Analyse syntaxique du français parlé

Christophe Cerisara, Claire Gardent

Analyse syntaxique du français parlé

Christophe Cerisara et Claire Gardent
CNRS/LORIA, Nancy
prenom.nom@loria.fr

17 novembre 2009

1 Introduction

Comme l'a montré la campagne d'évaluation des analyseurs syntaxiques menée dans le cadre de l'action EASy/EVALDA[1], il existe pour le Français, divers systèmes d'analyse syntaxique. Le spectre couvert inclut l' analyse syntagmatique profonde [16, 4, 12, 3] et l'analyse en dépendance [5, 8] ; les analyseurs basés sur des grammaires déclaratives [16, 4, 12, 3] et des analyseurs plus procéduraux [5, 8] ; les analyseurs symboliques et les analyseurs stochastiques [6, 2, 13].

Cependant, ces analyseurs sont conçus pour traiter la langue écrite. Ils ne permettent ni de traiter des spécificités de la langue orale telles que les disfluences (répétitions, hésitations, corrections) ni de traiter du caractère imparfait des transcriptions produites par les systèmes de reconnaissance automatique de la parole (SRAP). Pourtant, une piste possible pour l'amélioration des systèmes de reconnaissance de la parole est l'intégration dans le processus de reconnaissance, de connaissances syntaxiques et en particulier, d'une mesure de confiance syntaxique permettant de réordonner les hypothèses produites par le SRAP. Dans ce contexte, un analyseur syntaxique permettant d'associer à chaque analyse produite une probabilité est un élément potentiellement crucial.

Nous présentons ici une infrastructure linguistique et logicielle permettant d'envisager le développement d'un analyseur syntaxique pour le Français parlé. Cette infrastructure vise à faciliter la réutilisation de l'analyseur stochastique MALT [11] pour le français parlé et comprend (i) la définition d'un schéma d'annotation en dépendances et (ii) un environnement logiciel permettant l'annotation syntaxique, l'apprentissage sur un corpus annoté, l'analyse syntaxique et l'évaluation par rapport à une référence.

L'organisation de l'article est la suivante. Dans la section 2, nous présentons le schéma d'annotation utilisé et le comparons aux schémas d'annotations en dépendance existants pour le français à savoir, le format Easy/Passage et le format proposé récemment par [6]. La section 3 présente le logiciel utilisé pour l'annotation, l'apprentissage, l'analyse syntaxique et l'évalua-

[1] http://www.technolangue.net/article198.html

tion. La section 4 est consacrée à la description d'une première expérience sur un corpus de taille réduite (20 000 mots). La section 5 conclut l'article.

2 Schéma d'annotation

Comme le montre la dernière campagne d'évaluation sur l'analyse syntaxico-sémantique multilingues (CoNLL-2009 Shared Task : Syntactic and Semantic Dependencies in Multiple Languages), des schémas d'annotations en dépendances syntaxiques (et sémantiques) ont été définis et utilisés pour de nombreuses langues dont en particulier, la catalan, le chinois, l'anglais, l'allemand, le tchèque, l'espagnol et le japonais.

Pour le français, on peut recenser le schéma EASY [9], le schéma récemment défini par l'équipe INRIA ALPAGE [7] et dans une moindre mesure, le schéma d'anotation des dépendants verbaux utilisé pour le corpus arboré de Paris 7 [1].

Le schéma d'annotation EASY n'est pas réellement un schéma d'annotation en dépendances syntaxiques puisqu'il n'impose (et parfois ne permet) pas que l'annotation syntaxique d'une phrase soit une structure de dépendances. En effet, ce schéma préconise une annotation mixte en constituants et dépendances telle que les relations de dépendances ne relient pas uniquement des mots comme dans une structure de dépendance classique, mais également des mots et des constituants ou des constituants et des constituants. De plus, comme l'indique le tableau 1, le schéma d'annotation choisi ne couvre pas l'ensemble des relations de dépendances syntaxiques possibles entre les mots. Par exemple, la relation entre un nom et un déterminant n'est pas incluse. Plus généralement, l'annotation au format EASY du corpus d'évaluation utilisé dans la campagne EASY n'est pas une annotation en dépendances syntaxiques mais une annotation hybride en constituants et en dépendances.

Le schéma récemment défini par l'équipe INRIA ALPAGE reprend le schéma d'anotation des dépendants verbaux utilisé pour le corpus arboré de Paris 7 [1] et l'étend aux cas et gouverneurs non annotés dans ce corpus dont en particulier, les gouverneurs non verbaux. Les structures résultant de l'annotation sont des arbres orientés où les noeuds correspondent aux formes fléchies de la phrase et où les arcs sont étiquettés par l'une des relations de dépendances permises par le schéma. Nous utilisons un schéma qui s'inspire de ces deux schémas et permet une annotation en graphe de dépendances (un noeud peut avoir plusieurs parents). Les noeuds du graphes sont les tokens identifiés par la reconnaissance de la parole ou par la transcription humaine de l'oral. Les arcs sont étiquettés par l'une des relations de dépendances définies par le schéma. Comme dans le schéma ALPAGE, le schéma n'impose pas la projectivité si bien que la projection d'un noeud ne correspond pas nécessairement à un segment continu de la phrase analysée.

Les relations utilisées sont les suivantes : sujet (*suj*), objet (*obj*), objet prépositionnel (*pobj*), attribut du sujet (*atts*), attribut de l'objet (*atto*), modifieur de verbe (*modV*), de nom (*modN*), d'adjectif (*modAdj*) ou d'adverbe (*modAdv*), complément d'une préposition, d'un complémenteur ou d'un pronom relatif (*comp*), auxiliaire verbal (*aux*), apposition (*appos*), déterminant (*det*), coordination (*cc*), juxtaposition (*juxt*), complément réfléchi (*ref*), partie d'une locution multi-mots (*MultiMots*), expression figée (*dummy*).

Le tableau 1 résume les points communs et les divergences d'avec les schémas existants. Plus

NANCY	ALPAGE	P7	EASY
suj	suj	SUJ	SUJ_V
obj	obj	OBJ	COD_V
pobj	p_obj,	P-OBJ	CPL_V
	de_obj	DE-OBJ	CPL_V
	a_obj	A-OBJ	
	dep		
atts	ats	ATS	ATB_SO
atto	ato	ATO	
modV	mod	MOD	MOD_V
ref			
dummy	aff		
aux	aux_pass		
	aux_caus		
det	det		
modN	mod		MOD_N
comp	arg_cons, arg_comp, obj, p_obj		COMP
cc	coord, arg_coord		COORD
multimots			
	ponct		
modA			MOD_A
modaDV			MOD_R
			MOD_P
appos			APP
juxt			JUXT

FIG. 1 – Relations utilisées par les schémas d'annotation pour le français

généralement, les choix faits pour le schéma d'annotation résultent d'un objectif double.

Premièrement, l'annotation syntaxique doit permettre de distinguer les transcription erronées des transcriptions correctes produites par le système de reconnaissance de la parole. En d'autres termes, les structures de dépendances produites doivent encoder des connaissances syntaxiques. C'est ce qui justifie par exemple l'annotation des réfléchis : comme tous les verbes n'acceptent pas la forme pronominale, cette annotation peut permettre de détecter une phrase peu plausible syntaxiquement lorsqu'un tel verbe apparait dans une structure incluant un argument réfléchi.

Deuxièmement, la structure syntaxique doit permettre à plus long terme le calcul sémantique afin de pouvoir également utiliser des connaissances sémantiques pour contraindre le processus de reconnaissance de la parole. Pour cette raison, les arguments des nominaux déverbaux sont annotés comme tels.

Les différences avec le schéma ALPAGE portent sur la structure (graphe plutôt qu'arbre) et sur la précision de l'annotation. La distinction argument/ajout étant difficile à faire pour les annotatrices, nous avons décidé dans un premier temps, de ne pas différencier les A- et De-objets

des autres objets prépositionnels. Cette différenciation sera faite lors d'une deuxième passe par des linguistes experts. Les juxtapositions et appositions, très fréquentes à l'oral, justifient les relations supplémentaires correspondantes. La relation *dummy* est utilisée comme dans le schéma ALPAGE pour les expressions figées (réfléchis intrinsèques, clitiques figées, etc.) mais également pour l'annotation des répétitions et des hésitations. La relation PONCT n'est pas utilisée car absente des transcriptions. Comme dans le schéma EASY, les modifieurs sont différenciés suivant le type de leur gouverneur (verbe, nom, adjectif) afin de faciliter une deuxième passe visant l'annotation sémantique.

Comme le tableau 1 l'indique, le passage d'un schéma d'annotation à un autre est relativement simple. La conversion du format nancéen vers le format EASY est essentiellement une conversion par traduction ou élimination de relations (e.g., *suj*devient SUJ_V et *det* est éliminé). La conversion vers le format ALPAGE implique en outre soit de regrouper plusieurs catégories en une seule (e.g., regrouper de_obj, a_obj et p_obj sous p_obj), soit de différencier une catégorie unique en plusieurs sous-catégories (e.g., mod en *modV* , *modAdj* , *modAdv* , *modP*).

3 Environnement logiciel

Afin de faciliter l'expérimentation, nous avons développé un environnement logiciel, appelé JSYNATS pour *Java software for Syntax Analysis of Transcribed Speech* et intégrant les fonctionnalités suivantes :

- Annotation : permet d'annoter des textes suivant le schéma d'annotation présenté dans la section précédente
- Analyse : permet d'analyser du texte avec l'analyseur MALT
- Apprentissage : permet d'entraîner l'analyseur MALT sur un ensemble de fichiers annotés
- Evaluation : permet de calculer les performances de l'analyseur par rapport à un corpus de référence

Le logiciel implémenté en Java est disponible à l'url `http://talc.loria.fr/GraphEdit.html`.

Annotation. L'outil d'annotation est un outil de visualisation et d'édition de graphe syntaxique en dépendances qui permet d'annoter du texte conformément au schéma d'annotation JSYNATS . L'édition se fait par des raccourcis claviers opérant sur une interface graphique. Le texte d'entrée peut être ou non analysé. En pratique, l'annotation se fait par correction des analyses produites par MALT . Le format d'entrée et de sortie est le format ConLL. L'outil accepte également le format texte et XML utilisé par l'analyseur Syntex [5]. Une copie d'écran illustrant les représentations manipulées est donnée en Figure 2.

Analyse. La fonctionalité d'analyse permet d'analyser du texte avec l'analyseur MALT . Elle prend en entrée un fichier texte et produit en sortie un fichier texte où chaque phrase contenue dans le fichier entrant est annotée avec l'analyse produite par l'analyseur MALT (format ConLL).

Actuellement, l'analyse en dépendances de MALT est précédée d'une phase d'annotation automatique des séquences de mots en classes morpho-syntaxiques. Cette analyse morpho-syntaxique est réalisée par l'outil TreeTagger. Nous envisageons à court terme d'éliminer cette dépendance de la plate-forme proposée vis-à-vis de TreeTagger afin d'intégrer l'ensemble des outils nécessaires au sein d'un logiciel unique 100 % Java.

En pratique, la fonctionalité d'analyse est utilisée pour la pré-annotation syntaxique des textes permettant ainsi aux annotatrices de travailler sur des textes pré-annotés plutôt que sur des textes sans aucune annotation syntaxique. Comme l'analyseur est ré-entrainé à chaque nouvelle phase d'annotation, la qualité des pré-annotations croît avec le temps, diminuant ainsi les temps d'annotation. En outre, la pré-annotation est généralement correcte au niveau des syntagmes de base (groupes nominaux et prépositionnels non récursifs, noyau verbal, subordonnées relatives simples, etc.), ce qui permet aux annotatrices de se concentrer sur les questions plus complexes liées au rattachement de ces syntagmes entre eux.

Apprentissage. JSYNATS permet d'entraîner MALT sur un ensemble de fichiers contenant du texte annoté syntaxiquement et morpho-syntaxiquement. Des outils de conversion de formats permettent de supporter les formats CONLL, XML (Syntex) et TreeTagger. L'algorithme déterministe de Nivre-Eager est utilisé pour l'analyse, et sa version "oracle" produit pour l'apprentissage un ensemble de vecteurs d'observation, chaque vecteur étant associé à une des quatre "actions" de l'algorithme de Nivre (Shift, Reduce, Left-Arc et Right-Arc). Les vecteurs d'observation incluent les informations suivantes, qui sont celles proposées par défaut dans MALT :

- Formes fléchies et lemmes des deux mots potentiellement dépendants (L et R) ;
- Forme fléchie du mot suivant R ;
- Forme fléchie du mot gouvernant L ;
- Classes morpho-syntaxiques de L et R, du mot précédant L, et des trois mots suivants R ;
- Types des dépendances issues de L, des dépendants les plus à gauche et à droite de L, et du dépendant le plus à gauche de R.

Cet ensemble de vecteurs et leurs classes associées constitue le corpus d'apprentissage d'un classifieur à base de machines à vecteurs supports (SVM) servant dans MALT à décider des dépendances à établir.

Evaluation. L'environnement JSYNATS permet également de calculer les performances d'un analyseur produisant des données au format ConLL. Les scripts d'évaluation sont directement adaptés des scripts distribués pour les campagnes CONLL, et calculent donc les mêmes métriques, en particulier le "score de rattachement en dépendances typées" (*Labeled Attachment Score ou LAS*) [15] utilisé dans cet article, qui représente le pourcentage de mots pour lesquels le système a prédit le bon gouverneur et le bon type de dépendance.

4 Cadre expérimental et évaluation

Nous utilisons JSYNATS pour développer un corpus oral annoté syntaxiquement et entraîner l'analyseur MALT .

4.1 Corpus utilisé et procédure d'annotation

Le corpus utilisé pour l'apprentissage et le test est issu du corpus d'informations radiodiffusées produit par le projet Technolangue ESTER 2003-2005 [10]. Le corpus ESTER comporte les transcriptions manuelles de 37 heures d'émissions radiophoniques d'information francophone (années 1998 - 1999 et 2003). Les transcriptions manuelles de ESTER étant destinées au calcul du taux de reconnaissance des systèmes de reconnaissance automatiques de la parole, seuls les mots complets sont annotés : ainsi, les répétitions sont annotées si les mots répétés sont complets, les "euh" d'hésitation sont considérés comme des mots et sont donc également annotés, mais par contre les bruits, les mots incomplets, bref tout ce qui ne fait pas partie du "lexique" français, n'est pas annoté. Pour ce travail, nous avons également supprimé toute information de ponctuation avant l'étape d'analyse syntaxique, car les sorties des systèmes de transcription automatique ne disposent de ces informations.

Un sous-ensemble de ce corpus composé de 20000 mots est extrait d'émissions de France-Inter datées de 1999. Ce sous-corpus est annoté automatiquement en classes morpho-syntaxiques par l'outil TreeTagger [14], puis converti au format CONLL. Ensuite, ce corpus enrichi est resegmenté en phrases manuellement, puis annoté également manuellement en dépendances syntaxiques selon le guide d'annotation décrit précédemment. Cette annotation en dépendances est en fait réalisée itérativement : chaque itération est décomposée en deux phases, respectivement une phase d'annotation automatique en dépendances réalisée par l'analyseur MALT entraîné avec les données de l'itération précédente, suivie d'une phase de correction manuelle de ces dépendances. Les itérations ont pour objectif d'accroître la taille du corpus, et un nouveau corpus est considéré à chaque itération. L'itération initiale est réalisée avec un petit corpus de 458 mots annoté entièrement manuellement qui sert à entraîner une première version de l'analyseur MALT . Une petite dizaine d'itérations permettent d'aboutir au corpus décrit dans cet article.

Ce corpus annoté est alors divisé en deux parties, respectivement pour l'apprentissage et le test de MALT :

- Apprentissage : La partie du corpus réservée à l'apprentissage contient 13135 mots et 12199 dépendances. Cette partie est utilisée telle quelle pour l'apprentissage de Malt.
- Test : La partie du corpus reservée au test subit une phase supplémentaire de vérification et de correction par une linguiste expert. Elle contient 5305 mots et 4905 dépendances.

Au terme d'environ 6 semaines d'annotation, environ 20 000 mots ont pu être annotés.

4.2 Apprentissage et résultats

MALT est un système pour l'apprentissage d'analyseurs en dépendances syntaxiques. A partir d'un corpus annoté, le système apprend à projeter des traits syntaxiques et morphosyntaxiques sur des décisions d'analyse (shift, reduce, création d'arcs de dépendances). C'est un système libre source implanté en Java et disponible à l'url `http://w3.msi.vxu.se/~nivre/research/MaltParser.html`.

Nous avons entrainé le système MALT sur le corpus d'apprentissage décrit au paragraphe précédent. Les résultats sont largement inférieurs à l'état de l'art avec un score LAS de 71.8% en dépendances typées contre 86.56% pour l'analyseur décrit dans [6]. Ils s'expliquent cepen-

dant par la taille réduite du corpus d'apprentissage (15 000 mots contre 385 458 pour le corpus arboré utilisé pour apprendre les dépendances par [6]) et la courbe de progression (Figure 3) est encourageante.

Nous avons également étudié l'influence de l'étape supplémentaire de vérification des annotations en dépendance par une linguiste expert en incluant neuf dixièmes du corpus de test (vérifié) dans l'apprentissage de MALT , et en testant sur le dixième restant. Le taux de dépendances correctes final est calculé par validation croisée, en faisant varier le dixième du corpus réservé au test. Le score LAS passe alors de 70.3% (aucune phrase d'apprentissage n'a été vérifiée par la linguiste expert) à 71.8%. Ce résultat suggère que les erreurs d'annotation, qui sont présentes en bien plus grand nombre dans le corpus non vérifié, semblent avoir un impact relativement limité sur les performances du système, ce qui résulte probablement du fait que l'apprentissage statistique du classifieur tend à éliminer les erreurs non corrélées et assimilables à du bruit.

Nous avons finalement entraîné le système sur les données annotées fournies par la campagne d'évaluation des analyseurs syntaxiques EASY. Une évaluation préliminaire sur le fichier littéraire_1 de ce corpus donne une F-mesure de 50% en dépendances typées.

5 Conclusion

Cet article présente un environnement logiciel pour l'apprentissage d'analyseurs en dépendances syntaxiques et l'applique à l'apprentissage d'analyseurs en dépendances pour le Français oral et écrit.

Les résultats préliminaires obtenus à partir d'un corpus restreint sont encourageants et permettent d'espérer avoir prochainement à disposition un analyseur syntaxique de l'oral raisonnablement précis. Nous envisageons d'améliorer les performances à la fois par une annotation plus extensive et par la mise en place de techniques d'apprentissage semi-supervisées comme la méthodologie d'apprentissage actif afin d'augmenter la taille du corpus d'apprentissage.

Cet analyseur ayant pour objectif principal d'extraire des informations syntaxiques sur un corpus transmis automatiquement, il reste encore à évaluer ses performances sur des transcriptions automatiques et son impact sur la détection des différents types d'erreurs de la reconnaissance, insertions, omissions et substitutions.

Références

[1] Anne Abeillé. Guide des annotateurs : Annotation fonctionnelle. Technical report, Université de Paris 7, 2004.

[2] Abhishek Arun and Frank Keller. Lexicalization in crosslinguistic probabilistic parsing : The case of french. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 306–313, Ann Arbor, MI, 2005.

[3] Guillaume Bonfante, Bruno Guillaume, and Guy Perrier. Analyse syntaxique électrostatique. *Traitement Automatique des Langues*, 44(3) :93–120, 2003.

[4] Pierre Boullier, Benoit Sagot, and Lionel Clément. Un analyseur lfg efficace pour le français : Sxlfg. In *Actes de TALN 05*, pages 403–40, 2005.

[5] Didier Bourigault, Cécile Fabre, Cécile Frérot, Marie-Paule Jacques, and S. Ozdowska. Syntex, un analyseur syntaxique de corpus. In *actes du colloque TALN*, 2005.

[6] Marie-Hélène Candito, Benoit Crabbé, and Djamé Seddah. On statistical parsing of french with supervised and semi-supervised strategies. In *Proceedings EACL Workshop 2009 : Grammatical Inference for computational linguistics*, 2009.

[7] Marie-HÍène Candito, Benoit Crabbé, and Mathieu Falco. Dépendances syntaxiques de surface pour le français. Technical report, Université de Paris 7, 2009.

[8] Gil Francopoulo. Tagparser et technolangue-easy. In *Actes de l'atelier Easy, TALN*, 2005.

[9] Véronique Gendner, Anne Vilnat, Laurence Monceaux, Patrick Paroubek, Isabelle Robba, and Gil Francopoulo. Les annotations syntaxique de référence peas, version 1.11. Technical report, Projet ANR Passage, 2008.

[10] G. Gravier, J.-F. Bonastre, S. Galliano, E. Geoffrois, K. Mc Tait, and K. Choukri. Ester, une campagne d'évaluation des systèmes d'indexation d'émissions radiophoniques. In *Proc. JEP*, Fez, 2004.

[11] Joakim Nivre, Jens Hall, Jens Nilsson, A. Chanev, G. Eryigit, S. Kübler, S. Marinov, and E. Marsi. Maltparser : A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2) :95–135, 2007.

[12] Azim Roussanaly, Benoît Crabbé, and Jérôme Perrin. Premier bilan de la participation du LORIA à la campagne d'évaluation EASY. In *12e Conférence annuelle sur le Traitement Automatique des Langues Naturelles - TALN 2005*, Dourdan, France, 06 2005. ATALA.

[13] Natalie Schluter and Josef van Genabith. Treebank-based acquisition of lfg parsing resources for french. In *LREC*, 2008.

[14] H. Schmid. Improvements in part-of-speech tagging with an application to german. In *Proc. Workshop EACL SIGDAT*, Dublin, 1995.

[15] M. Surdeanu, R. Johansson, A. Meyers, L. Marquez, and J. Nivre. The conll-2008 shared task on joint parsing of syntactic and semantic dependencies conll 2008. In *Proc. 12th Conference on Computational Natural Language Learning*, pages 159–177, Manchester, August 2008.

[16] Éric Villemonte de La Clergerie. DyALog : a tabular logic programming based environment for NLP. In *Proceedings of 2nd International Workshop on Constraint Solving and Language Processing (CSLP'05)*, Barcelona, Spain, October 2005.

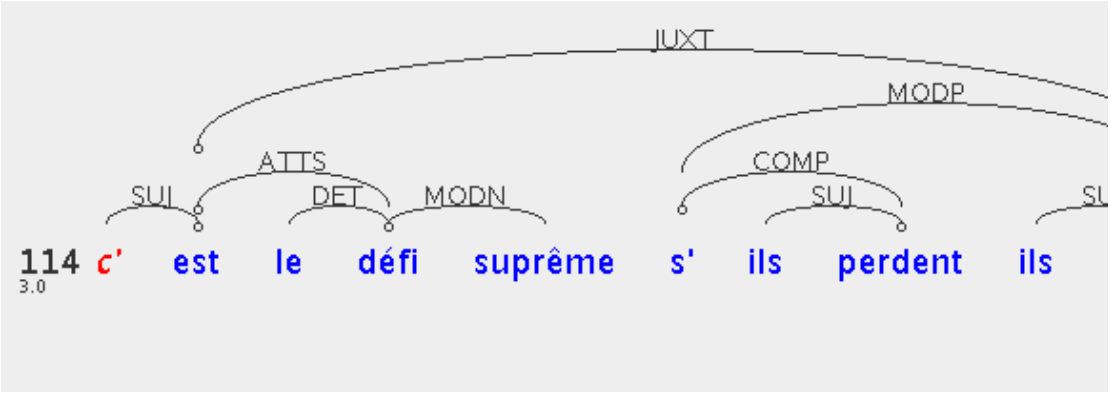

FIG. 2 – Interface graphique de JSYNATS

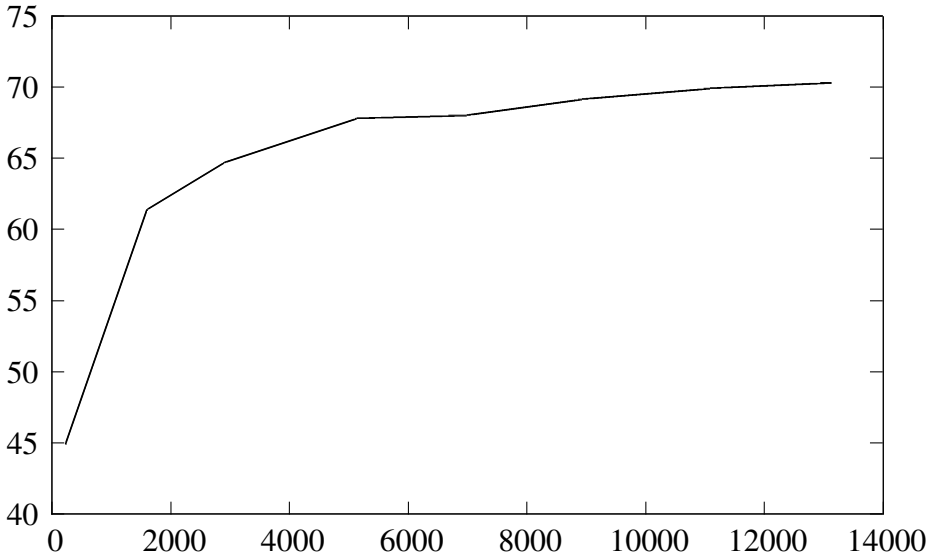

FIG. 3 – Évolution du score de rattachement en dépendances typées (score LAS) obtenu par JSYNATS en fonction de la taille du corpus d'apprentissage. L'axe des abscisses représente le nombre de mots utilisés pour apprendre MALT .