Romain Potier-Ferry

Rapport de stage préliminaire

encadrant : Delphine Bernhard

Évaluation de CasEN

Présentation de l'outil

La première partie de mon travail a consisté à procéder à une évaluation de l'outil de reconnaissance d'entité nommées CasEN[1] utilisant la plate-forme Unitex[2]. Cet outil reconnaît 10 types d'entités de base : les personne, les animaux, les fonctions, les organisations, les lieux, les résultats de la production humaine, les dates et heures, les montants et les événements. Chaque catégorie de base est étendue afin de préciser au mieux la nature de l'entité. Par exemple l'entité *Hum* (humain) est étendue pour inclure les civilités ou les adjectifs ethniques. La catégorie *Fonc* (fonction) est étendue afin de reconnaître le type auquel elle appartient (politique, religieuse…). Le lecteur pourra trouver une liste de toutes les catégories reconnues par l'outil sur le site officiel.

J'ai tout d'abord commencé l'analyse de but en blanc mais je me suis vite rendu compte qu'il était nécessaire que je cherche tout d'abord à comprendre comment celui-ci fonctionnait en analysant une partie des graphes qu'il utilise. L'outil mettant en application 468 graphes qui s'appellent mutuellement, j'ai décidé de partir des transducteurs « finaux » qui sont appelés par la cascade elle-même et de redescendre graduellement dans la hiérarchie des graphes.

L'outil consiste en un ensemble de grammaires Unitex de nature différentes : des listes, des *tools* dont le but est généralement la normalisation du texte (supprimer les points dans les sigles par exemple), des *patterns* dont le but est de reconnaître des motifs particuliers et enfin des transducteurs dont le but est d'annoter le texte et donc de mettre en évidence les entités nommées que l'outil a été capable d'identifier. Deux dictionnaires sont également fournis avec CasEN afin de renforcer les outils linguistiques de base d'Unitex. Ces dictionnaires contiennent essentiellement des listes de prénoms, de lieux, de professions ou de personnalités.

Pour la reconnaissance, l'outil se base à la fois sur les catégories présentes dans les dictionnaires mais également sur des indices de surfaces « purs ». Par exemple, le graphe utilisé pour la reconnaissance des villes utilise à la fois les catégories *Toponyme* et *Ville* des dictionnaires mais également des formes telles que les mots *villes/villages/bourg* ou encore des constructions typiques telles que *<MotMaj>*[3] *sous/les/la <MotMaj>* (ex : Marne la Vallée), ou encore *<MotMaj> La nouvelle/La neuve* ainsi que des noms contenant des particules telles que *Los/San* (ex : Los Angeles).

Il en va de même pour les noms propres qui sont reconnus à la fois en utilisant la catégorie *Prenom* reconnue par Unitex à l'aide des dictionnaires suivie d'un pattern <MojMaj> pour les noms propres « courant » mais également des motifs particuliers qui combinent expression régulières restreignant certains résultats (ces dernières permettant d'éliminer des résultats incorrects tels que les sigles) et formes de surfaces typiques de noms moins usuels telles que *Mac*, *El-*, *bin*, *van* ou *von*, ces derniers regroupant généralement des noms à forte connotation ethnique. Cette dernière méthode utilise généralement le masque <PRE> d'Unitex qui représente n'importe quel nom commençant par une majuscule qu'elle que soit sa nature sémantique. Les expressions régulières sont alors utilisées en contexte droit « de négation » qui permettent d'exclure certaines formes qui ne doivent pas être reconnues.

1 http://tln.li.univ-tours.fr/Tln_CasEN.html
2 http://www-igm.univ-mlv.fr/~unitex/
3 Ce formalisme est ici utilisé pour représenter un mot débutant par une majuscule.

L'annotation elle-même se fait par encapsulation en reconnaissant tout d'abord les unités de plus petites tailles (telles que les prénoms, les noms ou les professions) qui seront ensuite balisées en unités plus grandes. Par exemple le syntagme *capitaine Sanogo* sera annoté de la façon suivante :

{\{ capitaine\,\.entity\+fonc\+mil\+grffoncMilitaire\}\{\{ Sanogo\,\.N\+nom\}\,\.entity\+pers\ +hum\},.entity+pers+hum+grfpersNomCtxtG}

Comme nous pouvons le voir le mot *capitaine* est reconnu comme étant une fonction militaire et *Sanogo* est reconnu comme étant un nom. Le tout est alors encapsulé en une seule entité représentant une personne humaine. L'annotation suit le formalisme *Cassys*[4] qui permet entre autre de faire apparaître le nom du graphe responsable de l'annotation afin de faciliter le debugging. Un autre exemple serait le syntagme *le président Amadou Toumani Touré* qui sera étiqueté :

le{\{président\,\.entity\+fonc\+pol\+grffoncGouvernement\}\{\{ Amadou\,\.N\+Prenom\}\{ Toumani Touré\,\.N\+nom\}\,\.entity\+pers\+hum\},.entity+pers+hum+grfpersPrenomNomCtxtG}

Nous pouvons remarquer que les déterminants sont exclus des entités lorsque qu'il n'en font pas partie intégrante (comme ce serait le cas pour *président de **la** Chambre de Commerce*)

Résultats de l'évaluation

Pour l'évaluation elle-même, je me suis inspiré de la méthode utilisée par les développeurs de l'outil[5]. J'ai pour cela utilisé quatre types de précision et de rappel :

précision absolue = résultats corrects absolus/nombre de résultats
précision relative = résultats corrects relatifs/nombre de résultats
précision tagging = résultats bien étiqueté/nombre de résultats
précision majuscule = nombre de mots majuscule corrects/nombre de mots majuscule détectés

rappel absolu = résultats corrects absolus/nombre d'entités contenues dans le documents
rappel relatif = résultats corrects relatifs/nombre d'entités contenues dans le document
rappel tagging = résultats bien étiquetés/nombre d'entités contenues dans le document
rappel majuscule : nombre de mots majuscules corrects/nombre de mot majuscule document

résultats corrects relatifs = nombre de résultats total - nombre d'erreurs
résultats corrects absolus = nombre de résultats total - nombre d'erreurs - nombre de résultats incomplets
résultats bien taggés = nombre de résultats total - nombre d'erreurs – nombre de résultats mal taggés

Dans les mesures précédentes, le « résultat total » représente le nombre d'entité nommées annotées dans le document, le « nombre d''erreurs » représente le nombre d'entité nommées qui ont été annotées comme telle mais qui n'en sont pas (ce nombre est généralement bas) auquel s'ajoute le nombre d'entités qui ont été à la fois mal étiquetée et qui sont incomplètes (mais qui sont cependant bien des entités nommées). Le « nombre de résultats incomplets » représente les entités dont seule une partie a été détectée mais dont la catégorie de base est correcte, par exemple le syntagme *chef de la junte* a été annotée :
{chef,.entity+fonc+pol+grffoncGouvernement} de la junte
Nous pouvons voir que *chef* a bien été détecté comme une *fonction* mais l'outil aurait également dû détecter le reste du syntagme. Ce genre de résultat rentre donc dans la catégorie « résultats

4 Voir le manuel d'Unitex pour une présentation exhaustive du formalisme.
5 Cette méthode est présentée sur le site officiel de CasEN.

incomplets ». Les « résultats bien taggés » représentent les entités détectées qui possèdent une étiquette correcte et enfin les « mots majuscules corrects » représentent les entités nommées détectées commençant par une majuscule dont j'ai considéré que l'extraction serait facile. Les résultats de ce genre rassemblent les entités majuscules formant une seule unité telles que *{AFP,.sigle+grforgSigle}* ou bien les entités contenant plusieurs mots en une seule unité mais dont le premier mot est une majuscule. Dans ce cadre, une entité telle que *{chef de l'Etat,.entity+fonc+pol+grffoncGouvernement}* sera considérée comme une erreur.

L'évaluation elle-même a porté sur 12 textes dont les statistiques sont les suivantes :

Entités Nommées dans document : 1095
EN détectées : 672
Erreurs : 59
Résultats corrects relatifs : 613
EN incomplètes : 29
Résultats corrects absolus : 584
EN mal taggées : 120

mots majuscules dans le document : 933
mots majuscules détectés : 594
mots majuscules corrects : 581

Les résultats des mesures nous donnent alors :

précision absolue : 86.90 %
précision relative : 91.22 %
précision tagging : 73.36 %
précision majuscule : 97.81 %

rappel absolu : 53.33 %
rappel relatif : 55.98 %
rappel tagging : 45.02 %
rappel majuscule : 62.27 %

Discussion des résultats

En ce qui concerne la précision, les résultats que j'ai obtenu sont relativement proches de ceux obtenus par les développeurs lors de leur évaluation. Le rappel quant à lui est substantivement plus bas (environ 10 % inférieurs). Ceci peut tout d'abord s'expliquer par le fait que je ne connais pas l'outil en profondeur et qu'il est fortement possible que je lui en « demande trop ». Il est également possible que ce score ait été atténué par le fait que les textes que j'ai utilisé proviennent d'une extraction automatique et sont parfois mal formatés : ils contiennent des mots clés (tags), certaines phrases ne possèdent pas de point final ou sont incomplètes, ce qui induit Unitex en erreur. Certains d'entre eux contiennent également des commentaires d'utilisateurs, ces derniers n'étant pas soumis à des règles d'édition précises contiennent souvent des fautes d'orthographes, des exclamations en majuscules et sont généralement mal formatés (phrases incomplètes, ponctuation manquante). Ne possédant pas d'information au sujet des textes utilisés par les développeurs, je ne peux pas affirmer que les conditions d'évaluation permettent vraiment une comparaison objective des résultats. De plus ma méthode d'évaluation n'est pas très robuste car je n'ai pas annoté de texte de comparaison. J'ai compté le nombre d'entités nommées présentes dans le texte et comparé ce résultat aux résultats

de la cascade que j'ai ensuite analysés individuellement en étudiant également leur contexte à l'aide du concordancier d'Unitex. Je pense qu'il serait bon de procéder à une évaluation plus précise pour la fin du stage en comparant avec des textes annotés manuellement.

Cependant, je suis en mesure de faire quelques remarques quant au fonctionnement de l'outil. Tout d'abord, j'ai remarqué qu'il n'était pas capable de distinguer les prénoms, gentilés ou toponymes lorsque ceux-ci sont écrits en minuscules. Ceci provient principalement du fait que les patterns de reconnaissance de ces derniers font appel à des <MotMaj> ou à l'étiquette <*Prenom*> d'Unitex or celui-ci ne reconnaît pas non plus les prénoms en minuscules et les range dans sa liste de mots inconnus. Certains noms propres composés tels que *Château Pavie-Maquin* (domaine vignoble) ne sont pas reconnus entièrement. On peut observer également quelques résultats à priori inexplicables. En reprenant l'exemple précédent :

{Château,.entity+loc+admi+grflocDico}{ Pavie,.entity+loc+admi+grflocDico}-Macquin, Château Dassault, […]

On peut voir que le second vignoble n'a pas été détecté, or il apparaît exactement dans le même contexte que le premier.

Certaines erreurs sont également dues à l'ambiguïté des dictionnaires qui reconnaissent certains mots uniquement à cause de leur polysémie.

{ Bouquet,.entity+loc+admi+grflocPersOrgSeul} sur des fruits rouges et noirs

Ici le mot *bouquet* a été identifié comme un lieu bien qu'il s'agisse du bouquet d'un vin. Ces erreurs pourraient être évitée à l'aide de grammaires ELAG (grammaires spécialement dédiées à la désambiguïsation des dictionnaires) mais le temps développement de telles grammaires est considérable.

En ce qui concerne l'étiquetage, on peut tout d'abord remarquer que pour les noms propres il arrive souvent que le nom de famille ne soit pas identifié comme tel bien que l'entité ait été reconnue en entier.

{\{Christophe\,\.N\+Prenom\} Champenois,.entity+pers+hum+grfpersDico}

On peut voir dans cet exemple que le prénom a bien été identifié mais pas le nom de famille. Contrairement à l'exemple suivant :

{\{ Jean-Paul\,\.N\+Prenom\}\{ Hebrard\,\.N\+nom\}\,\.entity\+pers
+hum\},.entity+pers+hum+grfpersPrenomNomCtxtG}

Ceci peut s'expliquer par le fait que le graphe *grfpersDico* ne fait pas appel au pattern qui étiquette les noms de famille, ce qui est le cas pour le graphe *grfpersPrenomNomCtxtG*.

Certains graphes possèdent la capacité de reconnaître les même unités que d'autre graphes, il est alors uniquement question du premier qui le reconnaîtra. Ceux-ci étant parfois tellement intriqués qu'il paraît très difficile de prévoir quel chemin sera parcouru plutôt qu'un autre.

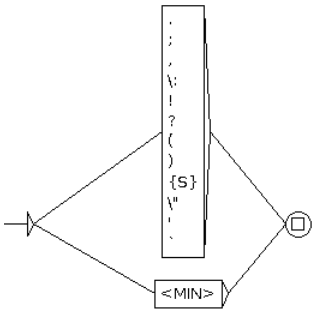
Une autre remarque peut être faite au sujet du fait que les noms de famille employés seuls ne sont reconnus que dans des contextes très particuliers - en présence d'indice de civilité par exemple - ce qui les empêche parfois d'être reconnus dans certains contextes (par exemple dans les textes sportifs où il est courant de faire référence aux personnes uniquement par leur nom de famille) ou bien

entraîne un mauvais étiquetage.

Ex : *au grand dam de{ Gerland,.entity+loc+admi+grflocDico} qui s'ennuyait ferme.*

Ici *Gerland* fait référence à une personne et non à un lieu et cela souligne un double problème. Le nom n'a pas été identifié comme tel non seulement car il apparaît seul mais également comme un lieu à cause de l'ambiguïté du dictionnaire.

Une dernière remarque peut être faite à propos de la façon dont CasEN définit les limites d'une entité. Les transducteurs ne les annotent que lorsqu'elles apparaissent dans des contextes particuliers.



Comme on peut le voir dans l'illustration ci-dessus, une entité doit être précédée ou bien d'un signe de ponctuation courant (ce qui inclut les marqueurs de début de phrase utilisés par Unitex) ou bien d'un mot commençant par une majuscule. Cette méthode permet de garder un certain contrôle sur les contextes d'apparition des entités et ainsi maintenir une précision élevée. Cependant, bien que cette méthode soit très efficace dans la plupart des cas, tout particulièrement lorsque les textes sont bien formatés, elle présente deux inconvénients qui peuvent entraîner une baisse du rappel dans certains cas particuliers.
Tout d'abord, le fait de limiter ainsi les marques de ponctuation empêche l'outil de reconnaître certaines entités lorsqu'elles apparaissent en présence de marqueurs qui ne sont pas définis ici.

*_ Lorient 1_1 Brest _ Nancy 0_1 Evian _ Lille 0_3 Lyon _ Sochaux 2_1 Montpellier _ Saint-Etienne 1_0 Nice _ Marseille 1_1Valenciennes _ Rennes 1_*

Cet exemple provient d'un texte sportif possédant des récapitulatifs de résultats de matchs de football. N'ayant pas accès au texte de base, je ne peux dire si ce formatage est dû à l'extraction automatique ou si le texte possédait cette forme dès le départ mais on peut remarquer qu'aucun des noms de ville n'a pu être reconnu car les marqueurs qui les entourent ne sont pas acceptés dans le pattern.

Deuxièmement, le fait qu'une entité ne puisse pas être précédée d'un mot majuscule signifie que certaines entités seront ignorées dans certains contextes.

Après 32 longues heures […]

Dans cet exemple, nous pouvons voir que le syntagme *32 longues heures* n'a pas été reconnu comme une mesure de temps (ces mesures sont cependant censées être détectées par CasEN) car il est précédé d'un mot commençant par une majuscule.
Ce phénomène peut également être observé lorsque pour une raison quelconque certains mots d'une

suite de mots tous débutant par une majuscule ne sont pas détectés. Leur non-détection entraîne alors une non-détection en chaîne des mots qui les suivent car ils seront alors précédés de mots commençant par une majuscule.

{Communauté économique des Etats,.entity+org+grforgDivers} d'{Afrique de l'Ouest,.entity+loc+admi+grflocOrgSuiviDeLoc}

Finalement, dans cet exemple, on peut voir que l'entité a été coupée en deux bien que l'outil aurait dû reconnaître le bloc en entier, cela est dû au fait que *d'* se trouve précédé d'un mot commençant par une majuscule, ce qui n'est pas prévu par le graphe que nous avons vu précédemment. Ces cas sont cependant relativement rares et n'entraînent pas de lourdes conséquences mais il me semblait important de le faire remarquer car c'est un problème que j'ai fréquemment rencontré lors de la mise au point de mon propre outil.

Mise au point d'un outil d'identification d'entité nommées dans le cadre du projet Logoscope

Il m'a été ensuite demandé de mettre au point un outil capable de détecter les noms propres, les noms de lieux et les gentilés lorsque ceux-ci commencent par une majuscule. J'ai tout d'abord commencé un brouillon de projet n'utilisant aucun graphe provenant de CasEN. Pour ceci je me suis principalement aidé des catégories sémantiques reconnues par Unitex telles que *Toponyme*, *Prenom*, *Npr*... Cette étape m'a permis de mettre en lumière certains problèmes qui pouvaient apparaître lors du développement d'une telle application.

Après avoir réussi à obtenir un squelette de projet qui fonctionnait sans trop produire d'erreur, j'ai commencé à intégrer des grammaires provenant de CasEN, ces dernières étant disponibles en licence libre. Cette étape m'a permis de comprendre la philosophie derrière cet outil mais également de voir comment certains problèmes pouvaient être résolus. Je me suis également inspiré de l'organisation du projet CasEN dans le sens où j'ai créé plusieurs types de graphes : des *patterns* de reconnaissance, des *listes* et enfin des transducteurs qui font appel à ces patterns. J'ai également utilisé les patterns de fin et de début d'entité défini par CasEN, ainsi que certaines listes.

J'ai complètement changé le système d'annotation afin de permettre une extraction aisée des entités identifiées. Les transducteurs appliquent des étiquettes sous forme de balises XML qui font apparaître le type d'entité reconnu ainsi que le graphe responsable de la reconnaissance. Lorsque le graphe contient plusieurs le chemin reconnu est indiqué à la suite du nom du graphe.

Les balises se présentent sous cette forme :

<ent type="x" graphes="nomDuGraphe.grf_y></ent>

où *x* représente donc le type d'entité[6] et *y* représente le chemin du graphe responsable de la détection lorsque cela s'applique.

Je vais à présent détailler le fonctionnement des transducteurs finaux de la cascade et expliquer certaines décisions que j'ai eues à prendre lors de la mise au point de l'outil.

La cascade fait tout d'abord appel à deux *tools* empruntés à CasEN dont le but est la normalisation

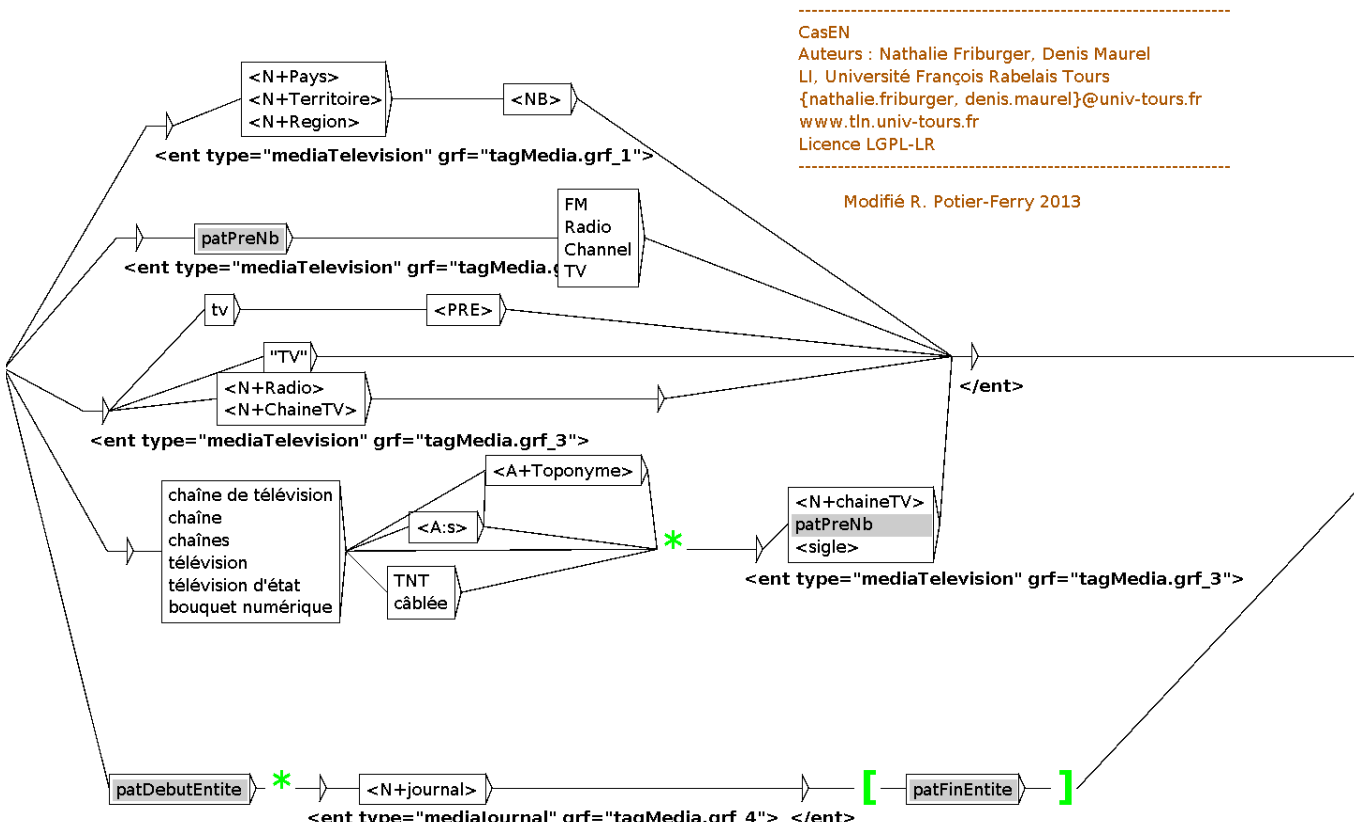
6 Parmi : *média, mots inconnus, dynaste, noms propres, pays, continent, ville, région, sigles, noms et prénoms* (employés seuls).

des sigles contenant des points. Le premier des graphes repère les sigles contenant des points et les étiquette. Le second graphe extrait ces sigles étiquetés et les réécrit sans les points. Par exemple *C.G.T* sera finalement réécrit *CGT*.

Le troisième transducteur appelé *tagMotInconnu* a pour but d'étiqueter les mots commençant par une majuscule qui n'auraient pas été reconnu par Unitex. Son fonctionnement est relativement simple comme on peut le voir sur l'illustration suivante.

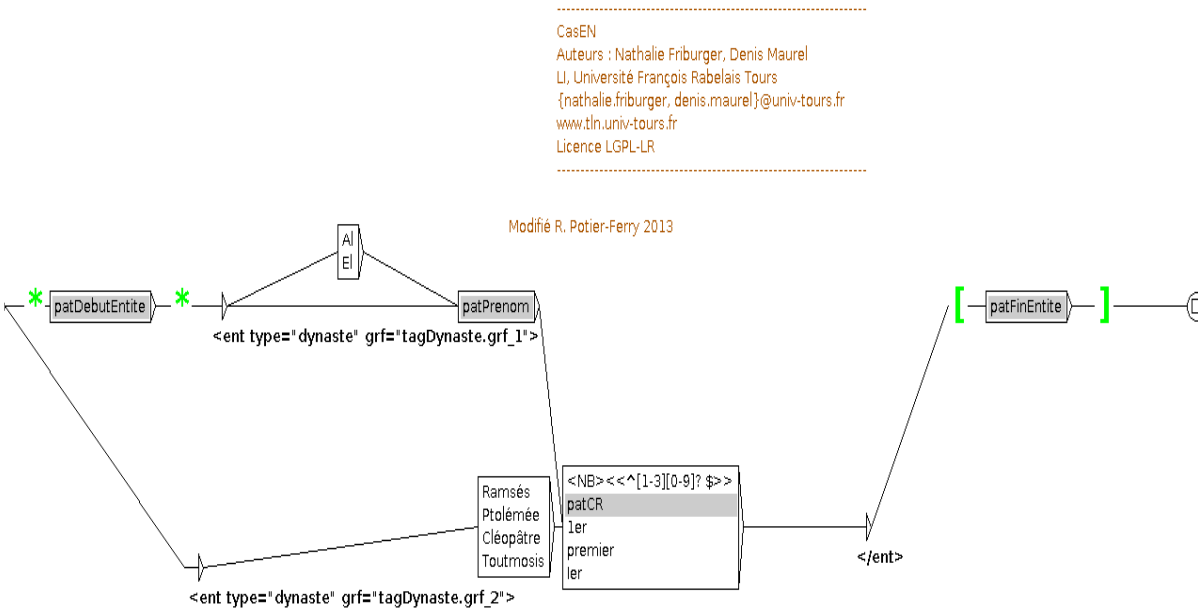
Le transducteur consiste a appliquer une étiquette à tout mot commençant par une majuscule qui ne serait pas reconnu par le dictionnaire. Ce cas est relativement rare car Unitex a tendance à attribuer la catégorie <*Npr*> à ce genre de mot. Nous pouvons voir que les contextes droits « de négation » permettent, entre autre, d'appliquer des restrictions sur les catégories reconnues par le pattern. Cette technique sera largement employée dans les graphes afin de limiter les erreurs dues à l'ambiguïté du dictionnaire.

Le quatrième transducteur, *tagMedia* a pour but de reconnaître différents types de média tels que les noms de chaînes de télévision, les noms de radio ou de journaux.



Je suis parti d'un graphe provenant de CasEN que j'ai modifié pour qu'il détecte les noms de chaîne de télévision contenant le nom d'un pays ou d'une région (ex : *France 24*). Le graphe de départ permettait uniquement de reconnaître les noms de chaînes identifiés par Unitex (voir chemin 3). Le premier chemin du graphe permet de palier à ce problème. J'ai modifié le chemin 3 afin d'exclure de la concordance les indices de surface utilisés pour la détection à l'aide d'un contexte « gauche ». J'ai enfin rajouté le chemin numéro 4 qui permet d'identifier les noms de journaux reconnus par Unitex. Il devrait être possible d'utiliser une liste de surface comme pour le chemin 3 afin de détecter les noms de journaux qui ne seraient pas reconnus par le dictionnaire. Je n'ai cependant pas eu besoin d'implémenter cette méthode jusqu'à présent.

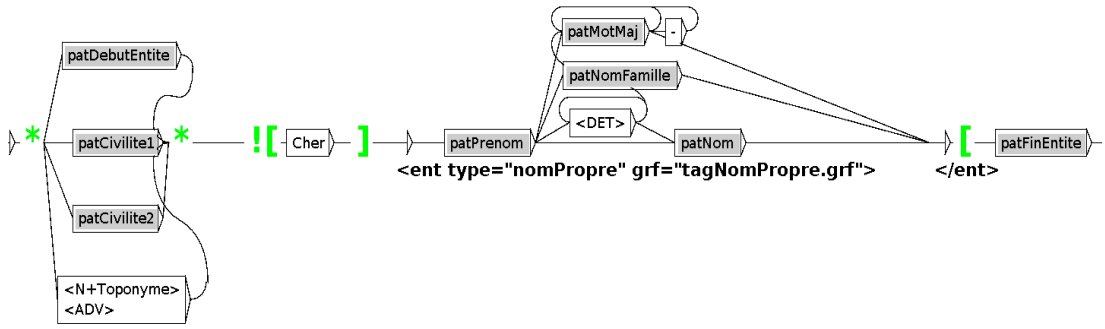
Le transducteur suivant, *tagDynaste* reconnaît les noms de rois ou de dynastes divers en utilisant essentiellement des indices de surface.



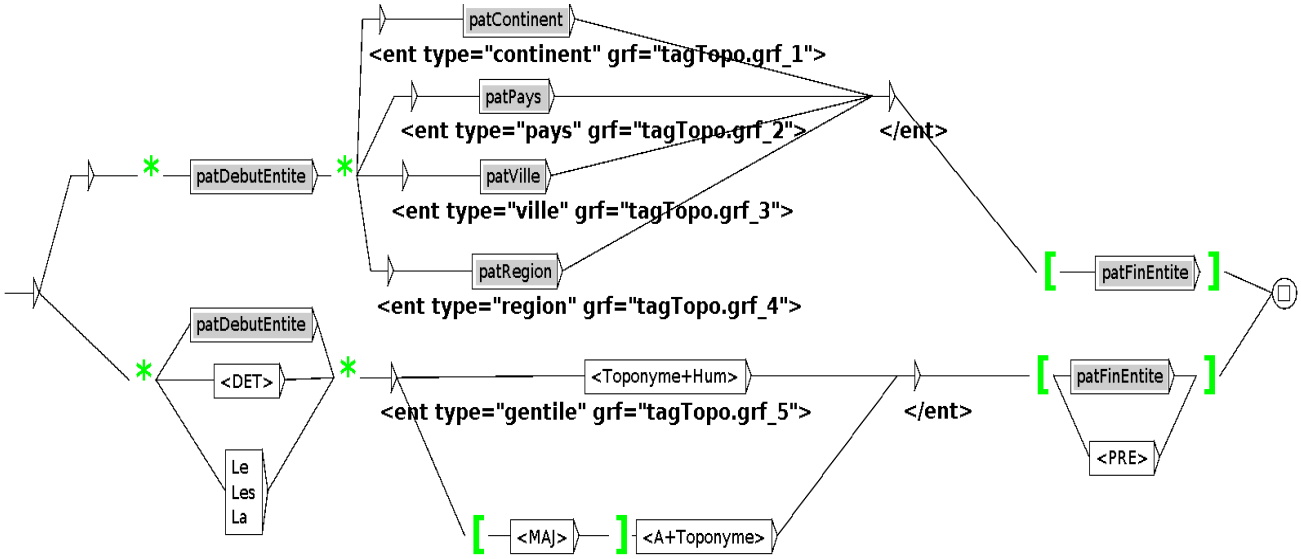
Encore une fois je suis parti d'un graphe de CasEn que j'ai modifié afin de supprimer les références à des annotations provenant de l'outil lui-même. Le premier chemin a été modifié dans le but de supprimer certains indices de surface qui ne nous sont pas utiles dans le contexte de ce travail. *patCR* fait référence à un pattern de reconnaissance de chiffres romains. Ce graphe est placé à cet endroit de la cascade car le transducteur suivant a la capacité de reconnaître certains de ces motifs, je l'ai donc placé avant afin de permettre une annotation plus précise.

La cascade fait ensuite appel au transducteur *tagNomPropre* qui reconnaît les patterns de type *<prénom> <nom de famille>*. Il est également capable de reconnaître les noms contenant des particules tels que *Charles de Gaule*. Les graphes *patCivilité1/2* contiennent des indices de surface tels que *monsieur, Mr., Ms, Sir.*. Les contextes gauches incluent également une liste de catégories qui peuvent précéder ce genre de pattern, cela permet de contourner certaines difficultés introduites par l'utilisation des patterns *patDebut/FinEntite* que je mentionnais précédemment.
Certaines parties de ce graphe paraissent redondantes mais elles permettent d'augmenter le rappel et la précision du graphe. Le graphes *patNomFamille* est une modification du graphe de CasEn que j'ai

mentionné précédemment. Cette modification consiste essentiellement à limiter les catégories reconnues par le patron afin d'éliminer les erreurs dues à l'ambiguïté.



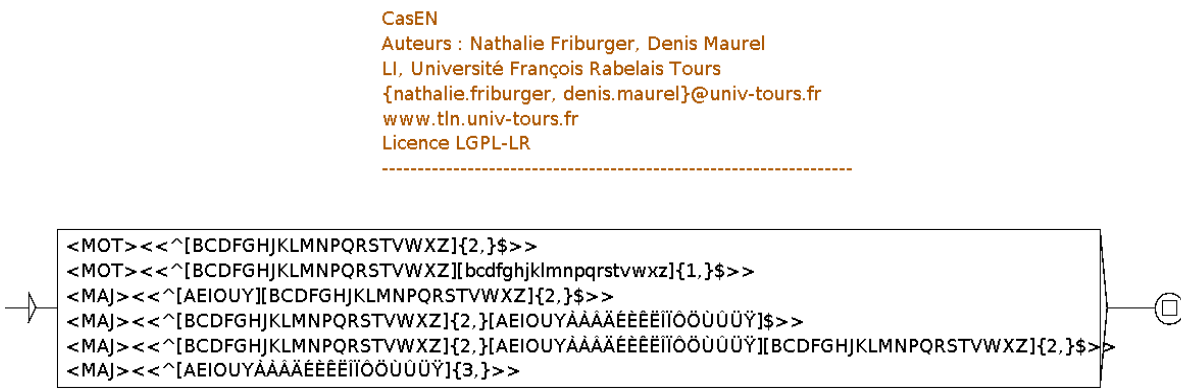
Le prochain transducteur a pour charge d'annoter les noms de lieux et leurs dérivés tels que les gentilés.



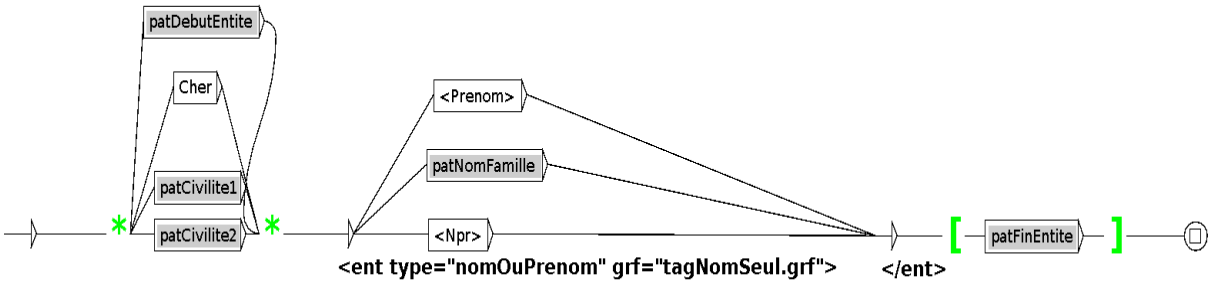
Le graphe *patContinent* contient une liste des 5 continents ainsi que certaines variations courantes telles que *Eurasie*, *patPays* utilise la catégorie *<Toponyme:Pays>* d'Unitex avec une restriction pour l'empêcher de détecter les gentilés. *patVille* utilisent la catégorie *<Toponyme:Ville>* avec certaines restrictions (gentilés, ambiguïté avec les prénoms...) ainsi que des motifs de surface courants inspirés de CasEn (utilisation de préposition ou de déterminant par exemple). *patRegion* utilise les catégories *<Toponyme:Region>* et *<Toponyme:Territoire>* restreintes pour ne pas détecter les gentilés et enfin le cinquième chemin détecte les gentilés qui débutent par une majuscule. Les contextes gauches sont encore une fois utilisés pour remédier aux restrictions apportées par les patron de début et fin d'entité.

Le prochain transducteur, *tagSigle*, reconnaît les sigles. Il s'agit d'une version légèrement modifiée du graphe de CasEN. Les modifications consistent essentiellement à restreindre les contextes d'apparition du patron qui avait tendance à produire des erreurs dans sa version originale (reconnaissance d'entités déjà balisées faisant intervenir des sigles telles que les chaînes de télévision). Le transducteur se base sur le patron de reconnaissance de sigle de CasEn que l'on peut voir dans l'illustration ci-dessous. Nous pouvons remarquer que ce patron possède de fortes restrictions morphologiques, ce qui l'empêche de reconnaître certaines combinaisons de lettres telles que *LOSC* (club de football), je reviendrai sur cette limitation plus tard dans le rapport et comment

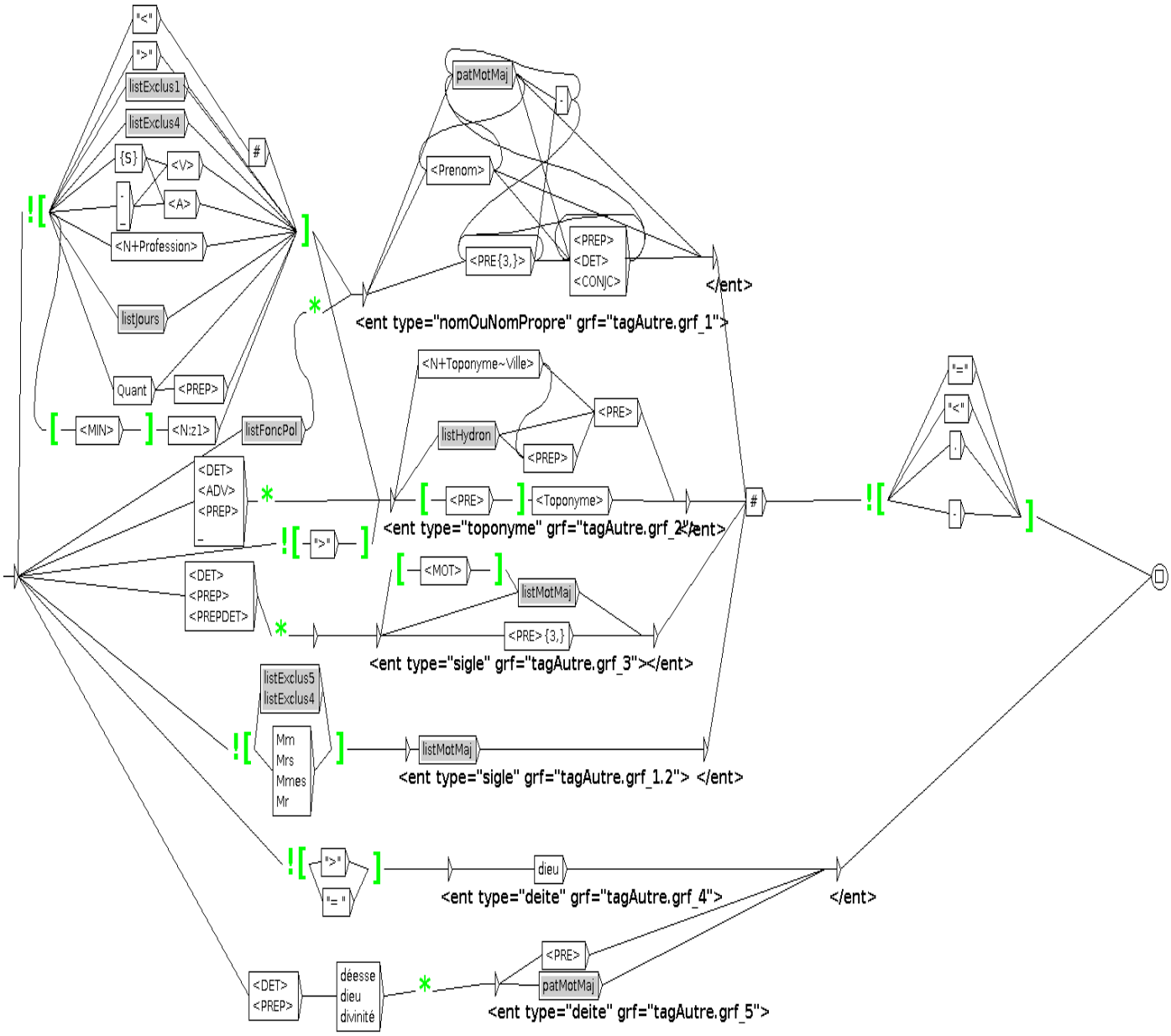
celle-ci peut être contournée au prix d'une baisse de précision de l'outil.



Le prochain transducteur, qui se nomme *tagNomSeul* a pour but de reconnaître les noms ou les prénoms employés seuls. Comme nous pouvons le voir dans l'illustration suivante, il est très simple, c'est pourquoi je ne pense pas qu'il soit nécessaire de détailler son fonctionnement.



Pour finir, la cascade fait appel au transducteur nommé *tagAutre*. Son but principal est d'augmenter le rappel de l'outil au prix d'une baisse de la précision, notamment en levant certaines contraintes (sémantique ou formelles) qui apparaissent dans les transducteurs précédents. Il permet également de détecter les sigles qui serait ignorés par le graphe discuté précédemment en reconnaissant toute suite de mot majuscule. Ce graphe reconnaît également quelques types toponymiques spéciaux tels que les hydronymes faisant apparaître une combinaison de mot majuscules et minuscules (ex : *mer Rouge*) ou bien des combinaisons de mots courants employés comme des noms propres tels que *Banc de la Reine* par exemple. L'idée derrière ce graphe est de « ramasser » tout ce qui aurait été ignorés par les graphes précédents ainsi que de capter l'information contenue dans les parties mal formatées du texte que je mentionnais au début du rapport. Ce graphes me sert également à tester certains chemins destinés à intégrer les transducteurs précédents lorsqu'ils ne produisent plus d'erreurs. Sa forme change donc constamment et il est très enchevêtré comme nous pouvons le voir sur l'illustration suivante. Si l'on souhaite obtenir une précision maximale, il est donc conseillé de désactiver ce graphe lors de l'utilisation de l'outil.



Pour finir, voici un exemple de sortie de la cascade :

Mots clés : football, ligue 1, evian tg, lille Eviant <ent type="sigle" grf="tagSigle.grf_2">TG</ent> <ent type="nomOuNomPropre" grf="tagAutre.grf_1">Lille</ent> : 0_3<ent type="nomOuNomPropre" grf="tagAutre.grf_1">Lille</ent> : <ent type="nomOuPrenom" grf="tagNomSeul.grf">Hazard</ent> (36e s.p), <ent type="nomOuPrenom" grf="tagNomSeul.grf">Payet</ent> (55e), <ent type="nomOuPrenom" grf="tagNomSeul.grf">Pedretti</ent> (67e) Bonne opération pour <ent type="ville" grf="tagTopo.grf_3">Lille</ent>, facile vainqueur d'<ent type="nomOuPrenom" grf="tagNomSeul.grf">Evian Thonon-Gaillard</ent> au terme d'un match pourtant inégal.{S} La première période s'annonçait compliquée pour les deux formations qui avaient surtout à coeur de ne pas se découvrir.{S} <ent type="ville" grf="tagTopo.grf_3">Lille</ent> encaissait le premier coup dur de cette rencontre en perdant <ent type="nomOuPrenom" grf="tagNomSeul.grf">Chedjou</ent> sur blessure (3e).{S} Claqué à la cuisse gauche, le défenseur central laissait sa place à <ent type="nomPropre" grf="tagNomPropre.grf">Mauro

Cetto</ent>, qui en profitait d'ailleurs pour se procurer la toute première occasion de la partie.{S} Seul au point de penalty, l'ancien joueur de <ent type="ville" grf="tagTopo.grf_3">Toulouse</ent> ne cadrait pas sa reprise de la tête sur un coup franc tiré par <ent type="nomOuPrenom" grf="tagNomSeul.grf">Payet</ent> (20e).{S} Invisible jusque-là, <ent type="nomOuPrenom" grf="tagNomSeul.grf">Hazard</ent> sortait peu à peu de sa léthargie.{S} A tel point que sur une chevauchée solitaire, l'international belge réussissait à obtenir un penalty après une intervention en retard de <ent type="ville" grf="tagTopo.grf_3">Cambon</ent>.{S} Le meneur du <ent type="sigle" grf="tagAutre.grf_3.2">LOSC</ent> se faisait justice lui-même pour ouvrir le score et permettre à <ent type="ville" grf="tagTopo.grf_3">Lille</ent> de prendre l'avantage dans une partie pourtant très fermée (0_1, 36e).{S} A la mi-temps, tout était encore jouable pour les locaux.{S} Les choses se précisaient au retour des vestiaires.{S} <ent type="ville" grf="tagTopo.grf_3">Lille</ent> prenait le jeu à son compte et <ent type="nomOuPrenom" grf="tagNomSeul.grf">Payet</ent> doublait la mise sur une belle action collective : De Melo déviait de la tête un long dégagement de <ent type="ville" grf="tagTopo.grf_3">Landreau</ent> pour <ent type="nomOuPrenom" grf="tagNomSeul.grf">Hazard</ent>.{S} Ce dernier, sans contrôle, envoyait <ent type="nomOuPrenom" grf="tagNomSeul.grf">Payet</ent> vers le but.{S} L'ailier reprenait alors l'offrande de volée pour inscrire le deuxième but des siens (0_2, 55e).{S} Assommés, les protégés de <ent type="nomPropre" grf="tagNomPropre.grf">Pablo Correa</ent> en perdaient leur football et péchaient même dans l'envie.{S} <ent type="nomPropre" grf="tagNomPropre.grf">Mickaël Landreau</ent> passait de son côté une soirée plutôt tranquille.{S} La paire <ent type="nomOuPrenom" grf="tagNomSeul.grf">Hazard-Payet</ent> continuait de mettre en difficulté la défense adverse et c'est <ent type="nomOuPrenom" grf="tagNomSeul.grf">Pedretti</ent> qui en profitait pour y aller de son but, une belle frappe à ras de terre de 25 mètres qui franchissait la ligne après avoir touché les deux poteaux d'<ent type="nomOuPrenom" grf="tagNomSeul.grf">Andersen</ent> (0_3, 67e).{S} A 3_0, la messe était dite et <ent type="nomOuPrenom" grf="tagNomSeul.grf">Evian</ent> se résignait à concéder une lourde défaite à domicile.{S} <ent type="nomOuPrenom" grf="tagNomSeul.grf">Angoula</ent> forçait <ent type="ville" grf="tagTopo.grf_3">Landreau</ent> à une parade réflexe en fin de rencontre (87e) mais le score en restait là.{S} <ent type="ville" grf="tagTopo.grf_3">Lille</ent> peut avoir le sourire.{S} Cette victoire permet au <ent type="sigle" grf="tagAutre.grf_3.2">LOSC</ent> de conforter sa 3e place au classement et d'éviter un éventuel retour de <ent type="ville" grf="tagTopo.grf_3">Toulouse</ent> qui joue dimanche face à <ent type="ville" grf="tagTopo.grf_3">Auxerre</ent>.{S} <ent type="nomOuPrenom" grf="tagNomSeul.grf">Evian</ent>, qui n'a toujours pas officialisé sur le plan comptable son maintien, devra afficher un autre visage le week-end prochain sur la pelouse de <ent type="ville" grf="tagTopo.grf_3">Lorient</ent>.{S} Le joueur du matchS'il n'a pas toujours été constant, <ent type="nomPropre" grf="tagNomPropre.grf">Eden Hazard</ent> a tout de même été le grand artisan de la victoire des siens.{S} Fantomatique jusqu'à la 35e minute de jeu, c'est lui qui dans un premier temps part obtenir seul le penalty pour ensuite se faire justice.{S} En deuxième période, ses dribbles ont souvent exaspéré les <ent type="gentile" grf="tagTopo.grf_5">Evianais</ent>, qui ont multiplié les fautes sur lui.{S} L'international belge a ensuite envoyé <ent type="nomOuPrenom" grf="tagNomSeul.grf">Payet</ent> inscrire le but du break avant d'être impliqué sur le troisième but de la rencontre inscrit par <ent type="nomOuPrenom" grf="tagNomSeul.grf">Pedretti</ent>, également très bon sur cette rencontre.{S} On n'a pas aiméIl régnait à <ent type="nomOuPrenom" grf="tagNomSeul.grf">Evian</ent> une ambiance électrique et ce, dès le coup d'envoi.{S} La sortie sur blessure dès la troisième minute d'<ent type="nomPropre" grf="tagNomPropre.grf">Aurélien Chedjou</ent> n'a rien arrangé du côté lillois.{S} Les <ent type="gentile" grf="tagTopo.grf_5">Nordistes</ent> se sont d'ailleurs montré les plus nerveux sur cette rencontre avec 4 cartons jaunes récoltés (<ent type="nomOuPrenom" grf="tagNomSeul.grf">Béria</ent>, <ent type="nomOuPrenom"

grf="tagNomSeul.grf">Hazard</ent>, <ent type="nomOuPrenom" grf="tagNomSeul.grf">Pedretti</ent> et De <ent type="nomOuNomPropre" grf="tagAutre.grf_1">Melo</ent>).{S} <ent type="nomPropre" grf="tagNomPropre.grf">Rudi Garcia</ent> et <ent type="nomPropre" grf="tagNomPropre.grf">Pablo Correa</ent> n'ont cessé d'arguer et de contester depuis leur banc de touche.{S} Résultats de la 29e journée : Samedi 24 mars Ajaccio _ Lorient 1_1 Brest _ <ent type="nomOuNomPropre" grf="tagAutre.grf_1">Nancy</ent> 0_1 Evian _ <ent type="nomOuNomPropre" grf="tagAutre.grf_1">Lille</ent> 0_3 <ent type="nomOuNomPropre" grf="tagAutre.grf_1">Lyon</ent> _ Sochaux 2_1 <ent type="nomOuNomPropre" grf="tagAutre.grf_1">Montpellier</ent> _ Saint-Etienne 1_0 Nice _ Marseille 1_1<ent type="nomOuNomPropre" grf="tagAutre.grf_1">Valenciennes</ent> _ Rennes 1_0 Dimanche 25 mars 17h Dijon _ Caen Toulouse _ Auxerre 21hParis <ent type="sigle" grf="tagSigle.grf_2">SG</ent> Bordeaux La 29e journée en images : La puissance financière du Top 14 effraie toute la planète ovale.{S} Cela a d'abord été les <ent type="gentile" grf="tagTopo.grf_5">Néo-Zélandais</ent>, par la voix à l'époque de <ent type="nomPropre" grf="tagNomPropre.grf">Graham Henry</ent>, qui ont rué dans les brancards.{S}..Prenez quelques journalistes tennis-addicts comme <ent type="nomPropre" grf="tagNomPropre.grf">Christophe Thoreau</ent> et <ent type="nomPropre" grf="tagNomPropre.grf">Julien Pichené</ent>, ajoutez une consultante ex-joueuse <ent type="nomPropre" grf="tagNomPropre.grf">Emilie Loit</ent>, saupoudrez de quelques intervenants de poids.{S}.. Alors que <ent type="ville" grf="tagTopo.grf_3">Sanya</ent> avait pris la tête de la flotte au petit matin ce jeudi, le bateau <ent type="nomOuNomPropre" grf="tagAutre.grf_1">Chinois</ent> a perdu un safran et a dû abandonner le leadership.{S} Une place de leader que se dispute le bateau néo-zélandais Camper et <ent type="nomOuPrenom" grf="tagNomSeul.grf">Groupama</ent>...
{S}VIDEO. Le Français <ent type="nomPropre" grf="tagNomPropre.grf">Romain Grosjean</ent> revient sur un week-end australien riche en sensations, de sa superbe performance en qualifications à son manque de réussite...
<http://www.lefigaro.fr/football-ligue-1_et-2/2012/03/24/02013_20120324<ent type="sigle" grf="tagAutre.grf_3.2">ARTSPO</ent>00405_evian-tg-lille-en-direct.php>