Logiciel d'aide à l'étiquetage morpho-syntaxique de textes de spécialité

Ahmed Amrani*, Jérôme Azé**, Yves Kodratoff**
*ESIEA Recherche, 9 rue Vésale, 75005 Paris, France
amrani@esiea.fr
** LRI, Université Paris Sud, 91405 Orsay Cedex, France
{aze,yk}@lri.fr, http://www.lri.fr/~{aze,yk}

Résumé. La compréhension de textes de spécialité nécessite un étiquetage morpho-syntaxique de bonne qualité. Or, lorsque les textes étudiés sont issus de domaines spécifiques et peu usités, il est rare de disposer de dictionnaires et autres ressources lexicales fiables. Le logiciel que nous proposons permet d'utiliser un étiquetage réalisé par un étiqueteur généraliste, puis d'améliorer cet étiquetage en intégrant des connaissances d'experts du domaine étudié. Grâce au logiciel développé, il est relativement aisé pour un expert du domaine de détecter des erreurs d'étiquetage et de mettre en place des règles de ré-étiquetage. Ces règles peuvent être obtenues de deux manières différentes : (1) soit en utilisant un langage de programmation permettant d'exprimer des règles complexes de ré-étiquetage, (2) soit par apprentissage automatique des règles à partir d'exemples corrigés au moyen d'une interface dédiée. Cet apprentissage propose de nouvelles règles à l'expert, acquises automatiquement.

1 Introduction

La compréhension de textes de spécialité repose sur un étiquetage morpho-syntaxique de bonne qualité. Or, lorsque les textes étudiés sont issus de domaines spécifiques et peu usités, il est rare de disposer de dictionnaires et autres ressources lexicales fiables. Ainsi, les systèmes d'étiquetage (Brill, 1994; Schmid, 1994) ne sont pas en mesure d'étiqueter correctement des textes spécialisés. Ayant réalisé ce constat et face au besoin d'avoir des textes correctement étiquetés pour pouvoir en extraire des connaissances utiles, il devient indispensable de corriger l'étiquetage. De nombreux outils peuvent être utilisés pour modifier et corriger l'étiquetage d'un texte.

L'étiqueteur de Brill (Brill, 1994) offre la possibilité d'écrire des règles contextuelles qui seront utilisées pour modifier l'étiquetage réalisé par défaut. Cependant, les règles ainsi exprimées ne sont pas utilisables en dehors de l'étiqueteur de Brill.

INTEX [1], bien que non conçu pour cette tâche, pourrait être utilisé pour détecter des erreurs d'étiquetage et pour les corriger. L'utilisation d'INTEX implique de disposer de dictionnaires dédiés au domaine pour obtenir un premier étiquetage relativement correct. Or, comme nous l'avons précédemment évoqué, il est difficile d'obtenir de telles ressources.

Des outils d'analyse syntaxique profonde des textes, tels qu'INTEX, sont certes nettement plus fiables qu'une simple analyse syntaxique de surface. Par contre, le temps de

1. http://www.nyu.edu/pages/linguistics/intex/

calcul requis pour réaliser une telle analyse est prohibitif pour traiter de gros volumes de textes techniques. Dans l'approche que nous proposons, les règles de correction de l'étiquetage peuvent être appliquées très rapidement à tout nouveau texte.

Le système que nous présentons, ETIQ[2] (Amrani et al., 2004), est un système convivial et inductif permettant d'améliorer l'étiquetage morpho-syntaxique des corpus de spécialité. Ce système est composé de deux modules : le module lexical et le module contextuel. La partie lexicale permet d'écrire des règles fondées sur des critères morphologiques tels que : les suffixes, préfixes et le mot lui-même. Le module contextuel permet d'écrire des règles qui corrigent l'étiquette du mot en fonction de son contexte dans la phrase, c'est-à-dire le mot lui-même, son étiquette, les mots voisins et leurs étiquettes. Actuellement, l'étiqueteur utilisé en entrée d'ETIQ est celui de Brill. Notre approche est divisée en deux phases : (1) application des règles lexicales de Brill puis des règles lexicales spécialisées de l'expert, (2) application des règles contextuelles de Brill puis des règles contextuelles spécialisées de l'expert. Après l'application des phases précédentes, nous avons remarqué que de nombreuses erreurs persistent. Ces erreurs sont de plusieurs natures : le lexique utilisé n'est pas adapté à la spécialité, effets de bords des règles lexicales de l'expert, règles contextuelles imparfaites, *etc.*

L'outil que nous proposons aide l'expert à détecter ces erreurs et lui facilite l'écriture des règles de correction. En l'état actuel du logiciel, la tâche de détection des erreurs d'étiquetage est dévolue à l'expert (le système permet simplement de faciliter cette détection). Selon la difficulté et l'importance des erreurs détectées, le logiciel permet d'enrichir la base de règles de deux manières différentes : écriture manuelle de règles (via une interface dédiée) et induction de règles à partir d'exemples annotés.

2 Écriture manuelle de règles

Étant donné un mot dont l'étiquette est incorrecte, l'expert du domaine peut utiliser ETIQ pour exprimer des règles contextuelles simples de ré-étiquetage.

Ces règles sont définies graphiquement via l'interface qui permet à l'utilisateur de visualiser le contexte proche du mot à ré-étiqueter. Un contexte de zéro à trois mots autour du mot mal étiqueté peut ainsi être utilisé.

Il existe de nombreuses situations dans lesquelles la grammaire de règles proposée par ETIQ n'est pas suffisante pour exprimer la correction à réaliser (contexte trop réduit, contrainte portant sur le début (ou la fin) de la phrase, *etc.*).

Pour résoudre ce problème et offrir plus de souplesse à l'expert du domaine, nous avons conçu, dans le cadre de la compétition TREC 2004[3] (Soboroff et Harman, 2003), un langage dédié au ré-étiquetage et permettant d'exprimer simplement des connaissances du domaine.

Le langage développé offre la possibilité à l'utilisateur d'exprimer ses règles contextuelles sous forme de **conditions** qui doivent être vérifiées et d'**actions** associées. Les règles peuvent admettre des **exceptions** (par exemple, tous les *être* sont des formes modales sauf s'ils sont précédés d'un article). La forme générale d'une règle est la suivante : **si** *conditions* **alors** *actions* **sauf** *exceptions*.

2. http://www.lri.fr/ia/Genomics/
3. TREC : **T**ext **RE**trieval **C**onference, http://trec.nist.gov/

Les conditions, actions et exceptions s'expriment généralement sous la forme de triplet : **(Pos, Mot, Etiquette)** où **Pos** est la position relative du mot dans la phrase, **Mot** est le mot situé à la position indiquée et **Etiquette** est l'étiquette du mot (par ex. une étiquette de Brill [4].

Les positions s'expriment relativement à un élément central que nous nommons le **pivot**. Le pivot est l'élément autour duquel la règle va s'articuler et il s'agit très souvent du mot qui doit être ré-étiqueté. Le pivot doit **obligatoirement** être présent dans la partie *conditions* de la règle et s'exprime de la manière suivante : (**0**, Mot, Etiquette). Les informations Mot et Etiquette peuvent ne pas être toutes les deux renseignées.

Par exemple, la règle **si** *(-1,,RB) (0,,NN) (+1,,JJ)* **alors** *(-1,,JJ)* **sauf** *(-2,,JJ)* exprime le fait que si l'élément central est étiqueté comme un nom *(0,,NN)* et qu'il est précédé d'un adverbe *(-1,,RB)* et suivi d'un adjectif *(+1,,JJ)* alors l'adverbe est ré-étiqueté en adjectif sauf s'il est lui même précédé d'un adjectif.

Ce type de règle peut s'écrire très facilement avec le logiciel Etiq. Par contre, le langage offre la possibilité d'écrire des règles plus complexes en permettant à l'expert de manipuler des éléments dont la position peut être inconnue lors de l'écriture de la règle mais qui seront instanciés lors de l'application de celle-ci.

Le langage dispose aussi d'une bibliothèque de fonctions intégrées qui permettent à l'expert d'exprimer des contraintes sur les mots, les étiquettes, les positions et la phrase. Il est ainsi possible de rechercher les mots en début de phrase, ou contenant une séquence déterminée de caractères et de vérifier leurs étiquettes.

3 Annotation des exemples et induction automatique des règles de correction

Pour certain type d'erreurs complexes, il devient très difficile de trouver une règle de correction générale qui prend en considération toutes les exceptions possibles. Dans ce cas, nous utilisons des algorithmes d'apprentissage de règles.

La méthodologie utilisée consiste à permettre à l'expert d'annoter facilement les exemples. À partir de ces exemples, nous apprenons automatiquement des règles de correction. Ces règles sont mises au format d'Etiq et insérées dans la liste des règles contextuelles.

Les sections suivantes présentent le principe de l'induction implantée dans Etiq.

3.1 La sélection des exemples

Après la détection des erreurs dans leur contexte, l'outil permet à l'expert de sélectionner les exemples concernés. Cette sélection peut se faire selon plusieurs critères : le mot lui-même, sa morphologie, son étiquette, les mots voisins, leurs morphologies et leurs étiquettes (voir Figure 1). Tous ces critères peuvent être combinés par des opérateurs logiques. Par exemple : le mot « *that* » est généralement mal étiqueté. Dans ce cas, nous sélectionnons des exemples où le mot central est *that*. Parmi les mots

4. Quelques étiquettes : (NN,Nom commun singulier), (JJ, adjectif), (RB, adverbe), (VBN, verbe au participe passé), (DT, déterminant)

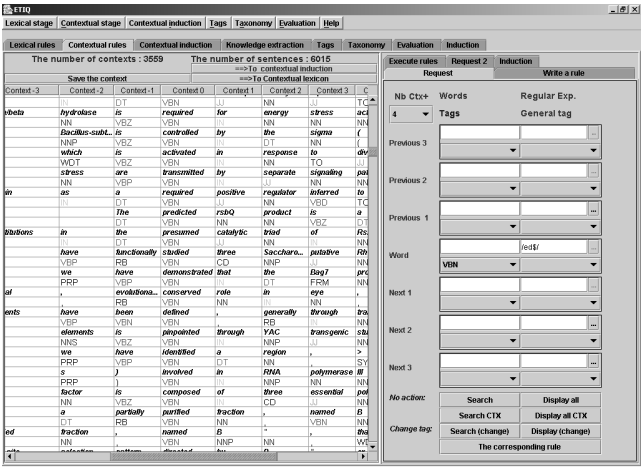

FIG. 1 – ETIQ : *La sélection des exemples à annoter. À gauche, la liste des exemples où le mot central a le suffixe « ed » et l'étiquette « VBN ». À droite, l'interface de sélection.*

étiquetés VBN et se terminant par le suffixe « ed », il y a une bonne proportion de prémodifieurs (étiquetés JJ) ou de prétérites (étiquetés VBD). Pour traiter ce cas, nous sélectionnons les exemples ayant l'étiquette VBN et se terminant par le suffixe « ed ».

3.2 Induction de règles

Les exemples sélectionnés doivent être annotés par l'expert (voir Figure 2). Les exemples sont ordonnés en fonction de leur similarité morphologique et de leur étiquette morpho-syntaxique. Ainsi, les exemples susceptibles d'avoir la même étiquette seront voisins. L'expert peut alors sélectionner un exemple (ou un groupe d'exemples) et lui associer l'étiquette correcte. L'exemple corrigé est alors transféré dans l'ensemble des exemples annotés.

Les exemples annotés permettent d'engendrer la base de données utilisée pour l'apprentissage des règles de correction. Avant d'engendrer la base, l'expert peut choisir la taille du contexte à prendre en considération, le type des attributs et l'utilisation ou non d'ontologies. Deux types d'ontologies construites manuellement par un expert sont actuellement disponibles : une ontologie des étiquettes et une ontologie des mots.

La forme générale d'une ontologie de n éléments est : <Nom-Groupe>élément$_1$, élément$_2$,..., élément$_n$</Nom-Groupe>. Par exemple : l'ontologie générale d'étiquettes « nom » prend la forme suivante : <Nom>NN,NNS,NNP,NNPS</Nom>. L'utilisation de cette ontologie permet d'exprimer l'étiquette générale *Nom* dans une règle au lieu d'utiliser plusieurs règles avec NN, NNS, NNP et NNPS. Par exemple : Si l'étiquette du mot suivant appartient à l'ontologie des étiquettes *Nom* et l'étiquette du mot précédent

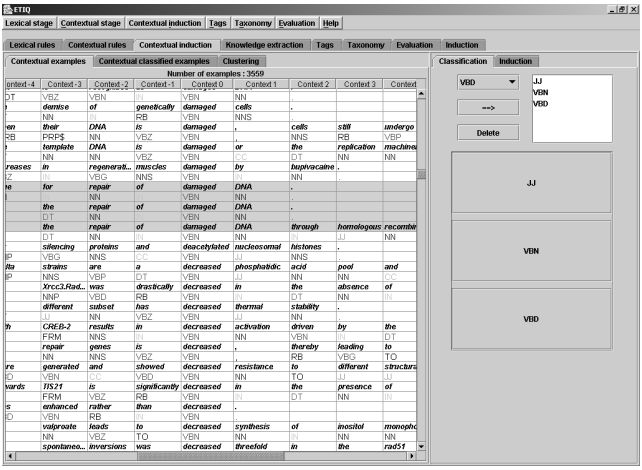

FIG. 2 – ETIQ : *Annotation des exemples : À gauche, trois exemples similaires où le mot central est 'damaged' sont sélectionnés. Dans ce cas, l'expert corrige l'étiquette de « VBN » à « JJ » en cliquant sur le bouton JJ (à droite).*

est DT alors l'étiquette du mot central est JJ.

À partir de la base de données engendrée, nous utilisons la collection d'algorithmes d'apprentissage de WEKA [5] (Witten et Frank, 2000).

Cet environnement nous permet de comparer plusieurs algorithmes d'apprentissage pour notre tâche. Nous avons intégré deux algorithmes propositionnel d'apprentissage de règles PART (Eibe et Witten, 1998) et RIPPER (Cohen, 1995). Les règles résultantes ont la forme d'une conjonction de conditions. Notons T_1 et T_2 et ... T_n le **corps de la règle** et C_x la **classe cible à apprendre**.

Une règle s'exprime donc de la manière suivante : si T_1 et T_2 et ... T_n alors la classe est C_x. Chaque condition T_i teste une valeur particulière d'un attribut, et elle prend la forme suivante : $A_n = v$, où A_n est un attribut nominal et v est une valeur possible de A_n. Les règles obtenues sont transformées au format ETIQ (ci-dessus), et insérées automatiquement à la suite de la liste des règles contextuelles.

4 Conclusions

Le logiciel présenté permet à un spécialiste du domaine étudié de détecter et de corriger facilement de nombreuses fautes d'étiquetage. Les corrections peuvent être réalisées soit en écrivant des règles de correction, soit en utilisant celles apprises par le système à partir de quelques corrections réalisées par l'expert.

5. http://www.cs.waikato.ac.nz/∼ml/weka/. L'archive jar contenant WEKA a été intégrée dans ETIQ.

Dans la version courante du langage intégré dans Etiq, le ré-étiquetage est effectué phrase à phrase. L'expert ne peut donc utiliser que les informations contenues dans la phrase courante pour exprimer ses règles de ré-étiquetage. La possibilité d'accéder aux phrases voisines, au paragraphe contenant la phrase ou aux textes du même contexte social devrait offrir encore plus de souplesse à l'expert dans l'écriture de ses règles.

L'induction de règles réalisées par Etiq se limite pour l'instant aux contextes simples centrés sur le mot à ré-étiqueter. Certaines règles de ré-étiquetage ne peuvent pas être apprises avec ces simples contextes. L'extension de l'induction aux méthodes d'apprentissage de la programmation logique inductive permettra d'étendre la famille de règles pouvant être apprises.

Références

Amrani, A., Kodratoff, Y., et Matte-Tailliez, O. (2004). A semi-automatic system for tagging specialized corpora. Dans *Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'04)*, volume 3056, pages 670–681.

Brill, E. (1994). Some advances in transformation-based part of speech tagging. Dans *AAAI*, volume 1, pages 722–727.

Cohen, W. W. (1995). Fast effective rule induction. Dans Prieditis, A. et Russell, S., editors, *Proc. of the 12th International Conference on Machine Learning*, pages 115–123, Tahoe City, CA. Morgan Kaufmann.

Eibe, F. et Witten, I. H. (1998). Generating accurate rule sets without global optimization. Dans Shavlik, J., editor, *Machine Learning: Proceedings of the 15th International Conference*. Morgan Kaufmann Publishers, San Francisco, CA.

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. Dans *Proceedings of the International Conference on New Methods in Language Processing*, pages 44–49.

Soboroff, I. et Harman, D. (2003). Overview of the TREC 2003 novelty track.

Witten, I. H. et Frank, E. (2000). *Data Mining: Practical machine learning tools with Java implementations*. Morgan Kaufmann, San Francisco.

Summary

Understanding texts of specialty relies on a good morpho-syntactic tagging. When these texts belong to a very specialized domain, dictionaries and other reliable lexical resources are seldom available. The tagging obtained from general taggers thus needs to be improved. The software we describe here uses a general tagger, and improves step-by-step the tagging, integrating more and more domain knowledge in the process. This software is friendly in that sense that a field expert can easily detect tagging errors and write him/herself rules in order to modify the tagging by using a programming language devoted to this task. The semantic of this language has been adapted to the flow of the sentences to be tagged. The tagging rules can be obtained in two different ways: (1) using our programming language (2) rule learning from examples. This learning proceeds by analyzing the new tags provided by the field expert.