TP Génie logiciel - Scrum

2023

Parseur d'articles scientifiques en format texte (carnet du produit)

Les chercheurs au Laboratoire Informatique d'Avignon (LIA) doivent lire des articles scientifiques publiés partout dans le monde. Malheureusement ils n'ont pas le temps de tout lire et voudraient avoir un système qui les présente un aperçu de l'article. Cela permettra de leur faire gagner du temps. Souvent les articles scientifiques sont en format PDF qui est loin d'être portable et loin d'être facile à analyser par les systèmes de Traitement Automatique de Langues (TAL). Au LIA on préfère des versions des articles au format texte (.txt) ce qui permet aux systèmes TAL de travailler correctement. D'où l'idée de faire une transformation de PDF a texte. Cependant cela n'est pas sans problèmes : pensez à la gestion de textes en double colonne, aux formules mathématiques, aux figures, etc. Le résultat peut être un fichier texte assez décomposé.

Le LIA considère qu'un minimum acceptable consiste à extraire les sections de l'article PDF dans une version texte (correctement) découpée. Cela implique une analyse capable d'identifier le titre, les auteurs, l'introduction, le développement, conclusions, etc.

Le LIA a besoin d'un analyseur (parseur ou *parser* en anglais) de documents texte convertis à partir d'un PDF. L'objectif étant de découper le mieux possible les sections d'un article (voir un exemple page suivante). Le LIA nécessite d'un système écrit dans un langage de programmation adéquat (voir plus loin) fonctionnant sur un système GNU/Linux.

Le système final et les versions intermédiaires doivent être placés sur Github ; chaque version dans une branche différente. Vous devrez créer un dépôt Github au début de la première séance et le partager avec votre enseignant. Le fichier README.md doit contenir une explication générale du système et la procédure nécessaire pour lancer les programmes.

Il faut documenter les différents événements de la méthode Scrum en ajoutant sur l'ENT des photos et captures d'écran de votre travail : réunions de planification de sprint, mêlée quotidienne, revue de sprint, rétrospective du sprint.

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 142 (2018) 339–346

Procedia Computer Science

www.elsevier.com/locate/procedia

The 4th International Conference on Arabic Computational Linguistics (ACLing 2018), November 17-19 2018, Dubai, United Arab Emirates

Automated Sentence Boundary Detection in Modern Standard Arabic Transcripts using Deep Neural Networks

Carlos-Emiliano González-Gallardo[a,*], Elvys Linhares Pontes[a], Fatiha Sadat[b], Juan-Manuel Torres-Moreno[a,b,c]

[a]LIA - Université d'Avignon et des Pays de Vaucluse, 339 chemin des Meinajaries, 84140, Avignon, France
[b]Université du Québec a Montréal, C.P. 8888, succ. Centre-ville, Montréal (Québec) H3C 3P8 Canada
[c]GIGL, École Polytechnique de Montréal, C.P. 6079, succ. Centre-ville, Montréal (Québec) H3C 3A7 Canada

Abstract

The increased volumes of Arabic sources of data available on the Web has boosted the development of Natural Language Processing (NLP) tools over different tasks and applications. However, to take advantage from a vast amount of these applications, a prior segmentation task call Sentence Boundary Detection (SBD) is needed. In this paper we focus on SBD over Modern Standard Arabic (MSA) by comparing two different approaches based on Deep Neural Networks (DNN) using out-of-domain and in-domain training data with only lexical features (represented as character embedding) while conducting two scenarios based on a Convolutional Neural Network and a Recurrent Neural Network with attention mechanism architectures. While tuning a big out-of-domain dataset with a smaller in-domain dataset, improves the performance in general. Our evaluations were based on IWSLT 2017 TED talks transcripts and showed similarities and differences depending of the SBD method. MSA carries certain complications given its rich and complex morphology. However, using only lexical features for Arabic SBD is an acceptable option when the source audio signal is not available and a certain level of language independence needs to be reached.

Keywords: Sentence Boundary Detection; Speech-to-Text; Transcription; Modern Standard Arabic; Deep Neural Networks

1. Introduction

Arabic language is known to be challenging given its complex linguistic structure [3] and dialect variations [14]. However, the development of Arabic Natural Language Processing (NLP) tools has increased these last years, creating a large set of state-of-the-art applications including POS taggers, syntactic parsers, information retrieval, machine translation, automatic speech recognition and synthesis systems [9, 17]. Some NLP libraries and tools like Python

Article scientifique en PDF

References

[1] Alotaiby, F., Foda, S., Alkharashi, I., 2010. Clitics in arabic language: a statistical study, in: 24th Pacific Asia Conference on Language, Information and Computation.
[2] Althobaiti, M., Kruschwitz, U., Poesio, M., 2014. Aranlp: A java-based library for the processing of arabic text, in: LREC.
[3] Attia, M., Somers, H., 2008. Handling Arabic morphological and syntactic ambiguity within the LFG framework with a view to machine translation. volume 279. University of Manchester Manchester.
[4] Bahdanau, D., Cho, K., Bengio, Y., 2014. Neural machine translation by jointly learning to align and translate. CoRR abs/1409.0473.
[5] Baldridge, J., 2005. The OpenNLP project. http://opennlp.apache.org/.
[6] Bojanowski, P., Grave, E., Joulin, A., Mikolov, T., 2016. Enriching word vectors with subword information. preprint arXiv:1607.04606 .
[7] Diab, M., 2009. Second generation amira tools for arabic processing: Fast and robust tokenization, pos tagging, and base phrase chunking, in: 2nd International Conference on Arabic Language Resources and Tools.
[8] El-Masri, M., Altrabsheh, N., Mansour, H., Ramsay, A., 2017. A web-based tool for arabic sentiment analysis. Procedia Computer Science 117, 38–45.
[9] Farghaly, A., Shaalan, K., 2009. Arabic natural language processing: Challenges and solutions. ACM Transactions on Asian Language Information Processing (TALIP) 8, 14.
[10] Ferrucci, D., Lally, A., 2004. UIMA: An Architectural Approach to Unstructured Information Processing in the Corporate Research Environment. Natural Language Engineering 10, 327–348. doi:10.1017/S1351324904003523.
[11] González-Gallardo, C.E., Torres-Moreno, J.M., 2018. Sentence Boundary Detection for French with Subword-Level Information Vectors and Convolutional Neural Networks. preprint arXiv:1802.04559 .
[12] Gotoh, Y., Renals, S., 2000. Sentence boundary detection in broadcast speech transcripts, in: ASR2000-Automatic Speech Recognition: Challenges for the new Millenium ISCA Tutorial and Research Workshop (ITRW).

Titre :
Automated Sentence Boundary Detection in Modern …

Auteurs :
Carlos-Emiliano González-Gallardo
Elvys Linhares Pontes
Fatiha Sadat
Juan-Manuel Torres-Moreno

Article scientifique en format texte

Abstract :
The increased volumes of Arabic sources of …

Introduction :
Arabic language is known to be challenging given …

Corps :
2. Sentence Boundary Detection for MSA Sentence Boundary Detection (SBD) is of vital importance given that in general, ASR systems focus on obtaining …

Conclusion :
In this paper we have studied the impact of …

Discussion :
Results for S1 and S2show that unbalanced classes …

Bibliographie :
[1] Alotaiby, F., Foda, S., Alkharashi, I., 2010. Clitics in arabic language: a statistical study, in: 24th Pacific Asia Conference on Language,Information and Computation.
[2] Althobaiti, M., Kruschwitz, U., Poesio, M., 2014. Aranlp: A java-based library for the processing of arabic text, in: LREC.
[3] Attia, M., Somers, H., 2008. Handling Arabic morphological and syntactic ambiguity within the LFG framework with a view to machinetranslation. volume 279. University of Manchester Manchester.

Sprint 1 Génie logiciel – Scrum - 1h30

Parseur d'articles scientifiques en format texte

Tout se passe au niveau terminal et ligne de commande : pas de convertisseurs PDF en ligne

Sur ENT vous disposez d'un corpus de 10 fichiers PDF d'articles scientifiques. Le parseur doit agir sur la version plain texte des articles. Ce corpus est le **corpus d'apprentissage**.

Vous disposez de 2 outils libres pour la conversion PDF à texte : **`pdftotext`** et **`pdf2txt`** avec des **options de lancement** qu'on aimerait tester.

Lequel de 2 logiciels est le plus adapté ? Quels options ? On l'ignore.

Vous devez tester tous les deux pour convertir les PDF du corpus d'apprentissage en plain texte. Ensuite, vous devez évaluer qualitativement (manuellement par inspection) les sorties afin de décider lequel utiliser : les frontières des phrases sont bien respectées ? Les mots sont mal découpés ? Les lignes (ou phrases) sont entremêlées par rapport au PDF ?, etc.

Note : le client (le LIA) a décidé de bonifier les équipes Scrum de la façon suivante :

- L'équipe ayant obtenu la meilleure Précision aura **2 points de plus** sur la note finale.
- L'équipe ayant obtenu la 2ème meilleure Précision aura **1 point de plus** sur la note finale.

La précision correspond à une mesure du découpage exact des sections, et elle sera définit formellement ultérieurement. Cette bonification aura lieu lors de l'évaluation finale du produit.

Consignes

- Équipes : 3 ou 5 personnes (pas un nombre paire)
- Méthodologie agile : SCRUM
- Définir le maître SCRUM
- Le maître SCRUM doit rendre (sur ENT) un petit rapport simple (1-2 pags max) avec les **options de lancement** des outils, ainsi que leurs avantages et inconvénients. Le rapport peut être fait à la main sur papier (déposer une image)
- Aucune conversion PDF-texte en ligne via navigateur n'est acceptée
- Soyez agiles !

Vous êtes en compétition ! Attention aux fuites des idées ou de logiciel !