PyRATA, Python Rule-based feAture sTructure Analysis

Nicolas Hernandez
LS2N, Université de Nantes, France
nicolas.hernandez@univ-nantes.fr

MOTS-CLÉS : analyse à base de règles, annotation sémantique, expression régulière, extraction d'information, fouille de texte, Python 3.

KEYWORDS: rules-based analysis, semantic annotation, regular expression, information extraction, text mining, Python 3.

1 Résumé

Les approches à base de règles ne doivent pas être opposées à celles à base d'apprentissage automatique. Les premières sont connues pour permettre d'obtenir rapidement des résultats auto-explicatifs avec seulement quelques règles, mais présentent l'inconvénient d'être difficiles à maintenir lorsque les règles deviennent complexes et que leur nombre croît. Les approches à base d'apprentissage sont capables de généralisation, et donc, de fournir des résultats même sur des données jamais rencontrées, mais elles requièrent des données d'entraînement lesquelles résultent souvent d'une tâche d'annotation manuelle coûteuse, et leurs résultats sont plus difficiles à expliquer. Actuellement les approches à base de règles ne sont pas populaires dans la communauté du Traitement Automatique des Langues (TAL). Néanmoins il y a encore de bonnes raisons de les utiliser : 1) Puisqu'elles ne requièrent pas de données d'entraînement, elles constituent une bonne première approche pour explorer des données et définir plus précisément un problème ; 2) Pour certaines langues, des problèmes peuvent être traîtées de manière déterministe ; 3) Pour produire des données qui serviront à entraîner un système à base d'apprentissage ; 4) Pour extraire des traits des données et laisser un système à base d'apprentissage apprendre à les combiner ; 5) Pour augmenter les capacités d'un système à base d'apprentissage en pré- ou post-traitant les données afin d'obtenir une performance de 100 % (Manning, 2011).

La communauté du TAL bénéficie de quelques solutions logicielles [1] pour rechercher des motifs d'annotations et traiter les résultats, à savoir : *GATE JAPE* [2] (Cunningham *et al.*, 1999) et *UIMA RUTA* [3] (Kluegl *et al.*, 2016). Ces solutions requièrent une prise en main de leur langage d'expression des règles, et s'insèrent dans l'adoption d'un cadre d'analyse textuelle plus global, qui a son tour requière la prise en main de ses concepts ainsi que quelques bagages techniques (RUTA est très intégré à l'environnement de développement Eclipse). De plus, le programmeur aura préférentiellement à développer en Java pour utiliser ces solutions. Les utilisateurs du langage de programmation Python ne bénéficient pas d'instruments aussi avancés, mêmes si quelques modules sont disponibles à savoir :

1. Nous ne considérons pas ici les environnements tels que l'*IMS Open Corpus Workbench (CWB) Corpus Query Processing (CQP)* (Evert & Hardie, 2011), Nooj (Silberztein, 2005) `http://www.nooj4nlp.net` ou Unitex (Paumier *et al.*, 2009) `http://unitexgramlab.org` qui sont très ancrés dans une analyse linguistique.
2. `https://gate.ac.uk/sale/tao/splitch8.html`, Java 8, GNU
3. `https://uima.apache.org/ruta.html`, Java 8, Apache v2

Python nltk chunk (Bird, 2006), *clips pattern.search* (De Smedt & Daelemans, 2012) et *spaCy*. Le module spaCy présente les meilleures performances en temps de traitement mais a une expressivité très limitée et requière des compétences en programmation. Le module pattern se focalise sur certains types d'annotation et contraint à l'usage de ses traitements linguistiques. Le module pattern n'autorise pas d'exprimer des motifs sur plus d'un type d'annotation à la fois.

Nous présentons *PyRATA* (*P*ython *R*ules-based fe*A*ture s*T*ructure *A*nalysis) un module Python (version 3) diffusé sous licence Apache V2 et disponible sur github [4] et dans les dépots `pypi` [5]. PyRATA a pour objectif de permettre de l'analyse à base de règles sur des données structurées. Le langage de PyRATA offre une expressivité qui couvre les fonctionnalités proposées par les modules alternatifs et davantage. Conçu pour être intuitif, la syntaxe des motifs et l'interface de programmation (API) suivent les définitions de standards existants, respectivement la syntaxe des expressions régulières de Perl et l'API du module Python `re`. PyRATA travaille sur des structures de données simples et natives de Python : une liste de dictionnaires (c-à-d une liste de tables d'associations). Cela lui permet de traiter des données de différentes natures (textuelles ou non) telles qu'une liste de mots, une liste de phrases, une liste de messages d'un fil de discussion, une liste d'événements d'un agenda... Cette spécificité le rend indépendant de la nature des annotations (a fortiori linguistiques) associées à la donnée manipulée. Ce travail a été financé par le projet ANR 2016 PASTEL [6].

Références

BIRD S. (2006). Nltk : The natural language toolkit. In *Proceedings of the COLING/ACL on Interactive Presentation Sessions*, COLING-ACL '06, p. 69–72, Stroudsburg, PA, USA : Association for Computational Linguistics.

CUNNINGHAM H., CUNNINGHAM H. & TABLAN V. (1999). Jape : a java annotation patterns engine.

DE SMEDT T. & DAELEMANS W. (2012). Pattern for python. *Journal of Machine Learning Research*, **13**, 2063–2067.

EVERT S. & HARDIE A. (2011). Twenty-first century corpus workbench : Updating a query architecture for the new millennium. In *Proceedings of the Corpus Linguistics 2011 conference*, University of Birmingham, UK.

KLUEGL P., TOEPFER M., BECK P.-D., FETTE G. & PUPPE F. (2016). Uima ruta : Rapid development of rule-based information extraction applications. *Natural Language Engineering*, **22**, 1–40.

MANNING C. D. (2011). *Part-of-Speech Tagging from 97% to 100% : Is It Time for Some Linguistics ?*, In A. F. GELBUKH, Ed., *Computational Linguistics and Intelligent Text Processing : 12th International Conference, CICLing 2011, Tokyo, Japan, February 20-26, 2011. Proceedings, Part I*, p. 171–189. Springer Berlin Heidelberg : Berlin, Heidelberg.

PAUMIER S., NAKAMURA T. & VOYATZI S. (2009). UNITEX, a Corpus Processing System with Multi-Lingual Linguistic Resources. In *eLexicography in the 21st century : new challenges, new applications (eLEX'09)*, p. 173–175.

SILBERZTEIN M. (2005). Nooj : A linguistic annotation system for corpus processing. In *Proceedings of HLT/EMNLP on Interactive Demonstrations*, HLT-Demo '05, p. 10–11, Stroudsburg, PA, USA : Association for Computational Linguistics.

4. `https://github.com/nicolashernandez/PyRATA`
5. `https://pypi.python.org/pypi/PyRATA`
6. `http://www.agence-nationale-recherche.fr/?Projet=ANR-16-CE33-0007`