Extraction de résultats expérimentaux d'articles scientifiques pour le peuplement d'une base de données

Anne-Lyse Minard, Anne-Laure Ligozat [1], Brigitte Grau [1]

LIMSI-CNRS - BP 133 – 91403 Orsay Cedex – France

Résumé

QKDB (Quantitative Kidney DataBase) est une base de données relationnelle créée pour centraliser les résultats d'expérimentation sur le rein parus dans des articles scientifiques. Chaque résultat est caractérisé par différents attributs. Actuellement cette base de données est alimentée manuellement par des experts du domaine, et vérifiée par un curateur. Nous présentons dans cet article une solution pour extraire automatiquement les connaissances désirées des articles qui procède en deux étapes : annotation automatique des documents et proposition de celles-ci pour validation ou modification via une interface.

Abstract

The Quantitative Kidney DataBase (QKDB) is a relational database that was created in order to centralize kidney-related experimental results. Each result is characterized by different attributes and the scientific paper from which it was extracted. Currently, this database is populated by hand by experts of the domain. We present a solution to extract automatically the desired knowledge from papers that consists in two steps: automatic annotation of a paper and validation process by a curator by mean of an interface.

Keywords : template information extraction from text, database populating, intelligent assistant tool, kidney experimental results

1. Introduction

La croissance de la littérature scientifique est telle que sa maîtrise par les chercheurs qui ont à l'explorer pour retrouver les informations qui leur sont utiles en est impossible. La biologie n'échappe pas à cet état de fait, et l'extraction d'information purement manuelle afin de peupler des bases de données ou des ontologies pour centraliser ou uniformiser des informations demande beaucoup trop de temps tout en étant source d'erreurs et d'omissions. L'alimentation de la base de données QKDB [2] (Quantitative Kidney DataBase) en est un exemple typique. Ce projet vise à rendre disponible pour la communauté scientifique des données physiologiques relatives au rein. Ces données concernent les résultats d'expérimentations qui relèvent de la physiologie rénale quantitative, et constituent des données pertinentes pour permettre l'évaluation de paramètres de modèles mathématiques de la fonction rénale. Ces données doivent aussi permettre la validation expérimentale et clinique des résultats de simulations.

Dans ce contexte, il est nécessaire de concevoir des outils s'appuyant sur des techniques de traitement automatique de la langue (TAL) pour faciliter l'accès aux articles scientifiques et

[1] Et ENSIIE.

[2] http://physiome.ibisc.fr/qkdb.

permettre d'en extraire des informations, créant ainsi des assistants intelligents (Gaizauskas et al., 2003 ; Corney et al., 2004 ; Alex et al., 2008 ; Karamanis et al., 2006).

Notre projet consiste à fournir un outil interactif pour aider au peuplement de la base QKDB, qui s'appuie sur un processus avancé d'extraction d'informations, visualise les informations trouvées et propose à l'expert de les valider ou les modifier.

Les principales caractéristiques de notre tâche sont les suivantes :

- les traits descriptifs des résultats d'expérimentations, au nombre de 12, correspondent pour certains à des entités nommées (numériques ou symboles par exemple pour des noms de molécules), et pour d'autres à des termes du domaine. Se pose alors le problème de leur variabilité en langue et de leur identification sous toutes leurs formes ;
- les articles mentionnent plusieurs résultats, ce qui augmente les possibilités d'ambiguïtés, et le problème est d'associer les bons traits descriptifs à chacun des résultats ;
- les informations à extraire sont réparties sur l'ensemble des articles, et non sur de courts passages tels que les résumés, ce qui pose plus de problèmes pour les retrouver toutes.

A chaque ajout dans la base, un curateur, expert du domaine, doit vérifier et valider les données entrées. En particulier, il s'agit pour lui de reconnaître les types de descripteurs sous leurs différentes formes textuelles afin de ne pas admettre de valeurs redondantes. Les synonymes et acronymes doivent être normalisés et représentés par la valeur prototypique choisie par les concepteurs de la base.

Notre tâche d'extraction d'information peut être définie comme une tâche consistant à remplir un schéma prédéfini (« template »), telle qu'elle a été définie lors des conférences MUC (Humphreys et al., 1998). Mais, dans MUC les valeurs à trouver pour remplir les schémas correspondent pour la plupart à de la détection d'entités nommées, telles que PERSONNE, ORGANISATION, LIEU, DATE, dont la reconnaissance s'appuie sur des critères de surface (mots déclencheurs, présence de majuscule, etc.) et de listes de termes. Certains des systèmes développés lors de ces évaluations ont été adaptés au domaine biologique (Demetriou and Gaizauskas, 2002) et ont requis le développement de connaissances du domaine sous forme d'ontologies ou de lexiques de spécialités. De telles ressources dédiées à la physiologie rénale n'existent pas, ou seulement très partiellement, et nous verrons comment nous avons procédé pour compléter le lexique existant construit à partir des termes de la base de données.

Aussi, après avoir présenté l'état de l'art section 2, nous présenterons la base de données QKDB en détail (section 3) et décrirons ensuite les différents procédés que nous avons mis en œuvre pour reconnaître les résultats d'expérimentation avec leurs descripteurs dans les articles (section 4), en insistant sur la reconnaissance des termes du domaine et de leurs variations (section 5). Ensuite, nous présenterons les résultats obtenus ainsi que leur évaluation (section 6) et enfin, section 7, nous présenterons l'assistant que nous avons développé qui permet la conversion d'un fichier HTML au format XML que nous avons défini, applique le processus d'extraction d'information pour proposer les informations à extraire et permet de visualiser, modifier et valider ces informations avant leur insertion en base de données.

2. État de l'art

Notre objectif étant de proposer un assistant intelligent aux experts du domaine, nous avons conçu le processus d'extraction d'information comme un processus d'annotation qui propose des valeurs pour remplir les différents champs de la base de données et permet de les modifier

tout en gardant une relation au texte : les annotations ajoutées à l'article traité maintiennent une correspondance avec les tuples de la base de données à insérer, permettant ainsi de naviguer dans le texte ou dans les tuples proposés et de toujours visualiser en parallèle le passage où les annotations sont surlignées et le récapitulatif des champs de la base, permettant ainsi de simplifier le travail de l'expert en accélérant le processus d'analyse du texte et de validation/ sélection des données pertinentes. Alex et al. (2008) ont montré que de tels outils diminuent le temps de curation, et Karamanis et al. (2006), quant à eux, ont menés des études approfondies de l'efficacité apportée par de tels assistants.

L'extraction d'informations structurées relève de méthodes fondées sur le remplissage de schémas ou de méthodes se rapprochant du résumé automatique. Ainsi Ling et al. (2005) posent leur problème d'extraction de différents types d'informations à propos de gènes comme la production d'un résumé où chaque descripteur est rempli par des phrases sélectionnées dans différents textes, sans qu'ils sélectionnent l'information précise.

Les premiers systèmes en extraction d'information ont été développés lors des conférences MUC. Lors de sa dernière édition MUC-7, (voir Chinchor, 1998) pour la définition des tâches de MUC-7), une tâche (Template relation task) était dédiée à l'extraction de relations avec une organisation (« employee_of, manufacture_of, and location_of) pour remplir un schéma alors qu'une autre (Scenario Template Task) était construite autour d'un événement pré-spécifié dans lequel étaient impliqués des organisations, personnes ou artefacts particuliers. Ces deux tâches mettent en jeu les différents problèmes à résoudre : sélection de l'information pertinente, reconnaissance de termes et d'entités nommées et reconnaissance de relations entre ceux-ci.

L'un des systèmes issu de MUC, LaSIE (Humphreys et al., 1998), a été adapté à l'extraction d'informations biologiques, donnant le système PASTA (Demetriou and Gaizauskas, 2002) qui est assez emblématique. PASTA a pour but l'extraction d'informations sur le rôle des résidus présents dans les molécules de protéines. La tâche consiste à remplir un schéma défini par trois entités et deux relations à partir des résumés de MEDLINE [3]. PASTA met en œuvre des processus syntaxiques et sémantiques s'appuyant sur un modèle du domaine représenté par une hiérarchie de concepts.

La plupart des systèmes ne considèrent que les résumés des articles, et seuls quelques uns travaillent sur le texte entier. L'une des raisons en est la difficulté à convertir des articles au format TEXT. Ainsi BioRAT (Corney et al., 2004) et Pharmspresso (Garten and Altman, 2009) ont été conçus pour extraire des informations d'articles au format PDF convertis au format TEXT. Les deux systèmes reposent sur l'écriture de patrons d'extractions sous forme d'expressions régulières mettant en relation des termes liés à l'expression et l'interaction de protéines, qui proviennent de listes, avec des noms de protéines. L'information extraite est toujours localisée dans une phrase.

Ainsi, même si ces systèmes recherchent des informations dans l'ensemble du texte, ils extraient celles-ci de courts passages. Comme les résultats d'expérimentation que nous recherchons sont principalement dans le corps des articles (et généralement absents du résumé), avec des descriptions pouvant être réparties sur plusieurs sections, nous devons travailler au niveau du texte, en tenant compte de sa structure logique.

[3] http://www.ncbi.nlm.nih.gov/sites/entrez?db=pubmed.

3. Base de données

3.1. Description générale

QKDB contient environ 300 articles scientifiques relatifs au domaine de la physiologie rénale, provenant de plusieurs revues, telles que l'American Journal of Physiology – Renal Physiology ou le Journal of Clinical Investigation. De ces articles ont été extraits plus de 8000 résultats expérimentaux par des experts. Chaque résultat expérimental est stocké dans la base de données : ce résultat est représenté par une valeur numérique accompagnée de son unité et de sa précision, le nombre d'animaux sur lequel l'expérience a été menée, des données qualitatives et des commentaires, ainsi que plusieurs champs décrivant le résultat, comme l'espèce ou l'organe concernés.

3.2. Articles

Chaque résultat expérimental est relié à l'identifiant de l'article qui le contient. Ces articles sont au format PDF. Afin de pouvoir les analyser, il était nécessaire de convertir ces articles dans un format textuel, comme XML. La conversion de certains articles de PDF vers XML posant problème (difficulté d'extraire les tableaux et certains caractères spéciaux comme ±), les versions en ligne (au format XHTML) des articles récents ont été collectées, ce qui représente une vingtaine d'articles et environ 900 résultats expérimentaux. Ces documents ont été convertis dans un format XML qui comprend les balises suivantes : titre, auteurs, corps de l'article, paragraphes, tableaux (avec lignes et colonnes) et notes de bas de page.

3.3. Schéma relationnel

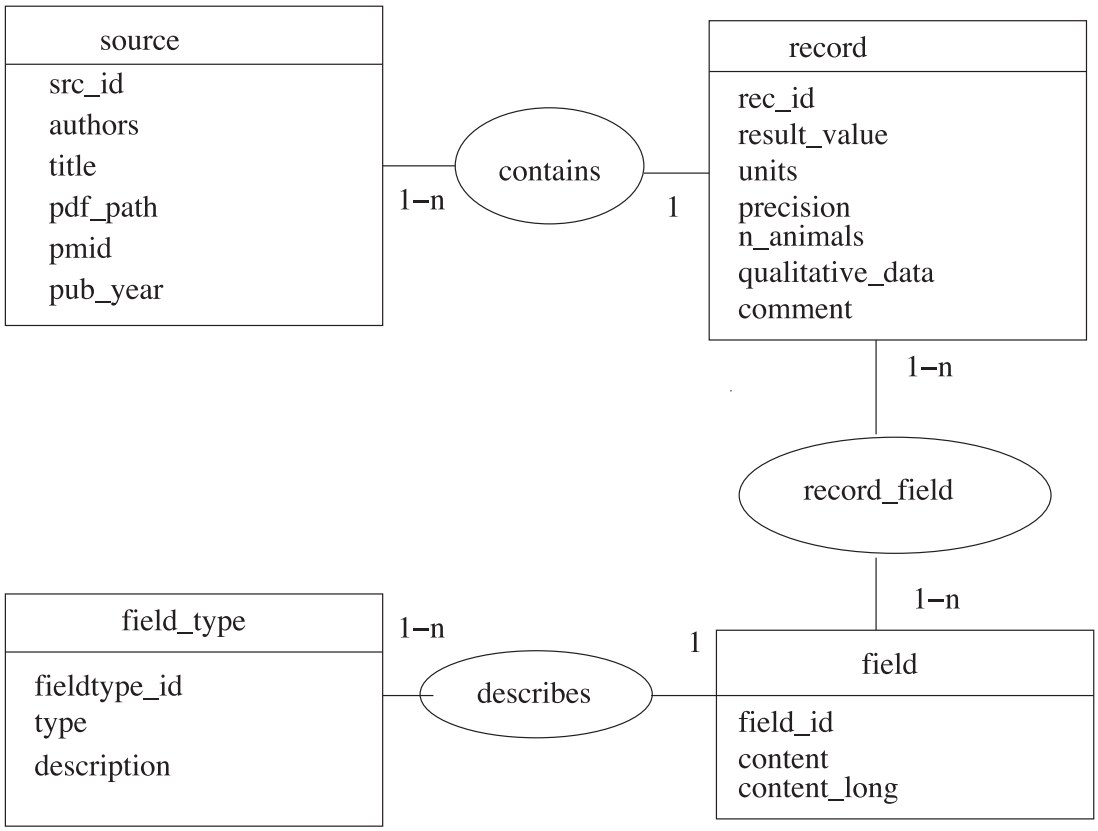

Figure 1 : Schéma relationnel de QKDB

Concrètement, quatre relations principales permettent de représenter les informations QKDB (voir Fig. 1) :

- la relation *source* décrit l'article contenant le résultat, c'est-à-dire ses auteurs, son titre, son année de publication...
- chaque résultat est stocké dans une table *record*, qui contient sa valeur numérique, l'unité, la précision, les conditions expérimentales...
- les autres informations qui décrivent le résultat et qui ont des valeurs prédéfinies sont stockées dans une table *field* : l'espèce (par exemple « mouse »), l'organe...
- chacune de ces informations est reliée à un *field_type*, qui donne son type : « mouse » sera par exemple reliée à un *field_type* 1, qui correspond aux espèces.

3.4. Résultats expérimentaux

Les résultats expérimentaux stockés dans la base de données se composent de plusieurs éléments qui seront retrouvés dans l'article :

- la valeur numérique du résultat ;
- l'unité du résultat, qui qualifie la valeur numérique ;
- une précision, qui indique généralement l'erreur standard de la mesure ;
- des données qualitatives, qui décrivent qualitativement le résultat ;
- un commentaire, qui donne des informations complémentaires par exemple sur les traitements appliqués aux animaux.

Toutes ces informations sont des attributs de la table *record* de QKDB ; ils n'ont pas de valeur prédéfinie. Les autres informations décrivant un résultat expérimental ont en revanche des valeurs prédéfinies dans la base (correspondant aux éléments de la table *field_type*). Ces informations sont les suivantes :

- l'espèce sur laquelle l'expérience a été menée ;
- l'organe, la région, le segment, le compartiment et éventuellement le type de cellule, qui représentent les endroits sur lesquels l'expérience a été menée ;
- le paramètre, qui indique la propriété qui a été mesurée, comme le poids, la perméabilité, le diamètre ou la concentration ;
- le soluté, qui précise ce qui a été mesuré, par exemple $HCO3^-$ si la concentration mesurée concerne ce soluté.

Toutes ces informations caractérisent un résultat expérimental. Certaines peuvent ne pas être indiquées pour certains résultats ; seule la valeur numérique est toujours présente.

3.5. Exemple de résultat

Voici un exemple de phrase contenant un résultat expérimental stocké dans QKDB : « serum osmolality increased to 517 mOsm compared with 311-325 mOsm in wild-type and heterozygous mice. »

Les informations qui sont stockées dans QKDB concernant le premier résultat sont les suivantes : valeur numérique : « 517 », unité : « mOsm », données qualitatives : « serum osmolality in KO mice », commentaire : « AQP1 knockout mice generated by targeted gene disruption; after water deprivation for 36h. », espèce : « mouse », organe : « kidney », région : « arterial plasma », paramètre : « osmolality ».

On peut remarquer que certaines informations sont données dans la phrase du résultat, comme le paramètre, mais d'autres doivent être inférées du reste de l'article, comme l'organe concerné. On peut également noter que certaines informations sont présentes, mais sous une forme différente de celle stockée dans la base de données : par exemple l'espèce est présente sous sa forme fléchie « mice », tandis que la région « arterial plasma » est exprimée par une variante sémantique « serum ».

Dans la perspective d'une annotation automatique de ces informations dans les articles, il est essentiel de tenir compte de ces variations. En particulier, pour les champs de la base ayant des valeurs prédéfinies, une seule forme est stockée pour chaque type de concept de chaque champ : celle-ci correspond au terme protypique choisi par les concepteurs de la base pour ce concept, et l'ensemble de ces valeurs constitue un lexique minimal de termes typés. Ainsi, si l'on considère les régions, la région « cortex » peut être exprimée sous cette forme, ou sous les formes « cortices », « pallium » ou « pallia ». Ainsi, un même concept d'un même champ de la base peut apparaître sous différentes dénominations dans les articles.

Nous allons d'abord présenter les principes du système d'extraction d'information que nous avons développé, puis nous montrerons comment les variations de ces champs ont été recherchées puis intégrées à ce système.

4. Annotation des résultats expérimentaux

Une étude de corpus nous a permis de mettre en évidence les caractéristiques de notre corpus, telle que la localisation des résultats (dans les parties « Results » et « Discussion »), et des descripteurs (phrase du résultat, et/ou phrase précédente et/ou paragraphe), et proposer un processus d'extraction adapté (Grau et al., 2009).

4.1. Valeurs numériques

Les articles à annoter sont tout d'abord analysés par le TreeTagger (Schmid, 1994), qui effectue une analyse morphosyntaxique des textes. Puis, les valeurs numériques des résultats sont recherchées grâce à trois patrons d'extraction, l'un recherchant un nombre suivi d'une indication de précision et d'un éventuel exposant, un autre effectuant la même recherche mais dans les tableaux, et un dernier recherchant un nombre suivi d'une unité. Les résultats sont recherchés dans les sections « Results » et « Discussion » des articles.

Ces valeurs numériques sont assez bien repérées puisque leur extraction (calculée sur un sous-corpus contenant environ 800 résultats constitué à partir des données entrées dans la base) se fait avec un rappel de 1 et une précision de 0,64. La précision est un peu basse car la totalité des résultats numériques pertinents dans les articles analysés n'a pas été insérée dans la base.

4.2. Autres champs

Pour chaque valeur numérique trouvée, ses descripteurs sont recherchés. Le script d'extraction de ces champs utilise deux types de ressources : des patrons et des listes de termes associés à leur type. Deux stratégies différentes sont adoptées :

- pour les champs n'ayant pas de valeur prédéfinie dans la base comme la précision ou le nombre d'animaux, des patrons d'extraction sont utilisés. Ainsi, le nombre d'animaux étudiés est soit précédé de « n= », soit suivi du nom de l'espèce. Ces patrons sont écrits sous la forme d'expressions régulières : le nombre d'animaux peut être reconnu par le patron suivant :

(\d+|$nombre) ($nom_espèce|males|females)
qui signifie que ce nombre doit être écrit sous forme décimale ou de nombre en toutes lettres, suivi d'un nom d'espèce ou des termes « males » ou « females ». Ces patrons sont facilement modifiables car regroupés dans des fichiers extérieurs au script d'extraction ;

- pour les champs ayant des valeurs prédéfinies, les termes correspondant aux valeurs de ces champs dans la base sont recherchés et celui qui est le plus proche de la valeur numérique du résultat est sélectionné, dans un contexte constitué soit de la phrase contenant la valeur, soit de cette phrase et la précédente, soit de son paragraphe. Si l'espèce n'est pas trouvée dans ce contexte, l'espèce la plus fréquente dans l'article est choisie, et un choix similaire est effectué pour l'organe. Pour repérer les termes de la base, ceux-ci sont regroupés dans des lexiques dont les entrées ont la forme suivante :
  PAR_109 thickness
  « PAR » donne le type de l'entrée, en l'occurrence « paramètre », « 109 » désigne l'identifiant du concept dans la base, et « thickness » est la forme de ce concept présente dans la base.

Le script d'annotation annote les champs trouvés par leur type et l'identifiant du résultat qu'ils décrivent. Une première application de ce procédé, utilisant les lexiques constitués à partir des valeurs contenues dans les champs de la base nous a permis d'établir une baseline (cf. section 6) et d'évaluer le taux de silence induit par cette forme minimale des lexiques (Grau et al., 2009). Nous avons ensuite cherché à améliorer la reconnaissance des termes du domaine.

5. Ajout de vocabulaire

5.1. Utilisation de ressources externes pour la complétion des lexiques

Afin d'améliorer l'extraction, nous avons tout d'abord cherché des ressources permettant de reconnaître de nouvelles formes des termes de la base, afin de compléter les lexiques. Celles-ci sont généralement sous forme de listes de termes structurées (thésaurus ou ontologie), ou de listes provenant de sites Internet comme Wikipédia.

La base CELEX a été utilisée pour ajouter des variations morphologiques à nos lexiques.

Puis d'autres ressources nous ont permis de compléter les lexiques avec de nouveaux termes ou des variantes des termes de la baDans le domaine médical, le métathésaurus UMLS (Unified Medical Language System) regroupe différents thésaurus en créant des relations entre les différents concepts. Il est développé par la NLM (National Library of Medicine). Celui-ci est très riche mais également très complexe. Nous avons donc choisi de n'utiliser qu'un des thésaurus de l'UMLS : FMA (Foundational Model of Anatomy), thesaurus (ou ontologie) qui décrit l'anatomie du corps humain. Il contient 75 000 classes, et plus de 120 000 termes reliés par 168 types de relations.

Pour l'extraction des unités nous avons utilisé l'ontologie units.obo, au format .obo, de Gene Ontology, et qui nous a permis de collecter une liste d'unités de base et d'unités composées.

Enfin, pour certains champs, nous avons complété les lexiques en utilisant des listes disponibles sur des sites spécialisés (tels que www.kterre.org/dossiers/atomes_liste.php ou les pages Wikipedia comme en.wikipedia.org/wiki/Dictionary of chemical formulas) contenant par exemple des listes de molécules avec leurs formules chimiques (1680 formules ont ainsi été collectées mais seules celles qui sont associées aux termes de la base ont été rajoutées).

5.2. Extraction de candidats termes

Une méthode pour construire des lexiques consiste à extraire des candidats termes des textes pour les donner à valider ensuite. Potentiellement tous les noms situés dans le contexte des résultats numériques, contexte pouvant correspondre à une phrase, sont candidats. Ils constituent de fait un trop grand nombre de mots qui, de plus, seraient difficiles à typer. Nous nous sommes donc intéressées à la modélisation des relations qui existent entre deux paramètres, leur reconnaissance dans les textes devant permettre de sélectionner uniquement des noms qui sont des candidats potentiels de descripteurs et de les typer.

Nous avons généré des patrons par apprentissage automatique pour représenter les relations à partir de notre corpus de référence. Pour cela nous avons collecté les formulations contenues entre deux paramètres définis. Par exemple entre le champ Parameter et le résultat numérique nous avons la séquence « PAR of the anesthetized ESP was RES » où PAR est un terme du champ Parameter, ESP, du champ Species et RES un résultat numérique.

Les patrons créés sont des patrons linguistiques multi-niveaux, c'est-à-dire des patrons contenant à la fois des lemmes et des catégories syntaxiques. En utilisant l'algorithme de (Pantel et al. 2004), nous recherchons l'alignement optimal des séquences deux à deux, dans le but de généraliser les patrons. L'algorithme est composé de deux parties, la première consiste à calculer le nombre minimum d'opérations d'édition (suppression, insertion et remplacement) permettant de passer d'une phrase à l'autre. La deuxième partie produit l'alignement optimal, suivant quatre principes appliqués successivement :

- si les lemmes sont équivalents, le lemme est conservé dans le patron,
- si les catégories syntaxiques sont équivalentes elles sont conservées,
- si les deux éléments sont différents, ils sont replacés par *g*,
- s'il y a eu une insertion, *s* est inséré.

Par exemple nous obtenons le patron suivant : PAR of the *g* ESP *s* RES à partir des phrases « PAR of the -/- ESP (RES » et « PAR of the anesthetized ESP was RES ».

s indique qu'il peut y avoir une instance de n'importe quel mot et *g* qu'il doit y avoir une instance de n'importe quel mot. Pour améliorer la pertinence des patrons, nous avons choisi de supprimer tous les patrons contenant plus de deux *g* ou *s*, comme dans Embareck (2008).

Nous avons ainsi extrait des patrons pour les relations : Parameter – Résultat, Solute – Résultat et Region – Résultat, nous en avons obtenu respectivement 39, 18 et 16. Les autres couples de paramètres ont été étudiés mais ne sont pas utilisés. La relation de certains couples n'est pas pertinente, par exemple la relation entre l'espèce et le soluté. Pour d'autres couples, le corpus ne contient pas assez d'exemples pour que des patrons puissent être appris, par exemple pour le couple « Structure type » et « Region ». Et enfin la relation entretenue entre deux descripteurs est parfois de type « complément du nom » (ils font tous les deux partis du même syntagme nominal), la relation n'est alors pas assez restrictive, et l'apprentissage de patron inutile. C'est le cas par exemple entre les termes du champ Parameter et du champ Solute (ex : « **Na+ excretion** was significantly increased in [...] »).

Les patrons ont été appliqués à un corpus de 20 articles (173 phrases contenant un résultat numérique) qui ne sont pas dans la base QKDB. Les termes extraits sont ensuite validés manuellement. Si nous étudions ces termes, nous observons de nouvelles variantes de termes de la base. Tab. 1 donne le nombre d'occurrences de termes typés par les patrons (plusieurs patrons peuvent être appliqués à la même phrase), la proportion de ceux qui sont corrects et

la proportion des termes corrects qui sont nouveaux, puis le nombre de termes différents et corrects ainsi que le pourcentage de nouveaux termes parmi ces derniers.

	Parameter	*Solute*	*Region*
Termes étiquetés	1111	464	291
Termes corrects	196 (= 17%)	33 (= 7%)	24 (= 8%)
Termes nouveaux/ termes corrects	156 (= 80%)	23 (= 70%)	17 (= 70%)
Nombre de termes différents corrects	15	16	7
Pourcentage de nouveaux termes	66 %	37 %	57 %

Tableau 1 : Nombre de termes typés par les patrons dans le corpus

Nous avons rajouté les nouveaux termes au lexique. Après l'enrichissement du lexique, nos lexiques pour les champs fixes sont constitués de plus de 860 termes (*Epithelial compartment or membrane* : 34, *Cell type* : 26, *Structure type* : 10, *Species* : 175, *Solute* : 124, *Parameter* : 304, *Tube segment* : 86, *Organ, tissue or cell line* : 24 et *Region* : 84).

5.3. Reconnaissance des variantes

Afin de détecter les variantes de termes, nous avons utilisé Fastr (Jacquemin, 1996), un analyseur de surface qui permet de détecter des variantes morphologiques, syntaxiques ou sémantiques de termes. Fastr fonctionne à partir de métarègles, telles que :

```
Metarule Coordination(N1 -> A2 N3) = N1 -> A2 C4 A5 N3: .
Metarule NameToVerb(N1 -> N2 PREP3 N4)=
    X1 -> V2 <ADV? [PREP? DET| PREP] A?> N4:
    <V2 root> = <N2 root> .
```

avec *A* pour adjectif, *N* nom, *C* coordination, *V* verbe, *Adv* adverbe, *Prep* préposition, *Det* déterminant. L'expression qui suit la première flèche décrit le modèle du terme sur lequel est appliquée la variation décrite après le signe =.

La première règle autorise l'insertion d'un adjectif coordonné entre l'adjectif et le nom du modèle. La seconde règle représente une variation morphosyntaxique, autorisant l'emploi d'un verbe de la même famille morphologique que le nom du modèle (<V2 root> = <N2 root>) qui accepte N4 en complément. Ce type de règle permet la reconnaissance de la variante « to reduce rapidly the noise » pour le terme « reduction of noise ».

Etant donné un texte, et une liste de termes constituée de mono-termes et de multi-termes, ces données sont analysées par le TreeTagger, puis Fastr compile ses métarègles de manière à les instancier sur les termes donnés, et finalement applique les règles instanciées sur le texte lemmatisé qui est annoté par les termes et leurs variations. De manière à pouvoir typer les termes reconnus, nous avons constitué un lexique par champ de la base.

6. Résultats et évaluation du processus d'extraction de schémas

Le système d'extraction d'information a été évalué sur le corpus de référence (cf. section 3.2). Ce corpus de référence a été semi-automatiquement annoté et corrigé ensuite et nous permet d'évaluer le processus d'extraction (cf. résultats Tab. 2). La précision correspond au nombre de descripteurs correctement annotés divisé par le nombre de descripteurs annotés. Un descripteur est considéré comme correct si il est du bon type et qu'il est relié au résultat qu'il décrit. Le rappel correspond au nombre de descripteurs correctement annotés divisé par le nombre total

de descripteurs à annoter (ceux qui sont dans QKDB). Nous ne présentons que les résultats concernant les valeurs d'expérimentation figurant dans la base. Les autres valeurs peuvent être des résultats pertinents que l'utilisateur n'a pas insérés, ou bien des valeurs qu'il a jugées inintéressantes ou encore des erreurs de reconnaissance, les moins nombreuses (un résultat qui ne correspond pas à une expérimentation).

La baseline a été produite en appliquant les règles de reconnaissance décrites section 4 et des lexiques constitués des valeurs provenant de la base de données. Nous avons évalué ensuite le processus d'extraction en augmentant manuellement les listes (LM) et en appliquant Fastr (LM+Fastr) puis en rajoutant les termes (Patrons) validés à partir de leur collecte automatique à partir des textes (LM+Patrons+Fastr). Les apports en vocabulaire ont nettement amélioré la baseline, que ce soit pour le rappel ou la précision. La découverte de nouveaux termes a permis une amélioration du rappel. Pour les termes dont l'extraction est fondée sur les listes (57%), notre système en reconnaît 50% avec LM+Fastr et 60% après l'apport des patrons.

	baseline	*LM + Fastr*	*LM + patrons + Fastr*
Rappel	0.45	0.63	0.68
Précision	0.52	0.74	0.73

Tableau 2 :Résultats du processus d'extraction

Fastr permet de ramener des variantes de différents types et son utilisation a nettement augmenté le nombre de termes reconnus. Nous avons effectué une étude à la suite du système LM + Fastr afin d'illustrer les différents types de variations trouvées. Globalement, 3000 occurrences des 194 multi-termes ont été trouvées, avec 520 flexions (16,7%), 377 variantes syntaxiques (10,8%) et 84 variantes morpho-syntaxiques (2,7%). Les variations syntaxiques correspondent essentiellement à des coordinations et des insertions de mots. Par exemple, pour le champ de type « tube segment », « glomerular cells and capillaries » est reconnu pour « glomerular capillary » ainsi que « distal convoluted tubule » pour « distal tubule ». Pour le champ « region », « outer and inner stripe » est reconnu pour « outer stripe ». L'exemple suivant montre l'application d'une règle de type « NameToVerb », avec la reconnaissance de « fraction of filtered » pour « filtration fraction ».

Les termes qui ne sont pas reconnus sont des termes présents dans des sections de l'article que nous ne traitons pas (un petit nombre), des termes dont la valeur n'est pas explicite dans le texte mais qui pourraient être déduit d'autres valeurs et des termes inconnus. Par ailleurs, la base de données est parfois incomplète, et donc notre référence aussi, et des descripteurs corrects proposés par le système sont parfois comptabilisés comme faux.

7. Assistant intelligent pour peupler la base QKDB

Les différents processus utiles à l'analyse d'un texte, allant de sa conversion au format XML requis jusqu'à la proposition de tuples à intégrer dans la base de données, sont intégrés au travers d'une interface. Après la phase d'extraction, l'outil propose la visualisation présentée in Fig. 2.

L'utilisateur peut :

- parcourir l'article en allant d'un résultat à l'autre ;
- visualiser, pour chaque résultat, les informations qui lui sont associées qui sont surlignées en couleur, avec un code couleur par type de champ ;

- avoir un récapitulatif des attributs caractérisant une expérimentation ; celui-ci est affiché dans une infobulle quand on passe au dessus d'un descripteur ;
- visualiser les attributs de la base de données dans un formulaire modifiable affiché en bas d'écran, qui correspond au résultat affiché dans la partie texte ;
- parcourir les schémas extraits, avec un affichage de la partie texte correspondante ;
- modifier des champs via le formulaire ; les modifications sont répercutées sur l'annotation du texte de manière à conserver la cohérence entre les deux points de vue sur le texte.

L'assistant peut remplir une double fonction et peut aussi être vu comme un outil d'aide à l'annotation d'articles, permettant de créer des corpus pour l'apprentissage ou l'évaluation.

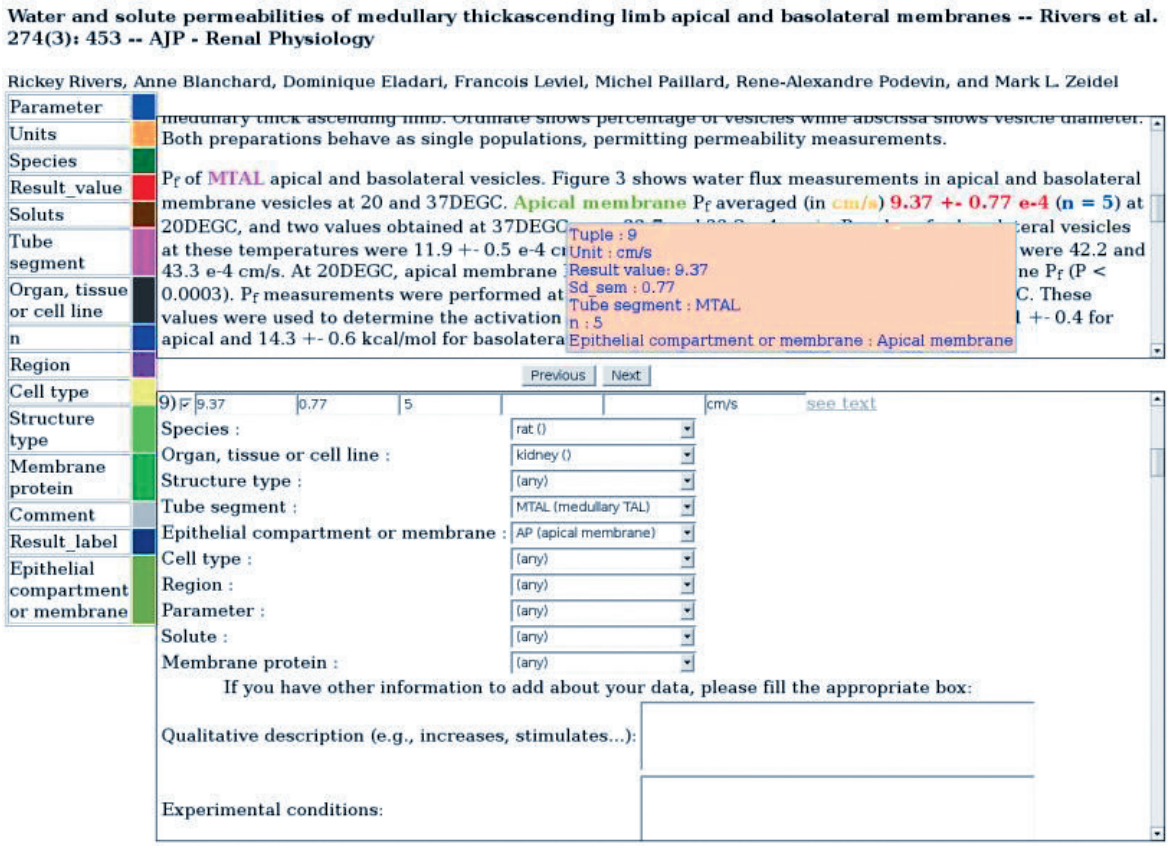

Figure 2 : Interface de l'assistant de peuplement de la base de données

8. Conclusion

Une première version du système d'extraction d'information afin de remplir un schéma de base de données a été créée à partir des informations fournies par QKDB afin d'établir une baseline. Celle-ci a montré le manque de la couverture linguistique, aussi avons-nous complété les lexiques par ajout manuel de termes trouvés dans des ontologies ou des listes disponibles sur Internet, mais aussi par des termes découverts grâce à un apprentissage de patrons de relations entre couples de types de termes.

Le système a été intégré dans un assistant intelligent permettant à un expert de remplir la base de données sans avoir à analyser complètement les articles. Il reste à évaluer cet assistant et à augmenter la taille de nos corpus d'apprentissage, mais cela nécessite une annotation qui ne peut être réalisée, ou validée, que par un expert. Notre outil peut être utilisé dans ce cadre.

Références

Alex B., Grover C., Haddow B., Kabadjov M., Klein E., Matthews M., Roebuck S., Tobin R. and Wang X. (2008). Assisted Curation: Does Text Mining Really Help? In *Proceedings the Pacific Symposium on Biocomputing*.

Chinchor N. (1998). Overview of MUC-7. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*.

Corney D.P.A., Buxton B.F., Langdon W.B. and Jones D.T. (2004). BioRAT: extracting biological information from full-length papers. *Bioinformatics*, 20(17) : 3206.

Demetriou G. and Gaizauskas R. (2002). Utilizing text mining results: The PastaWeb system. In *Proceedings of the Workshop on Natural Language Processing in the Biomedical Domain*, pp. 77-84.

Gaizauskas R., Demetriou G., Artymiuk PJ. and Willett P. (2003). Protein structures and information extraction from biological texts: the PASTA system. *Bioinformatics*, 19(1) : 135-143.

Garten Y. and Altman R. (2009). Pharmspresso: a text mining tool for extraction of pharmacoge- nomic concepts and relationships from full text. *BMC bioinformatics*, 10 (Suppl 2) : S6.

Grau B., Ligozat A-L. and Minard A-L. (2009). Corpus study of kidney-related experimental data in scientific papers. In *Proceedings of the Biomedical Information Extraction Workshop, International Conference RANLP (Recent Advances in Natural Language Processing)*.

Embarek M. (2008). *Un système de question-réponse dans le domaine médical – Le système Esculape*. Thèse de l'université Paris-Est.

Humphreys K., Gaizauskas R., Azzam S., Huyck C., Mitchell B., Cunningham H. and Wilks Y. (1998). University of Sheeld: Description of the LaSIE-II system as used for MUC-7. In *Proceedings of the Seventh Message Understanding Conferences (MUC-7)*.

Jacquemin C. (1996). A symbolic and surgical acquisition of terms through variation. In Wermter, S., Riloff, E. and Scheler G., editors, *Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing*, Heidelberg: Springer, pp. 425-438.

Karamanis N., Lewin I., Seal R., Drysdale R. and Briscoe E. (2006). Integrating Natural Language Processing with Flybase Curation. In *Proceedings of the Pacific Symposium on Biocomputing*.

Ling X., Jiang J., He X., Mei Q., Zhai C. and Schatz B. (2005). Automatically Generating Gene Summaries from Biomedical Literature. In *Proceedings of the Pacific Symposium on Biocomputing*.

Medhi E. (2008). Un système de question-réponse dans le domaine médical – Le système Esculape, Thèse de l'université Paris-Est.

Pantel P., Ravichandran D. and Hovy E. (2004). Towards terascale knowledge acquisition. In *International Conference on Computational Linguistics (COLING'04)*, Geneva, pp. 771-777.

Schmid H. (1994). Probabilistic Part-of-Speech Tagging Using Decision Trees. In *International Conference on New Methods in Language Processing, Manchester*, Royaume-Uni.