Analyseur d'articles scientifiques

Donovan Casagrande [*†]
Benoît Maurice [‡]
Arthur Gros [§]
Benjamin Haurogné [¶]
Université Bretagne Sud, Vannes, France

[*] casagrande.e2002711@etud.univ-ubs.fr
[†] Scrum Master
[‡] maurice.e2100135@etud.univ-ubs.fr
[§] gros.e2101298@etud.univ-ubs.fr
[¶] haurogne.e2000935@etud.univ-ubs.fr

Table des matières

1 Résumé

Ce projet d'étude se concentre sur la création d'un analyseur d'articles scientifiques afin de simplifier la lecture des chercheurs de l'IRISA. Les articles scientifiques sont souvent au format PDF, qui n'est pas facilement portable ou analysable par les systèmes de traitement du langage naturel (NLP).

Notre analyseur d'articles scientifiques a donc pour objectif de générer une version plus portable, permettant aux systèmes de Traitement Automatique de Langues (TAL) de fonctionner correctement.

Nous extrayons et traitons le contenu de chaque PDF avec PyPDF2, identifiant les sections clés telles que l'abstract, les auteurs, et les discussions, tout en utilisant des techniques avancées comme la distance de Levenshtein pour l'analyse des e-mails et des affiliations, afin de les convertir avec le moins d'erreurs possibles en formats texte et XML. Pendant les cinq mois qu'a duré le projet, nous avons été en concurrence avec d'autres équipes de notre promotion. Notre équipe, **DABB**, a surpassé tous les autres groupes. Elle a été la plus performante et a récolté les meilleurs résultats parmi tous les groupes. Nous avons également dû rester vigilants quant à l'espionnage industriel, les dépôts publics GitHub des équipes d'Ewen (Resumax) et de Matthias (ParsePartout), nous ont permis de pouvoir déterminer et surveiller l'avancer des autres équipes.

Cet article détaillera notre approche, les outils utilisés et les résultats obtenus lors de l'évaluation de notre analyseur sur un ensemble varié de documents scientifiques, démontrant une précision globale supérieure à 99%. La discussion portera sur l'efficacité et les limites de notre analyseur, y compris ses performances dans différentes langues et son applicabilité à diverses disciplines scientifiques.

2 Introduction

Les chercheurs de l'Institut de Recherche en Informatique et Systèmes Aléatoires sont en constantes explorations et ne cessent de lire et d'analyser le contenu de nombreux articles scientifiques et revues académiques. Cette tâche, fastidieuse et chronophage, nous a amené à créer une solution simple afin d'automatiser l'analyse des articles scientifiques dans le but d'optimiser le temps des chercheurs.

Notre équipe a rencontré de nombreuses difficultés dans la transformation des articles au format PDF en format texte et XML. Le format PDF est largement utilisé dans la publication d'articles scientifiques. Ce format présente plusieurs obstacles à franchir comme la gestion de textes sur deux colonnes, les formules mathématiques, les figures (tableaux et images), etc. D'autre part, l'analyse automatique de ces documents par les systèmes de Traitement Automatique de Langues (TAL) est rendue complexe en raison d'une absence d'une structure universelle comprenant des standards.

En fournissant un aperçu clair des articles, notre analyseur permettra aux chercheurs de gagner du temps dans leur exploration de la littérature scientifique, tout en offrant de

nouvelles opportunités pour des analyses approfondies et des découvertes innovantes. Dans ce contexte, notre équipe propose un analyseur d'articles scientifiques pour faciliter l'analyse et l'exploitation de ceux-ci. Notre article expliquera en détail son approche pour résoudre les différents défis rencontrés, les outils utilisés ainsi que les résultats obtenus lors de l'évaluation de notre analyseur sur un ensemble de documents scientifiques.

3 Méthodologie

Nous allons ouvrir chaque PDF pour en extraire tout le contenu grâce à la librairie **PyPDF2** et le stocker dans une variable où nous allons pouvoir faire du traitement de texte. Pour ce qui est du traitement de texte, nous recherchons des mots clés afin de déterminer les différentes sections. L'objectif de notre analyseur est d'identifier ces différentes sections :

1. **Préambule :** Nous récupérons directement le nom du PDF.
2. **Titre :** On établit une fenêtre dans laquelle nous supposons que le titre s'y trouve. Si on ne le trouve pas, à cause d'un trop grand nombre d'éléments ou que ça finit par \n ou bien qu'il n'y a rien, alors on agrandit la fenêtre et continuons à itérer.
3. **Abstract :** On va chercher plusieurs mots clés classiques tels que "abstract", "keyword" ou encore "introduction". Une fois les mots localisés, on vient extraire le texte qu'il y a entre le mot "abstract" et le deuxième mot clé le plus proche suivant. Parfois, tous les mots ne sont pas présents. Par exemple, si le mot "abstract" est absent dans le PDF, nous partons de l'introduction, puis nous remontons jusqu'au début de l'abstract. Autre cas de figure, si "introduction" est également absent, nous cherchons le titre de la première section et extrayons le texte entre "abstract" et ce titre.
4. **Auteurs :** La récupération des auteurs est complexe en raison des différentes façons dont ils sont affichés dans les PDF. Il n'y a aucune façon commune pour récupérer les auteurs, nous avons donc développé un outil pour gérer ces variations. En règle générale, toutes les informations sur les auteurs se trouvent entre le titre et l'abstract (ou l'introduction s'il n'y a pas d'abstract), mais elles peuvent parfois se trouver ailleurs.

   Dans un premier temps, nous localisons la position du titre et de l'abstract et récupérons la section entre les deux. Ensuite, on vient retirer certains éléments inutiles comme "/natural" ou "1st", "2nd". Puis, nous utilisons des expressions régulières pour récupérer les adresses e-mail, ce qui nous donne plusieurs informations quant à la disposition des auteurs. Les e-mails possèdent plusieurs formats tels que :

   (a) Les e-mails sont uniques à chaque auteur et sont écrits en entier (par exemple `roller@cs.utexas.edu`).
   (b) Les noms sont écrits entre parenthèses ou accolades avec le nom de domaine à la suite
   (par exemple `{elvys.linhares-pontes, juan-manuel.torres, stephane.huet}@univ-avignon.fr`).
   (c) Les e-mails sont uniques, mais écrits ailleurs dans le PDF.
   (d) Les e-mails sont absents du PDF.

   Selon la façon dont les e-mails sont présentés, nous pouvons classer leur type de la manière suivante :

   (a) -1 : non trouvé (pas de mail),
   (b) 0 : normal (nom et mail),

(c) 1 : entre parenthèses ou accolades,
(d) 2 : normal, mais ailleurs dans le PDF et non au niveau des auteurs.

La disposition des e-mails dans le PDF nous aide à déterminer comment les affiliations et les noms des auteurs sont placés, ce qui nous permet d'en déduire le type de PDF.

Nous avons déduit plusieurs types de PDF selon la disposition des mails et des auteurs :

(a) -1 : non trouvé
(b) 0 : nom – université – mail pour chaque auteur
(c) 1 : nom sur une seule ligne – université – mail entre parenthèse ou accolade
(d) 2 : (nom – université) et mail autre part
(e) 3 : (nom – université) et pas de mail

Nous vérifions s'il y a des astérisques (*) car, sur certains PDF, les auteurs sont sur une ligne unique sans autre information.

Nous procédons ensuite à la récupération des auteurs en fonction de la présence des e-mails. Lorsqu'il y a zéro mail, on récupère les premières lignes jusqu'à tomber sur un mot que l'on retrouve en général dans les affiliations. S'il n'y a qu'un seul mail, il s'agit probablement d'une équipe, donc un seul mail pour tous (ou un seul auteur), et nous récupérons la première ligne. Enfin, si le type de PDF est zéro, il est probable que l'on se trouve dans le cas nom, affiliation, mail.
Pour finir, nous trions les informations récupérées, ajustons le type de PDF en fonction du nombre d'e-mails et d'auteurs, et si aucun auteur n'a été récupéré, nous prenons les premières lignes de la section des auteurs en ajustant le type du PDF.

Les premières étapes de récupération étant effectuées, les prochaines étapes consistent à séparer les auteurs et à créer le lien auteur-mail.

Pour la séparation des auteurs, on vient séparer les auteurs selon différents marqueurs comme des virgules (","), "and", etc. tout en vérifiant s'il y a des marqueurs liés à l'affiliation des auteurs (comme des chiffres ou des symboles). Nous effectuons un nettoyage afin d'enlever des chaînes de caractères vides, des espaces simples, etc.
Pour lier les auteurs à leurs e-mails, nous utilisons la distance de Levenshtein entre chaque auteur et le nom de l'e-mail avant le "@". Pour chaque auteur, nous récupérons l'e-mail avec la distance la plus faible. Si le nombre d'e-mails est égal au nombre d'auteurs et que les types de PDF et d'e-mails sont 0, il est alors probable que l'ordre soit conservé. Nous pouvons donc lier les auteurs directement. Si un seul e-mail est présent, il s'agit d'un e-mail d'équipe ou bien, il n'y a pas de mail.

La dernière étape consiste à récupérer l'affiliation pour chaque auteur en utilisant les informations disponibles.

D'abord, nous extrayons la section située entre le titre et l'abstract. Si le PDF est de type zéro, alors pour chaque auteur avec son mail, nous récupérons pour

chaque auteur son affiliation située entre son nom et son e-mail. Sinon, si le PDF est de type 1, nous commençons par supprimer les lignes des auteurs ainsi que d'autres éléments superflus. Puis, nous séparons chaque ligne et vérifions s'il y a des lettres minuscules au début de chaque ligne, signifiant un lien avec un ou plusieurs auteurs. Si des lettres minuscules sont présentes, nous les remplaçons par des symboles afin de traiter uniformément les cas de symboles ou de chiffres sans rajouter un troisième cas plus compliqué.

Si des auteurs sont associés à des symboles ou des chiffres, nous établissons les liens correspondants. Sinon, nous attribuons une affiliation commune à tous les auteurs.

Pour le dernier cas de figure (les PDF de type 2 ou 3), nous séparons chaque ligne d'affiliation, vérifions la présence de certains éléments comme les e-mails, puis attribuons une affiliation commune à chaque auteur.

Enfin, nous nettoyons les affiliations des éléments inutiles et, s'il n'y a pas eu d'affiliation trouvée, nous indiquons "N/A".

5. **Introduction :** Nous recherchons le mot clé "ndroduction" au lieu de "Introduction", car les variations entre "I" et "i" ainsi que les espaces éventuels entre les deux peuvent compliquer la détection. Le mot clé "ntroduction" est donc plus optimal. Ensuite, nous regardons si le titre de la section utilise un chiffre romain ou un chiffre arabe et s'il y a un point ou non afin de savoir comment sera numérotée la prochaine section. Enfin, nous repérons la position du titre de la section suivante et nous venons récupérer le texte entre les deux mots.
6. **Discussion :** Nous localisons des mots clés tels que "Discussion", "Conclusion", "References", "Appendix", etc. Ensuite, nous récupérons la position du mot, ("Discussion" dans ce cas précis) et le mot clé suivant dans le PDF. Enfin, nous extrayons le texte entre les deux termes.
7. **Conclusion :** Nous appliquons la même méthode que pour la discussion.
8. **Bibliographie :** Nous suivons le même procédé que pour la discussion, mais en récupérant tout le texte jusqu'à la fin du document.

Parallèlement au développement de l'analyseur, nous avons dès le début du projet créé nos propres tests en utilisant la méthode de Levenshtein. Cela nous a permis de suivre l'évolution et d'obtenir un aperçu de la précision de notre analyseur. Un affichage visuel des résultats a en plus été intégré au projet avec Colorama. De plus, nous avons réalisé une interface graphique directement dans le terminal grâce au module PyTermGUI.

4 Résultats

Les résultats obtenus ont été évalués sur un ensemble diversifié d'articles scientifiques provenant de différentes disciplines. Nous discuterons plus en détail de ces résultats dans la section suivante. L'évaluation s'est penché sur les sections clés telles que le résumé, l'introduction, la discussion, la conclusion et la bibliographie. Le titre, les auteurs, leur courrier électronique et leur affiliation ont, eux aussi, été pris en compte. La moyenne pour chaque article est supérieure à 99%.

	acl2012.xml	b0e5c09.xml	BLESS.xml	C14-1212.xml	Guy.xml
Précision (%)	99,56	99,73	99,74	99,86	99,49

	infoEmb.xml	IPM1481.xml	L18-1504.xml	OnMoral.xml	survey.xml
Précision (%)	99,49	99,62	99,95	99,70	99,37

Nous avons obtenu 100% sur l'ensemble des articles en ce qui concerne l'extraction des préambules et des titres. Plus de la moitié des abstracts sont à 100%. La note maximale a aussi été atteinte en ce qui concerne l'introduction et la conclusion de plusieurs articles. Nonobstant, les parties ayant les plus faibles résultats sont la bibliographie.
Sur le site officiel de notre Professeur Rémy Kessler, nous atteignons une moyenne de **99,651%**.

5 Discussion

La discussion se concentrera sur l'efficacité et la pertinence de notre analyseur d'articles scientifiques.
Nos performances sont comparées aux résultats d'autres équipes travaillant sur ce projet. Toutes les équipes sont soumises à une même contrainte : Les résultats sont comparés en fonction d'un site regroupant les articles au format XML. Le site compare donc notre résultat par rapport à la référence.
Nous pouvons souligner que la limite de notre analyseur d'articles scientifiques est la langue dans laquelle le chercheur va écrire son article. En effet, nous sommes restés dans le contexte des langues latines ainsi que l'anglais. Bien que la langue principale dans le monde de la recherche soit l'anglais, nous devons considérer que certains chercheurs pourraient l'écrire en mandarin ou en arabe par exemple. Ces langues ayant de nombreuses règles spécifiques, notre analyseur pourrait avoir des difficultés à correctement extraire les informations clés.

6 Conclusion

En conclusion, notre projet a abouti au développement d'un analyseur d'articles scientifiques efficace et simple à utiliser, capable d'extraire automatiquement les informations essentielles des articles au format PDF. Cette solution offre aux chercheurs un moyen rapide et précis d'accéder et d'analyser des articles scientifiques, tout en réduisant considérablement le temps nécessaire à cette tâche. Grâce à son potentiel d'automatisation et d'optimisation des processus d'analyse, notre analyseur couvre l'ensemble des demandes du laboratoire et ouvre la voie à une meilleure exploration des connaissances du monde scientifique.

7 Références

Librairie python utilisé :

1. PyPDF2 (3.0.1)
2. Levenshtein (0.25.0)
3. PyTermGUI (7.7.1)
4. colorama (0.4.6)