Un outil d'étiquetage rapide, libre et open source

Yoann Dupont

Un outil d'étiquetage rapide, libre et open source

Yoann Dupont[1]
(1) Laboratoire Lattice (CNRS, ENS, Université Sorbonne Nouvelle, PSL Research University, USPC)
1 rue Maurice Arnoux, 92120 Montrouge
yoa.dupont@gmail.com

RÉSUMÉ

Dans cet article, nous présentons un outil pour effectuer l'étiquetage rapide de textes bruts. Il peut charger des documents annotés depuis divers formats, notamment BRAT et GATE. Il se base sur des raccourcis claviers intuitifs et la diffusion d'annotation à l'échelle du document. Il permet d'entraîner des systèmes par apprentissage que l'on peut alors utiliser pour préannoter les textes.

ABSTRACT

A fast tagging tool, free and open source

In this article we present a tool for fast tagging of raw texts. It handles multiple input and output formats, such as BRAT and GATE. For fast tagging, the tool relies on intuitive keyboard shortcut and document-wide annotation broadcasting. The tools allows to train machine learning systems that can be used to preannotate texts.

MOTS-CLÉS : étiquetage, annotation hiérarchique, GUI, CRF, résaux de neurones.

KEYWORDS: tagging, hierarchical tagging, GUI, CRF, neural networks.

1 Introduction

Pour de nombreuses tâches de TAL, des textes annotées sont capitaux mais demeurent trop peu nombreux, ou ont une licence restrictive. Il existe déjà de nombreux outils pour annoter des textes bruts, parmi lesquels nous pouvons citer GATE (Cunningham *et al.*, 2013) ou BRAT (Stenetorp *et al.*, 2012). Ces outils ont cependant deux inconvénients principaux : le premier est d'être plutôt lents pour annoter et le second est qu'ils ne gèrent qu'un format, le leur. Pour cette raison, nous proposons ici un outil d'annotation rapide et capable de gérer des données de formats divers. L'outil que nous présentons est un module de SEM (Dupont, 2017).

2 L'outil

L'outil que nous présentons a été conçu pour annoter rapidement dans le cadre de tâches comme l'étiquetage morphosyntaxique ou la reconnaissance d'entités nommées, mais peut se montrer utile pour toute tâche où des empans textuels doivent être annotés comme le parsing. Il est écrit en python en utilisant la librairie Tkinter (Shipman, 2013) Il permet d'annoter un corpus document par document. Afin d'améliorer la vitesse d'annotation l'outil recourt à des raccourcis claviers déduits du jeu d'annotation qui doit être chargé (plusieurs jeux peuvent être gérés de manière indépendante).

Si le jeu d'annotation contient un type "lieu", son raccourci par défaut sera "l". Nombre d'éléments peuvent se trouver répétés à de nombreuses reprises dans le texte. Par exemple, annoter toutes les occurrences d'une même personnes d'un roman peut s'avérer fastidieux et sujet à l'erreur s'il faut annoter les éléments un à un. Pour combler ce problème, si l'utilisateur souhaite annoter un élément textuel, il peut diffuser l'annotation à l'échelle du document. Cette opération n'est pas sans source d'erreur, il n'est pas impossible que certains "Rennes" annotés soient en fait "Inria de Rennes". Pour gérer ce cas, l'outil propose d'explorer l'historique des annotations, classées par date décroissante, effectuées par l'utilisateur afin de les réviser. Une autre source de vitesse est dans l'utilisation de données préannotatées et dans l'apprentissage de systèmes à partir des données annotées.

Nous offrons une gestion simple des annotations hiérarchiques, chaque niveau de la hiérarchie peut s'annoter indépendamment, la cohérence des annotations proposées par l'utilisateur étant gérée automatique. Il est également possible de diffuser les sous-étiquettes : il est par exemple possible d'annoter l'intégralité des "Rennes" déjà annotés en tant que "nom" en tant que "nom propre".

L'outil propose actuellement d'entraîner des CRF (Lafferty *et al.*, 2001) à l'aide de Wapiti (Lavergne *et al.*, 2010) ainsi que des modèles neuronaux pour l'annotation de séquences (Lample *et al.*, 2016). À terme, il proposera également d'entraîner des systèmes sur d'autres tâches comme le parsing.

Nous souhaitons que les annotations effectuées soient réutilisables d'autres outils d'annotation. En effet, ces outils proposent des analyses plus fines des textes, il convient donc que les données annotées soient réutilisables. À cet effet, il est possible de convertir les données en divers formats, entre autres GATE, BRAT et un format XML-TEI.

Références

CUNNINGHAM H., TABLAN V., ROBERTS A. & BONTCHEVA K. (2013). Getting more out of biomedical documents with gate's full lifecycle open source text analytics. *PLoS computational biology*, **9**(2), e1002854.

DUPONT Y. (2017). Exploration de traits pour la reconnaissance d'entités nommées du Français par apprentissage automatique. In *24e Conférence sur le Traitement Automatique des Langues Naturelles (TALN)*, p. 42–55.

LAFFERTY J., MCCALLUM A. & PEREIRA F. (2001). Conditional random fields : Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML 2001*, p. 282–289.

LAMPLE G., BALLESTEROS M., SUBRAMANIAN S., KAWAKAMI K. & DYER C. (2016). Neural architectures for named entity recognition. *arXiv preprint arXiv :1603.01360*.

LAVERGNE T., CAPPÉ O. & YVON F. (2010). Practical very large scale CRFs. In *Proceedings of ACL'2010*, p. 504–513 : Association for Computational Linguistics.

SHIPMAN J. W. (2013). Tkinter 8.4 reference : a gui for python.

STENETORP P., PYYSALO S., TOPIĆ G., OHTA T., ANANIADOU S. & TSUJII J. (2012). Brat : a web-based tool for nlp-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, p. 102–107 : Association for Computational Linguistics.