Scientific article analysis

CAPPELLE, Bryan
cappelle.e1802256@etud.univ-ubs.fr

MAZALEYRAT, Erwan
mazaleyrat.e1800109@etud.univ-ubs.fr

DEHOUSSE, Erwan
dehousse.e1800327@etud.univ-ubs.fr

CORNEC, Kilian
cornec.e1801357@etud.univ-ubs.fr

Avril 2021

Abstract

Cette article explique l'ensemble des étapes et les résulats pour parser des articles scientifiques. Le parse des articles permet d'isoler le titre, les auteurs et les différentes parties. Enfin la precision du parseur est calculé en comparant le résultat sur un ensemble d'article avec le parseur et de manière manuelle.

1 Méthode

Pour le développement du parseur nous avons explorés plusieurs solutions possibles :

1.1 Intelligence Artificielle

Tout d'abords nous avons regardé pour la possibilité de le faire via un algorithme de machine learning. Malheureusement après quelques recherches

nous avons laissé tomber cette idée car nous avions trop peu d'exemple, c'est à dire une dizaine ce qui est trop peu pour entrainer un modèle. Nous aurions pu en créer d'autres mais ça aurait été bien trop long.

1.2 Regex & Intelligence Artificielle

Nous avons donc testé une autre solution qui avait pour objectif d'utiliser des regexs pour reconnaitre les parties (introduction, discution, conclusion, ...) et un modèle déjà entrainé pour reconnaitre les noms propres.

Titre : Pour la recherche du titre les 5 premières lignes sont prises et retourne celles où ils n'y a pas de nombre ni de nom propre. Ensuite les grandes parties :

Auteurs : On boucle sur i lignes, i étant fixé à 10 de base. Puis on découpe chaque lignes en mots et pour chaque mots on applique le ner_tagger de la librairie nltk pour analyser si le mot est un Nom propre ou pas. Dans le cas où c'est un nom propre on le rajoute à une variable temporaire en attendant de trouver la suite du prénom. Une fois qu'on a un nom complet on le rajoute à la chaîne de caractère que l'on renvera! De plus à chaque fois qu'on trouve un nom on rajoute 5 lignes de plus à analyser car entre les noms il y a parfois des informations

Afiliation : On prends en entrée une liste des auteurs puis on boucle sur les lignes du texte. Pour chaque ligne si on a pas trouvé un auteur dans une ligne alors on rajoute des informations dans notre variable temporaire une fois qu'on à trouver un nom on rajoute le contenu qu'il y a entre le nom et un autre nom ou un email car on suppose que les informations seront toujours écrit dans ce sens : @Nom1 @info1 @email1 @nom2 + @Nom2 @info2 @email2 ... = la même chose sans le mail. On renvoi une liste des affiliations par auteur. On arrête la boucle quand on trouve la ligne qui on indique qu'on est à l'abstract!

Introduction : Une première regex recherche le mot clé "Introduction" et toute les lignes sont récupérés jusqu'à tomber sur le début de la deuxième partie aussi détectée grâce à une regex.

Abstract :
Ici c'est le même principe sauf que le mot de début est "Abstract" et que le mot de fin soit "Introduction".

Corps :
Même principe :
Mot de début : Début de la deuxième partie
Mot de fin : "Discussion" ou "Conclusion"

Conclusion :
Mot de début : "Conclusion"
Mot de fin : "References" ou "acknowlegments"

Discution :
Mot de début : "Discussion"
Mot de fin : "acknowlegments" ou "Conclusion"

Biblio :
Mot de début : "References"
Jusqu'à la fin

Enfin pour que les regexs aient le plus de chance possible de trouver le bon mot clé, la recherche s'effectue seulement à partir de la fin de la dernière partie trouvée. Aussi de nombreux tests de regex différentes ont été testées pour maximiser la réussite de celles-ci dans plusieurs articles différents.

2 Résultats

Pour tester les résultats comme dis précédemment on compare les résultats d'un set d'article de test avec le parseur et sans le paseur (de manière manuelle). Pour chaque article on donne la précission du parseur via la formule ci-dessous :
$Precision = (Titre + Auteurs + SectionsCorrectes)/(Titre + Auteurs + SectionsVeritables)$ Une fois cela fait on fait la moyenne des précisions. Voici donc la precision pour chaques articles :

acl2012: 64% (stricte) et 73% (souple)
b0e5c43edf116ce2909ae009cc27a1546f09: 72% (stricte) et 83% (souple)
BLESS: 75% (stricte) et 88% (souple)
C14-1212: 68% (stricte) et 77% (souple)
Guy: 72% (stricte et souple)
infoEmbeddings: 56% (stricte et souple)
IPM1481: 0% (stricte et souple)
L18-1504: 59% (stricte) et 68% (souple)
OntheMoralityofArtificialIntelligence: 50% (stricte et souple)
surveyTermExtraction: 69% (stricte) et 81% (souple)

La moyenne est donc de : 58,7% (stricte) et 67,5% (souple).

3 Conclusion

Pour conclure on peut dire que le projet est arrivé à son terme avec une précision convenable. Avec presque 60% de précision en méthode stricte et 68% en méthode souple, notre programme permettra d'analyser la grande majorité des articles scientifiques qui lui seront fournis. Afin d'améliorer les performances du programme, on pourrait rélfléchir à un moyen d'affiner davantage les expressions régulières responsables de la détection de chaque partie. Avec plus de connaissance, il serait également possible de mettre en place du *Machine Learning* ce qui permettrait au logiciel d'améliorer considérablement sa précision.