La réécriture de graphes au service de l'annotation de corpus et de l'exploitation de corpus annotés

Dans l'annotation linguistique de corpus, il existe de multiple formats d'annotation liés d'une part aux différents niveaux linguistiques existants et d'autre part aux différents choix linguistiques pour un même niveau. Il est utile de disposer d'outils automatiques qui permettent de transformer un corpus annoté à un niveau linguistique donné en un corpus annoté à un niveau voisin, mais il est également utile, à un même niveau linguistique, de pouvoir convertir d'un format à un autre. La construction de tels outils est facilitée par le fait d'inscrire les différents types d'annotation dans un cadre formel commun ; nous pensons que le cadre mathématique des graphes est approprié pour cela. En effet, lorsque l'on travaille au niveau de la syntaxe, les structures manipulées sont des arbres, mais il est souvent utile de considérer un niveau voisin, par exemple le niveau phonologique avec l'ordre des mots. Les graphes permettent de représenter ces deux dimensions (dépendances syntaxiques et relations d'ordre) dans la même structure. Plusieurs théories linguistiques [SHP86, Mel88] conçoivent un niveau intermédiaire (appelé syntaxe profonde) entre la syntaxe (rebaptisée syntaxe de surface) et la sémantique. Dans les structures syntaxiques profondes, on représente des relations qui ne sont pas explicites en surface ; ces nouvelles relations induisent naturellement une structure de graphe. Enfin, les structures sémantiques s'apparentent naturellement à des graphes.

Dans ce cadre commun, les transformations d'annotations, que ce soit d'un niveau linguistique à un autre ou à un même niveau, sont des transformations de graphes. La *réécriture de graphes* est donc un formalisme naturel pour réaliser cette tâche. Un système de réécriture de graphes est un ensemble de règles de transformations élémentaires locales qui se prêtent bien à la modélisation de phénomènes linguistiques. Une règle est formée d'une partie gauche (qui décrit le motif à rechercher dans le graphe) et d'une partie droite (qui décrit comment modifier le graphe). La difficulté est de préciser comment le résultat de l'application de la règle va être connecté au contexte. Il n'y a pas mathématiquement de façon standard de faire, nous avons donc conçu un modèle de la réécriture de graphes spécifiquement adapté au TAL. Nous avons implanté ce modèle sous forme d'une application, que nous appellerons ici RG, qui prend entrée un système de règles de réécriture écrites et un corpus annoté et retourne le corpus avec l'annotation transformée par le système de règles. En pratique, de nombreuses applications de règle sont nécessaires pour une transformation et il est possible de décrire des stratégies d'application pour contrôler l'enchaînement des règles. Nous avons écrit huit systèmes de règles pour des corpus français annotés en syntaxe de surface, en syntaxe profonde, en sémantique et même en parties du discours car l'un des systèmes permet de faire de l'analyse syntaxique. Les transformations peuvent poser problème à automatiser quand le format de sortie est plus riche que le format d'entrée ; dans ce cas, une partie des problèmes peut être résolue par appel à des lexiques pour aller y chercher l'information absente de l'annotation d'entrée.

On peut aussi utiliser RG pour faire de la recherche de motifs dans un graphe en ne considérant que la partie gauche des règles. C'est utile pour explorer des corpus à des fins linguistiques mais cela permet également d'aider la construction et la maintenance des corpus annotés : avec la recherche de motifs, on peut détecter des erreurs ou des incohérences dans l'annotation. La syntaxe des motifs de RG permet de représenter des graphes (y compris avec des cycles) tout en ordonnant partiellement les sommets des graphes. Les motifs comportent deux parties : une partie positive qui décrit le motif recherché et une partie négative qui filtre les résultats. En pratique, on commence souvent avec un motif relativement général et on ajoute progressivement de nouveau filtres pour raffiner la recherche. Cette utilisation de RG est accessible directement en ligne sur internet. La figure 1 montre un exemple de recherche de dépendances non projectives dans un corpus annoté en syntaxe de dépendances.

```
pattern {
    N1 -> N2; M1 -> M2;
    N1 << M1; M1 << N2; N2 << M2;
}
```

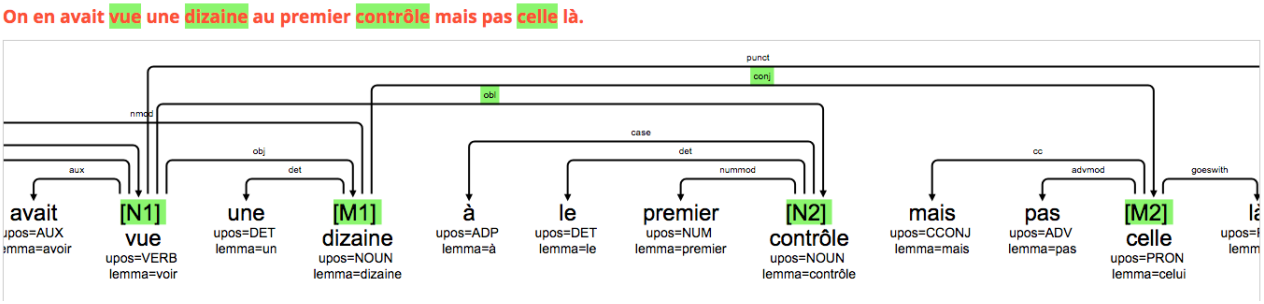

FIGURE 1 : recherche de dépendances non projectives (en haut le motif recherché, en bas un des exemples trouvés avec le texte puis la structure syntaxique)

Références

[Mel88] I. Mel'čuk. *Dependency Syntax : Theory and Practice*. Albany, N.Y. : The SUNY Press, 1988.

[SHP86] Petr Sgall, Eva Hajicová, and Jarmila Panevová. *The meaning of the sentence in its semantic and pragmatic aspects*. Springer Science & Business Media, 1986.