Les premiers pas vers la création d'un extracteur automatique de groupes nominaux à partir de textes étiquetés hongrois
Une problématique de l'intelligence artificielle

Ágoston NAGY

Introduction

La reconnaissance des groupes nominaux (GN) ne constitue pas une problématique à part de la linguistique informatique. Elle s'insère notamment dans l'analyse syntaxique automatique qui regroupe les unités minimales des textes en unités plus larges comme les syntagmes. La reconnaissance des GN est essentielle dans la traduction automatique (Pohl 2005), la reconnaissance des termes (Kis 2005) ou l'indexation automatique des textes (Fournas 1987). Parmi ces procédés, l'indexation a une importance primordiale pour les moteurs de recherche sur Internet, étant donné que la plupart des recherches se font par des GN et non pas, par exemple, par une conjonction. Il est donc crucial d'indexer tout document web par les GN qu'il comporte en vue d'accélérer les recherches.

Dans cet article, nous nous proposons de démontrer quelles sont les problématiques rencontrées lors d'une extraction de GN et de présenter les premières étapes de l'élaboration d'un code source dont le but final est d'extraire des GN à partir de textes hongrois étiquetés. Nous commencerons par la description de notre corpus d'analyse qui servira de point de repère pour le code source (écrit en Java). Cette partie est suivie de la présentation des méthodes les plus éminentes d'extraction de GN (Section 2) ; il s'agit surtout de méthodes hybrides combinant en même temps méthodes statistiques et linguistiques. La Section 3 présente la façon dont nous cherchons à extraire des patrons de GN qui doivent être ensuite filtrés, comme nous le montre la Section 4. La Section 5 se concentre sur les tâches à effectuer par la suite.

1. Le corpus

Le corpus de notre analyse est constitué par Szeged Treebank 2.0, développé sous la direction du Département de linguistique informatique de l'Université de Szeged. Ce corpus contient des textes de divers genres (romans, essais courts rédigés par des lycéens, articles de journaux, textes juridiques, etc.). Le corpus contient 82.000 phrases, soit 1,2 millions de mots (*Szeged Treebank 2.0*). La totalité des textes sont annotés et lemmatisés, ce qui signifie que pour chaque mot du texte, le radical et la catégorie grammaticale correspondants sont marqués. Une partie des textes, dont celui que nous avons choisi d'analyser, ont également subi une analyse syntaxique automatique (appelé aussi *parsing*), c'est-à-dire les mots y sont regroupés en unités plus larges (syntagmes) pour que les arbres syntaxiques associés à chacune des

phrases puissent être établis par des logiciels de visualisation. Dans les textes similaires (sous format XML) sont marqués aussi les GN, ce qui est un point de repère inestimable pour notre recherche.

Comme texte à analyser, nous avons choisi la traduction hongroise du roman *1984* de Orwell, donc une oeuvre littéraire. Le corpus a été divisé en deux parties : un corpus d'apprentissage et un corpus de test dont les taux respectifs sont de 90% et de 10%. Le premier sert de modèle ou de patron pour le code source où ce dernier cherche des régularités avec lesquelles il pourrait reconnaître les GN dans n'importe quel texte étiqueté. La recherche des régularités ne se fait pas, bien sûr, complètement automatiquement, il faut apprendre à l'application (comme aux non-initiés) les méthodes et les décisions qu'elle peut ou doit appliquer quand il s'agit de déterminer si tel ou tel syntagme est vraiment un GN ou non. Dans le corpus de test, tout renseignement complémentaire, comme les indications de syntagmes, a été supprimé. La raison pour cet effacement est que les indications de GN y seront insérées par le code source.

La version électronique de *1984* se compose de 6658 phrases (soit 79350 mots) qui contiennent 22405 GN. Le corpus d'apprentissage (les 10% de ce dernier) contient 2226 GN, ce qui correspond parfaitement aux 10% du corpus entier même si la division a été établie sur la base du nombre des mots et non sur celle du nombre des GN.

2. Tour d'horizon des méthodes de détection de GN

Dans cette sous-section, nous nous proposons de décrire les extracteurs de GN les plus importants, dans lesquels nous puisons aussi pour créer notre propre application de cette sorte. La plupart de ces systèmes fonctionnent sur la base d'automates à états finis mais il existe également des applications qui se reposent sur des méthodes purement statistiques.

La reconnaissance des GN fait partie d'une opération informatique et linguistique qui s'appelle analyse syntaxique automatique (*parsing*). L'analyse syntaxique automatique consiste au regroupement des unités minimales en unités plus larges comme le syntagme.

Muñoz et al. (1999), ainsi que Ramshaw et Marcus (1995) utilisent des méthodes purement statistiques et ne prennent guère en considération la structure interne des GN. Muñoz et al. (1999) se concentrent sur les deux frontières de GN : O (ouverture) et C (clôture). En fonction des données statistiques, le logiciel marque entre chaque deux mots un O, un C ou rien, selon qu'il y a ou non une frontière de GN entre ces deux mots. Ramshaw et Marcus (1995) se concentrent sur les éléments susceptibles d'apparaître dans un GN, et considèrent cette tâche comme celle de l'étiquetage pour chaque mot de texte (comme s'il s'agissait d'un étiquetage de chaque mot en tant que nom, adjectif, etc.). Ils utilisent deux étiquettes : I (à l'intérieur d'un GN) ou O (hors les GN).

La plupart des systèmes combinent des méthodes linguistiques et des méthodes statistiques, comme celui de Araujou et Serrano (2008) ou celui d'Allan

(2009). Chacun de ces auteurs se base sur la méthode des automates à états finis combinés avec les méthodes statistiques. Les automates à états finis sont créés à partir de corpus d'apprentissage mais les patrons ne sont pas filtrés, ceux-ci sont donc tous gardés. Dans cette méthode, la statistique est déjà insérée dans l'automate et en fait un automate pondéré : les transitions d'un état à l'autre sont complétées par un poids qui marque la probabilité de cette transition et si le poids total du patron n'atteint pas un niveau nécessaire, le groupe ne sera pas considéré comme un GN.

Il existe aussi des méthodes qui utilisent des grammaires hors contexte (*context-free grammars*) qui sont plus ou moins équivalentes aux règles de réécritures connues du domaine de la linguistique générative. Ces grammaires sont hors contexte car les règles le sont aussi, comme :

(1) i. S → NP VP
ii. NP → Dét N
iii. Dét → le | la | l'
etc.

Ce sont bien des règles hors-contexte car selon la définition, les règles d'une langue hors contexte s'appliquent au schéma suivant :

> Soit G=(N, T, S, P) une grammaire formelle, où N est un ensemble fini de symboles non-terminaux, T un ensemble fini de symboles terminaux, S (Є N) l'axiome (l'état initial) et P un ensemble des règles de production et soit L une langue reconnue par G. La grammaire G est considérée comme une grammaire hors contexte si toute règle de production de G est le suivant :
> A → w, où A est un non-terminal, w est un mot composé de symboles terminaux et non-terminaux.[1] (Martín-Víde 2008)

En ce qui concerne notre objectif, ce type de grammaire s'avère peu efficace. En effet, nos analyses portent sur l'extraction des GN et non pas sur leur restructuration interne. Pour construire une telle représentation structurelle, il nous faudrait aussi des règles de réécriture pour analyser les autres syntagmes, comme les groupes adjectivaux, car en syntaxe les différents syntagmes s'incluent[2]. Or, rien que pour les GN, il existe plus de huit cents patrons, comme nous allons le voir plus tard.

Pour l'extraction des GN, nous allons également utiliser un automate à états finis élaboré automatiquement à partir du corpus d'apprentissage. L'automate ne sera pas pondéré, comme dans le cas de Araujou et Serrano (2008) ou de Allan (2009), mais il ne sera capable de reconnaître que les GN que nous allons lui apprendre. Les GN seront filtrés sur la base des algorithmes statistiques connus du domaine de l'intelligence artificielle. Nous avons choisi une méthode qui est apte à saisir cette problématique, notamment la méthode naïve de Bayes.

---

[1] Si nous considérons les exemples en (1), nous pouvons constater que dans toutes les règles il n'y a que des non-terminaux (ce ne sont que des étiquettes de grammaire) à gauche tandis qu'à droite, il peut y avoir des non-terminaux (comme en (i) et (ii)), ou des terminaux (comme en 3).

[2] Un GN peut contenir un groupe adjectival, comme dans le cas du GN *le petit enfant*, le GN contient un groupe adjectival aussi, notamment *petit*.

3. Extraction des patrons

Dans une première étape, nous avons opté pour une extraction des patrons des GN, nous avons donc essayé de trouver toutes les structures internes possibles des groupes nominaux à partir des données du corpus d'apprentissage. Tout impossible que ce soit, selon le corpus d'apprentissage, il existe 835 différentes formes de groupes nominaux dont 656 ne figurent qu'une ou deux fois dans le texte. Par ailleurs, il existe 30 patrons qui surgissent plus de 50 fois avec une occurrence totale de 18040, soit 89% du nombre des GN. Ceci montre déjà qu'il sera plus facile de reconnaître les patrons les plus fréquents dans le corpus de test ; ce qui sera plus difficile, c'est de reconnaître les GN dont la structure interne est moins fréquente. Le tableau suivant représente, par ordre de fréquence, les dix patrons les plus fréquents des GN se trouvant dans le corpus :

Patron	Fréquence	exemple
Det N	4034	az órák (les horloges)
N	3431	Winston
Pron	3357	én (je)
Adj	982	látható (visible)[3]
Adj N	920	hatalmas méreteivel (avec sa taille)
Det Adj N	731	a gonosz széltől (du vent méchant)
Adv	621	vele (avec lui)[3]
Det N N	578	az utca szintjén (au niveau de la rue)
Pron N	512	mindegyik emeleten (à chaque étage)
Pron Det N	435	ezt a munkát (ce travail)

Tableau 3.1. Les patrons de GN les plus fréquents dans *1984*.

Ce qui est frappant à première vue, c'est qu'il y a des patrons qui ne contiennent même pas de noms (ou de pronoms qui leur sont équivalents). En fait, ces patrons constituent environ 11,8% du corpus, ce qui a plusieurs raisons : soit l'étiquetage du corpus, soit l'analyse syntaxique ne nous est pas adéquate. En d'autres termes, il peut arriver que le GN a été bien catégorisé en tant que tel mais l'annotateur a catégorisé le nom qui s'y trouve par exemple en tant qu'adverbe. Ou bien, l'analyse syntaxique a considéré comme GN un autre syntagme, mais le taux de cette source d'erreur ne peut pas être mesuré avec des outils informatiques[4].

L'étiquetage (ou plus fréquemment *POS-tagging*) consiste à associer à chaque mot du texte sa catégorie grammaticale, et d'autres propriétés morphologiques comme le genre, la personne, la dérivation, etc. Ce qui est le plus problématique dans l'étiquetage, c'est la désambiguïsation, étant donné que la plupart des mots sont ambigus dans les langues naturelles. Cela se fait sur la base

[3] Il est clair que, dans ces cas, il s'agit d'une analyse syntaxique qui n'est pas conforme aux notations traditionnelles car *látható* (visible) est vraiment un adjectif mais il ne peut pas constituer seul un GN et que *vele* (avec lui) constitue vraiment un GN mais il n'est pas un adverbe mais un pronom.

[4] Pour chacun de ces deux cas, un exemple sera traité plus tard dans cette section.

des règles et, conjointement, par des méthodes statistiques (Voutilainen 2003). Par exemple, à défaut d'autres points de repères, on constate que le mot *tables* est plus fréquent comme nom (au pluriel) que le verbe *tabler* (conjugué à la deuxième personne du singulier), ainsi, s'il se trouve devant une préposition ou un déterminant, il est plus probable qu'il s'agisse d'un nom. Le corpus a été étiqueté par le logiciel HuMor qui fonctionne aussi sur ces méthodes, mais utilise une grammaire d'unification et des automates à états finis (Prószéky & Kis 1999).

Pour y voir clair dans les exemples suivants, nous expliquons brièvement le format XML utilisé par l'étiqueteur HuMor. Pour chaque mot, il peut y avoir plusieurs analyses, comme dans l'exemple de *tables* mentionné ci-dessus, qui sont marquées entre les balises <anav> et </anav>. La catégorie du mot et les informations concernant ses flexions sont marquées entre des crochets, entre les balises <mscat> et </mscat>. L'analyse choisie par le logiciel est marquée entre les balises <ana> et </ana> : ce choix est basé sur une méthode (non publique, mais probablement) relevant du calcul de probabilité.

Pour montrer un exemple concret, où il n'y a même pas d'ambiguïté, considérons un exemple où le mot à étiqueter est *törvényeket* (l'accusatif du mot pluriel *lois*) :

```
<w>törvényeket
<ana>[...]<lemma>törvény</lemma><mscat>[Nc-pa]</mscat> [...]</ana>
<anav>[...]<lemma>törvény</lemma><mscat>[Nc-pa]</mscat>[...]</anav>
</w>
```

À partir de cet exemple, nous pouvons voir que l'étiqueteur a bien considéré comme [Nc-pa] le mot *törvényeket* car il s'agit vraiment d'un nom commun au pluriel au cas accusatif.

Un exemple tiré du corpus du mot *hidegnek* dans la phrase *Odakünn, még a bezárt ablakon keresztül is, hidegnek látszott a világ*[5]. montre la première source d'erreur, notamment l'analyse syntaxique qui ne nous est pas adéquate :

```
<NP id="Ohu.1.2.4.1.6">
<w>hidegnek
<ana> [...] <lemma>hideg</lemma><mscat>[Afp-sd]</mscat>[...] </ana>
<anav> [...] <lemma>hideg</lemma><mscat>[Afp-sd]</mscat>[...]</anav>
<anav> [...]<lemma>hideg</lemma><mscat>[Afp-sg]</mscat>[...]</anav>
<anav>[...]<lemma>hideg</lemma><mscat>[Nc-sg]</mscat></anav>
<anav>[...]<lemma>hideg</lemma><mscat>[Nc-sd]</mscat></anav>
</w>
</NP>
```

On voit ici que l'étiqueteur a associé correctement cette fois-ci la catégorie grammaticale adjectif [Afp-sd] à ce mot mais le balisage en syntagmes l'a marqué comme un groupe nominal (NP). Les quatre analyses possibles de *hidegnek* (marquées entre les balises <anav>) sont donc : adjectif singulier dont le cas est soit

[5] Dehors, le monde m'a semblé froid, même à travers les fenêtres fermées.

datif ou génitif, ou nom commun singulier génitif ou datif. De ces quatre possibilités, l'étiqueteur a choisi le premier, marqué entre les balises <ana>.

Dans d'autres cas, c'est l'étiquetage qui n'est pas adéquat. Un exemple tiré du corpus : *Csendben ült, mint egy egér, abban a hiú reményben, hogy akárki is az, egyszeri próbálkozás után továbbmegy*[6]. Selon l'analyse syntaxique *akárki* est un GN (ce qui est sans doute le cas) mais l'étiquetage l'a marqué, pour une raison ou une autre, comme une conjonction au lieu d'un pronom ([Pg3-sn], pronom à valeur générale de la 3^{e} personne du singulier au cas nominatif), ce qui aurait été le premier choix :

```
<NP id="Ohu.1.2.45.2.11"> [...]
<w>akárki
<ana>[...]<lemma>akárki</lemma><mscat>[Cssp]</mscat>[...]</ana>
<anav>[...]<lemma>akárki</lemma><mscat>[Pg3-sn]</mscat>[...]</anav>
<anav>[...]<lemma>akárki</lemma><mscat>[Cssp]</mscat>[...]</anav>
</w>[...]
</NP>
```

4. Filtrage des patrons

À partir des exemples et des statistiques, nous pouvons conclure que l'extraction des patrons est loin d'être suffisante pour une reconnaissance des GN. En effet, il existe des patrons qui ne représentent que rarement des GN, comme une unité étiquetée comme conjonction au lieu d'un pronom, et il existe des patrons qui surgissent rarement dans des corpus mais qui représentent probablement des GN. Il faut ajouter donc des méthodes d'apprentissage au corpus par lesquelles le logiciel pourrait décider si un patron constitue vraiment un GN ou non. Pour cela, nous pouvons prendre en considération plusieurs facteurs :

- est-ce que le patron relève uniquement des GN ou peut représenter un autre groupe ?
- est-ce que le début du patron correspond vraiment ou non aux GN (les GN commencent en général par un déterminant, un adjectif ou un nom) ?
- est-ce que le contenu du patron nous indique aussi s'il désigne un GN ou non (s'il ne contient ni P ni N, alors il n'est pas probable que ce soit un GN) ?

Comme aucune analyse syntaxique automatique n'est parfaite, il faut prendre des risques : pourquoi extraire des GN qui sont étiquetés comme conjonctions (au lieu d'être identifiés comme pronoms) ? Il vaut mieux ne pas les extraire sinon toutes les conjonctions seront identifiés comme des GN.

Pour cela, il faut un algorithme qui décide pour chaque patron s'il représente un GN ou non. À partir du corpus d'apprentissage, nous pourrons établir un tableau qui contient une valeur de probabilité pour chaque élément qui sera pris en considération lors de l'apprentissage. Ces valeurs doivent être ensuite totalisées pour

[6] Il était assis tranquillement comme une souris, se nourissant du fol espoir que tout le monde s'en ira qui que ce soit.

chaque patron qui contiendra deux valeurs, l'une est la probabilité de l'appartenance à la catégorie GN, l'autre est la probabilité de son appartenance à un autre groupe. Si le premier est plus grand que l'autre, le patron sera inclus dans la liste des patrons de GN. Cette méthode s'inspire de la classification naïve de Bayes de l'intelligence artificielle. (Russel & Norvig 2002)

5. Les dernières étapes

À partir de la liste filtrée, nous pouvons donc automatiquement créer un automate à états finis, ce qui est essentiel pour extraire les GN. Il faut ajouter que cet automate doit être déterministe (pour chaque transition de l'état a à b, le caractère lu par l'automate doit être différent des autres qui lient les deux mêmes états). Les automates non-déterministes demandent des algorithmes récursifs ou de retour sur trace (*backtracking*) qui, eux, exigent du temps d'exécution superflu et une mémoire plus importante.

Il est aussi essentiel de trouver une solution aux problèmes des analyses syntaxiques qui sont apparemment incorrectes et qui ont été mentionnées dans la Section 3. En effet, le nombre des défauts de l'application pourrait être réduit si les erreurs de l'analyse syntaxique pouvaient être retracées.

La dernière étape de cette analyse est constituée par le test de l'application des patrons sur le corpus de test qui constitue 10% du texte d'apprentissage. Ce choix résulte du fait que nous avons besoin d'un corpus d'apprentissage suffisamment exhaustif des GN (et dans la plupart des cas 90% est suffisant).

Conclusion

Cet article a pour but de décrire les difficultés qui sont posées par l'extraction automatique des GN à base de patrons et de méthodes statistiques, ainsi que d'esquisser les étapes principales de la création d'un tel extracteur :

1. trouver un corpus de repère et le diviser en deux : en corpus d'apprentissage et corpus de test
2. extraction des patrons de GN du corpus d'apprentissage
3. filtrage de ces patrons par des méthodes statistiques et des méthodes de décisions
4. création de l'automate à états finis à partir de la liste filtrée
5. test de l'application

Nous avions également l'intention de démontrer quelles sont les difficultés principales de la création d'un tel système. Les problèmes se manifestent dès le deuxième stade, notamment dans l'extraction des patrons. Les sources d'erreur sont d'ordres divers : l'étiquetage du texte original peut déjà contenir de fausses étiquettes, ou encore c'est l'analyse syntaxique qui est erronée dans certains cas.

Les étapes à réaliser peuvent également réserver des surprises qui se manifesteront pendant la validation du code source.

Source

SzegedTreebank 2.0. (Version cédérom), Université de Szeged, Département de linguistique informatique, s.d.

ORWELL, G., *1984*, Budapest, Európa Könyvkiadó, 1989. (version XML de Szeged Treebank 2.0.)

Références

ALLAN, J., *Automata for language processing*, Lecture notes of the course Speech recognition and statistical language models, consulté le 20 janvier 2009, www.cs.rochester.edu/u/james/CSC248/Lec7.pdf

FURNAS, G.W., LANDAUER, T.K., GOMEZ, L.M., DUMAIS, S.T., The vocabulary problem in human-system communication, *Communications of the ACM 30(11)*, 1987, p. 964-971.

KIS, B., Automatikus terminológia keresés számítógéppel – kísérlet. *Fordítástudomány 7 (1)*, 2005, p. 84-96.

RAMSHAW, L.A., MARCUS, M.P., Text chunking using transformation-based learning, in: *Proceedings of the third ACL workshop on very large corpora*, Cambridge (MA), Association for computational linguistics, 1995, p. 82-94.

ARAUJO, L. J., SERRANO, I., Highly accurate error-driven method for noun phrase detection, *Pattern Recognition Letters (29)*, 2008, p. 547-557.

MARTÍN-VIDE, C., Formal Grammars and languages, in: *The Oxford Handbook of Computational Linguistics*, sous la dir. de Ruslan Mitkov, Oxford, Oxford University Press, 2003, p. 157-177.

MUŇOZ, M., PUNYAKANOK, V., ROTH, D., ZIMAK, D., A learning approach to shallow parsing, in: *Proceedings of EMNLP-WVLC'99*, sous la dir. de Pascale Fung and Joe Zhou, Association for computational linguistics, University of Maryland, 1999, p. 168-178.

POHL, G.: « Angol-magyar szótáralapú főnévicsoport-szinkronizáció és fordításalapú főnévicsoport-meghatározás », in : *III. Magyar számítógépes nyelvészeti konferencia*, sous la dir. de Z. Alexin et D. Csendes, Szeged, SZTE Informatikai tanszékcsoport, 2005.

PRÓSZÉKY, G., KIS, B., *Számítógéppel – emberi nyelven [Intelligens szövegkezelés számítógéppel]*, Bicske, SZAK Kiadó, 1999.

RUSSEL, S., NORVIG, P., *Artificial intelligence – A modern approach* (2e édition), Upper Saddle River, New Jersey, Prentice Hall, 2002.

Szeged Treebank 2.0, Magyar természetesnyelvi adatbázis teljes szintaktikai elemzéssel. Site web des projets du Département de linguistique informatique de l'Université de Szeged. http://www.inf.u-szeged.hu/projectdirs/hlt/ (consulté le 10/11/2008)

VOUTILAINEN, A., « Part-of-speech tagging », in : *The Oxford Handbook of Computational linguistics*, sous la dir. de Ruslan Mitkov, Oxford, Oxford University Press, 2003.