), with quite a lot of confusions about subtle distinctions such as between `OBJC` – physical objects and `DEVI` – devices or between `PROC` – procedures and `CHEM` – chemical and drugs. As for example, in (11), "*injection*" was automatically categorized `PROC` instead of `CHEM` and "*seringue*" (*syringe*) was automatically categorized `OBJC` instead of `DEVI`.

Example 11. *L ' **injection** sous-cutanée est réalisée de la même façon qu ' avec une **seringue** classique .*`[EMEA/334_3]`
*The subcutaneous **injection** must be conducted in the same way as with a classical **syringe***

Classification errors may also be linked with polysemy as for example with the word "*bouton*" which means either a concrete object (*button*) or an anatomic/physiological entity (*pimple*).

A second group of FPs may be linked to the well-known problem of distinguishing between specialized vs. common usages. Such cases occur when a common word or compound was incorrectly recognized as biomedical entities (e.g. "*anomalies*" recognized as `DISO` or "*Mélange de couleur bleu*" as `CHEM`).

A last possible explanation for FPs is related to compounds boundaries as for example when a `CHEM` is syntactically linked to a `LIVB` but not manually annotated as a compound (e.g. "*olanzapine chez les enfants*" or "*olanzapine sur les protéines*"). Same mistakes may be observed due to prepositional phrase attachment ambiguity as for example in (12)

Example 12. *Traitement de l' ankylostomiase par le tétrachloréthylène chez l' adulte et le grand enfant .* `[MEDLINE/13515790]`

where our system recognized "*ankylostomiase par le tétrachloréthylène*" as a `DISO` because of the pattern `NC (B) P DET NC (B)` whereas th prepositional phrase "*par le tétrachloréthylène*" is syntactically attached to "*Traitement*" and not to "*ankylostomiase*".

Moreover, when exact matching, other FPs may be observed such as "*troubles hématologiques périphériques*" detected as a single entity instead of two entities "*troubles hématologiques*" and "*périphériques*".

This compound boundaries problem has very strong consequences on recall and may explain the large amount of FNs (more than 1,300 in EMEA and 2,100 in MEDLINE, see Table 4). Indeed, when "*troubles hématologiques périphériques*" is detected instead of two entities "*troubles hématologiques*" and "*périphériques*", it causes one FP and two FNs. A simple way for dealing with this problem will be to systematically split detected compounds into as many single entities as single words.

The converse is also observed i.e. when our system extracts only one term from a compound, as for example "*Olanzapine Teva*" manually annotated as `CHEM` while our system recognized only "*Olanzapine*" and not "*Teva*". Such FNs were essentially due to the limited coverage of our external resources.

References

1. Abacha, A.B., Zweigenbaum, P.: Une étude comparative empirique sur la reconnaissance des entités médicales. Traitement Automatique des Langues 53(1), 39–68 (2012)
2. Ben Abacha, A.: Recherche de réponses précises à des questions médicales : le système de questions-réponses MEANS. Thèse de doctorat en informatique, Université Paris Sud - Paris XI (2012)
3. de Bruijn, B., Cherry, C., Kiritchenko, S., Martin, J., Zhu, X.: Machine-learned solutions for three stages of clinical information extraction: the state of the art at i2b2 2010. Journal of the American Medical Informatics Association 18(5), 557–562 (2011)
4. Ghiasvand, O.: Disease Name Extraction from Clinical Text Using Conditional Random Fields. Master's thesis, University of Wisconsin-Milwaukee (2014)
5. Kelly, L., Goeuriot, L., Suominen, H., Névéol, A., Palotti, J., Zuccon, G.: Overview of the clef ehealth evaluation lab 2016. In: Proceedings of CLEF 2016 - 7th Conference and Labs of the Evaluation Forum, Lecture Notes in Computer Science (LNCS). Springer (September 2016)
6. Lafferty, J., McCallum, A., Pereira, F.C.: Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of the 18th International Conference on Machine Learning (ICML). pp. 282–289 (2001)
7. Leaman, R., Gonzalez, G., et al.: Banner: an executable survey of advances in biomedical named entity recognition. In: Pacific Symposium on Biocomputing. vol. 13, pp. 652–663 (2008)
8. Liu, S., Tang, B., Chen, Q., Wang, X., Fan, X.: Feature engineering for drug name recognition in biomedical texts: Feature conjunction and feature selection. Computational and mathematical methods in medicine 2015 (2015)
9. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Lingvisticae Investigationes 30(1), 3–26 (2007)
10. Névéol, A., Goeuriot, L., Kelly, L., Cohen, K., Grouin, C., Hamon, T., Lavergne, T., Rey, G., Robert, A., Tannier, X., Zweigenbaum, P.: Clinical information extraction at the clef ehealth evaluation lab 2016. In: Proceedings of CLEF 2016 Evaluation Labs and Workshop: Online Working Notes. CEUR-WS (September 2016)

11. Névéol, A., Grouin, C., Leixa, J., Rosset, S., Zweigenbaum, P.: The Quaero French medical corpus: A ressource for medical entity recognition and normalization. In: Proc BioTextM, Reykjavik (2014)
12. Névéol, A., Grouin, C., Tannier, X., Hamon, T., Kelly, L., Goeuriot, L., Zweigenbaum, P.: Clef ehealth evaluation lab 2015 task 1b: clinical named entity recognition. In: Proceedings of CLEF (2015)
13. Urieli, A., Tanguy, L.: L'apport du faisceau dans l'analyse syntaxique en dépendances par transitions : études de cas avec l'analyseur Talismane. In: Actes de la 20e conférence sur le Traitement Automatique des Langues Naturelles (TALN'2013). pp. 188–201. Les Sables d'Olonne, France (2013)
14. Wang, A.Y., Sable, J.H., Spackman, K.A.: The SNOMED clinical terms development process: refinement and analysis of content. In: Proceedings of the AMIA Symposium. p. 845. American Medical Informatics Association (2002)