Rapport projet EIT

-

Mohamed Beldi

Alban Descottes

Elodie Lam

-

ET5 INFO - EIT

10/03/2019

Introduction

Pour le Traitement Automatique des Langues (TAL), la description linguistique est, sans doute, la ressource centrale pour automatiser la compréhension et la génération des discours. Mais cette description, faite manuellement, nécessite un temps considérable d'analyse qui ne permet pas son utilisation dans des logiciels industriels. Par ailleurs, une langue est un ensemble qui évolue dans le temps, ce qui occasionne un travail considérable. Aujourd'hui, grâce à la démocratisation et du développement de la communication linguistique par l'outil informatique, la richesse des corpus linguistiques a été multiplié. De plus, l'informatique apporte des outils de traitement et de gestion de masses de données qui sont incomparablement supérieurs à ceux que peut développer un humain. Dans ce contexte et dans le cadre de ce projet d'Extraction d'Information - Traduction automatique, nous allons procéder à l'évaluation de deux plateformes d'analyse linguistique open sources : l'analyseur multilingue LIMA et l'analyseur Stanford CoreNLP. L'objectif de ce projet est donc d'expérimenter ces deux frameworks d'analyse linguistique, de les évaluer et de les comparer entre eux.

L’état de l’art

Outils et plateformes de gestion existantes

L‘université de Grenoble a une longue expérience en édition de dictionnaires pour la traduction automatique, aboutissant aujourd‘hui à une proposition pour un « métaEnvironnement de Développement Linguiciel ». A la même université, l‘une des applications apparues au cours de ces travaux est le dictionnaire multilingue résultant du projet Papillon32, détaillé dans [Mangeot et al., 2003] et issu des travaux de thèse de [Sérasset, 1994]. Le développement de Jibiki, une « plateforme de gestion de ressources lexicales », a été utilisée dans les projets Papillon33, GDEF34 et LexAlp35. Papillon est une base lexicale multilingue à structure pivot couvrant des langues européennes et asiatiques. La structure des entrées est construite selon les principes du DEC36. Le but du projet GDEF est de réaliser un dictionnaire estonien-français de 80.000 articles. Jibiki est aussi utilisé dans le projet LexALP, qui vise l‘harmonisation de la terminologie juridique transalpine en quatre langues : allemand, français, italien et slovène.
Aleth est l‘ensemble de ressources et d‘outils commercialisé par feu GSI-ERLI. Parmi les applications on compte la recherche documentaire, la génération de texte, la traduction etc. Tous les traitements reposent sur la même ressource : une base de connaissance linguistique au format Genelex. Depuis le projet Genelex, toutes les connaissances lexicales ont été centralisées, et des lexiques spécialisés sont extraits en fonction des projets (d‘après la description dans [LIM, 1994]). AlethGD, aussi connu comme Station Genelex, est le système de gestion de cette base de connaissances qui assure « (1) la consultation et la visualisation de l‘ensemble des données, (2) la mise à jour (création, suppression et modification des données du dictionnaire) et (3) les opérations massives sur le dictionnaire : suppression et fusion de listes de données extérieures dans le dictionnaire. Le laboratoire LPL de l‘Université de Provence a développé une plateforme de développement lexical pour la maintenance du lexique DicoLPL. Ce lexique est codé en extension au format texte tabulaire : chaque ligne est un quadruplet .

Motivations

Nous avons choisi l’analyseur multilingue LIMA et l’analyseur Stanford CoreNLP pour notre étude car les deux analyseurs sont open source. Il existe une version du système LIMA sous licence libre. En effet, conformément à la licence Affero GPL v3, il est possible d'utiliser LIMA librement tant que les logiciels qui y sont liés, qu'ils soient distribués ou exécutés à travers le réseau, sont eux aussi des logiciels libres avec une licence compatible. Quant à Stanford CoreNLP, il a initialement été conçu à des fins internes mais il est devenu open source depuis 2010. L'adoption de ces deux analyseurs par une large communauté (chercheurs, industries, applications open source…) est un avantage car cela permettra une amélioration plus rapide et robuste, aussi bien d'un point de vue purement logiciel que scientifique. De plus, l'intérêt pour les utilisateurs de la version Libre est de disposer d'un outil de qualité industrielle dont les développements open source sont supportés par un acteur pérenne (CEA).

Description des deux plateformes

Lima

Le système LIMA (Libre Multilingual Analyzer) est un analyseur linguistique développé par le laboratoire LVIC (Laboratoire Vision et Ingénierie des Contenus) du CEA LIST. Il a été conçu dans le but de l'intégrer à des applications basées sur le traitement du langage naturel.
LIMA supporte l'ensemble des étapes de l'analyse linguistique (morphologie, syntaxe, sémantique, entités, coréférences) selon une architecture modulaire et hautement configurable, avec des performances élevées aussi bien en vitesse qu'en qualité d'analyse. Le système présente un principe de configuration dit « en cascade » qui permet de traiter chaque étape de l'analyse linguistique selon le langage étudié. LIMA est donc un analyseur basé sur un ensemble de règles qui permet de facilement analyser des langages différents. Son interface a été conçu pour être facilement utilisable par des non spécialistes en informatique et sa versatilité lui permet d'être aisément intégrable dans de plus grandes applications. La version Libre est complète avec des modules de traitement et des ressources permettant d'analyser des textes en anglais et en français. La version commerciale est complétée avec des modules et des ressources permettant d'analyser des textes dans de multiples langues supplémentaires comme l'allemand, l'arabe, le chinois, l'espagnol, l'hongrois, l'italien …

Stanford CorNLP

Stanford CoreNLP est un analyseur linguistique développé par des chercheurs de l'université de Stanford aux Etats-Unis. La boîte à outils Stanford CoreNLP fournit la plupart des ressources de base au traitement naturel du langage. A partir d'un texte brute, l'analyseur va fournir une version annotée de ce texte contenant des informations de l'analyse. Le système fournit des annotations préconfigurées pour chaque langue mais l'utilisateur peut aussi les configurés lui-même. Stanford CoreNLP utilise les technologies du machine learning pour améliorer son analyse. Le système est « entraîné » sur des corpus de texte. C'est est donc un analyseur à base de corpus qui permet d'étudier plusieurs langues comme l'anglais, le français, l'allemand, le chinois ou encore l'arabe. Stanford CoreNLP a été développé en langage Java. Le système fournit tout simplement une API Java, sa prise en main est donc très simple car elle demande très peu de configuration. L'utilisateur n'a pas besoin de connaître le langage. De plus, sa flexibilité le rend facilement extensible puisqu'il est compatible avec d'autres langages comme Python, Ruby, Perl, JavaScript, F#, et autres .NET et JVM langages.

Description des expérimentations

Résultats de chaque plateformes pour les parties 2 et 3

```
descottes@descottes-R2D2:~/Documents/TP1_TAL$ python2 evaluate.py wsj_0010_sample.txt.pos.univ.lima wsj_0010_sample.txt.pos.univ.ref
Word precision: 1.0
Word recall: 1.0
Tag precision: 0.781818181818
Tag recall: 0.781818181818
Word F-measure: 1.0
Tag F-measure: 0.781818181818
descottes@descottes-R2D2:~/Documents/TP1_TAL$ python2 evaluate.py wsj_0010_sample.txt.pos.univ.stanford wsj_0010_sample.txt.pos.univ.ref
Word precision: 1.0
Word recall: 1.0
Tag precision: 0.981818181818
Tag recall: 0.981818181818
Word F-measure: 1.0
Tag F-measure: 0.981818181818
descottes@descottes-R2D2:~/Documents/TP1_TAL$ python2 evaluate.py wsj_0010_sample.txt.pos.lima wsj_0010_sample.txt.pos.ref
Word precision: 1.0
Word recall: 1.0
Tag precision: 0.0818181818182
Tag recall: 0.0818181818182
Word F-measure: 1.0
Tag F-measure: 0.0818181818182
descottes@descottes-R2D2:~/Documents/TP1_TAL$ python2 evaluate.py wsj_0010_sample.txt.pos.stanford wsj_0010_sample.txt.pos.ref
Word precision: 1.0
Word recall: 1.0
Tag precision: 0.981818181818
Tag recall: 0.981818181818
Word F-measure: 1.0
Tag F-measure: 0.981818181818
```

Globalement Lima est précis à 78% et Stanford CorNLP à 98%.

Les résultats sont un peu meilleurs en utilisant les tags universels puisque la précision de Lima augmente à 81%. Quant à Stanford CorNLP, il reste précis à 98%. De plus, avec l'utilisation de Lima, on peut noter qu'avec l'exemple que nous avons utilisé, à savoir le fichier wsj_0010_sample.txt, on observe une perte de certains mots dans l'analyse. Dans la première phrase, on perd le " 's " après nation. Cette perte, ajouté aux mots composés, donne certaines explications quant aux résultats moins bons qu'avec l'outil de Stanford.

Résultats de l'extraction des entités nommées de la partie 4

results / iv / resultPart4ws10.txt

Entité nommée	Type	Nombre d'occurence	Proportion dans le texte
Hot	LOCATION	1	11.11111111111111
Association	ORGANIZATION	1	11.11111111111111
Springs	LOCATION	1	11.11111111111111
Indianapolis	LOCATION	1	11.11111111111111
National	ORGANIZATION	1	11.11111111111111
Boca	LOCATION	1	11.11111111111111
Raton	LOCATION	1	11.11111111111111
Manufacturers	ORGANIZATION	1	11.11111111111111
of	ORGANIZATION	1	11.11111111111111

Résultats de chaque plateformes pour la partie 5

Les résultats que nous avons obtenu grâce à nos scripts était très décevant. En effet nous sommes tombés sur des résultats avec une précision autour de 10%. Nous nous sommes alarmés face à ces résultats. En effet nous avons trouvé d'où venait le problème. Le fichier que nous avons utilisé n'était pas bien écrit, les phrases n'étaient pas toutes sur une ligne. Cette mise en forme pose problème dans le scripts qui transmet le fichier référence utilisable avec evaluate.py. Les analyseurs Lima et Stanford mettaient les bons tags mais la comparaison avec le fichier de référence n'était pas bonne. Nous avons donc dû changer à la main les phrases pour que la comparaison soit correcte.

Comparaison Lima / Référence

```
Word precision: 0.824758842444
Word recall: 0.832792207792
Tag precision: 0.789389067524
Tag recall: 0.797077922078
Word F-measure: 0.828756058158
Tag F-measure: 0.793214862682
```

Comparaison Stanford / Référence

```
Word precision: 0.78729689808
Word recall: 0.86525974026
Tag precision: 0.768094534712
Tag recall: 0.844155844156
Word F-measure: 0.824439288476
Tag F-measure: 0.804331013148
```

Pour résumer, dans cette partie, on compare seulement les entités nommées comme les Personnes, les Organisations et les localisations. Les résultats que nous obtenons sont donc corrects. On observe cependant une meilleur précision avec l'outil de Stanford.

Conclusion

Limitations de Lima

La performance du POS tagger est un peu en dessous des autres taggers sur le marché pour la langue anglaise (environ 96% ou 97%). LIMA n'est pas encore au même niveau que les meilleurs systèmes sur l'analyse syntaxique, en particulier sur les relations. Cependant, la vitesse de traitement est plutôt bonne par rapport aux autres processeurs NLP. Néanmoins, la vitesse de performance dépend des langages. LIMA est un système à une grande échelle, sa complexité rend difficile de prédire l'impact de changements comme l'ajout ou la modification d'une règle linguistique. En effet, il pourrait y avoir des effets secondaires imprévisibles.

Limitations de Stanford CorNLP

La possibilité de personnalisation de Stanford CorNLP est moindre. Bien que la possibilité de transmettre des propriétés à des annotateurs soit prise en charge, elle est nettement plus encombrante que l'interface de pipeline d'annotations et est généralement déconseillée. De plus, rien ne garantit que le même algorithme sera utilisé pour calculer la fonction demandée à chaque invocation. Par exemple, si une analyse de dépendance est demandée, suivie d'une analyse de circonscription, nous allons calculer l'analyse de dépendance avec le Neural Dependency Parser, puis utiliser le Stanford Parser pour l'analyse de circonscription. Si, toutefois, vous demandez l'analyse de la circonscription avant l'analyse de la dépendance, nous utiliserons le Stanford Parser pour les deux.

Bibliographie

Hamon, T. (2014, 02). Soft: L'analyseur multilingue LIMA sous licence libre. Récupérée 03, 2019, à partir de http://listserv.linguistlist.org

Besançon, R. (2005, 06). L'analyseur syntaxique de Lima dans la campagne d'évaluation EASY. Récupérée 02, 2019, à partir de https://perso.limsi.fr

Cailliau, F. (2010, 12). Des ressources aux traitements linguistiques : le rôle d'une architecture linguistique. Récupérée 03, 2019, à partir de https://tel.archives-ouvertes.fr

Cartier, E. (2012, 02). Néologie et description linguistique pour le TAL. Récupérée 03, 2019, à partir de https://www.cairn.info

Simple CoreNLP. Récupérée 03, 2019, à partir de https://stanfordnlp.github.io

Articles:
LIMA: A Multilingual Framework for Linguistic Analysis and Linguistic Resources Development and Evaluation
The Stanford CoreNLP Natural Language Processing Toolkit