



CASIMIR : un Corpus d'Articles Scientifiques Intégrant les ModIfications et Révisions des auteurs

Léane Jourdan, Florian Boudin, Nicolas Hernandez, Richard Dufour

► To cite this version:

Léane Jourdan, Florian Boudin, Nicolas Hernandez, Richard Dufour. CASIMIR : un Corpus d'Articles Scientifiques Intégrant les ModIfications et Révisions des auteurs. Atelier sur l'Analyse et la Recherche de Textes Scientifiques, CORIA-TALN 2023, Jun 2023, Paris, France. hal-04103347

HAL Id: hal-04103347

<https://hal.science/hal-04103347>

Submitted on 24 May 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

CASIMIR : un Corpus d'Articles Scientifiques Intégrant les Modifications et Révisions des auteurs

Léane Jourdan¹ Florian Boudin¹ Nicolas Hernandez¹ Richard Dufour¹

Nantes Université, École Centrale Nantes, CNRS, LS2N, UMR 6004, F-44000 Nantes, France

prénom.nom@univ-nantes.fr

RÉSUMÉ

Écrire un article scientifique est une tâche difficile. L'écriture scientifique étant un genre très codifié, de bonnes compétences d'écriture sont essentielles pour transmettre ses idées et les résultats de ses recherches. Cet article décrit les motivations et les travaux préliminaires de la création du corpus CASIMIR dont l'objectif est d'offrir une ressource sur l'étape de révision du processus d'écriture d'un article scientifique. CASIMIR est un corpus des multiples versions de 26 355 articles scientifiques provenant d'OpenReview accompagné des relectures par les pairs.

ABSTRACT

CASIMIR : a Corpus of Scientific Articles Integrating the Modifications et Revisions of authors

Writing a scientific article is a challenging task as it is a highly codified genre. Good writing skills are essential to convey ideas and the results of research work properly. This paper describes the motivations and first steps of the CASIMIR creation process. The objective is to offer a new resource for text revision on the revision step of scientific article writing process CASIMIR is a corpus of the multiple versions of 26 355 scientific papers from OpenReview with their associated peer reviews.

MOTS-CLÉS : corpus, jeu de données, articles scientifiques, openreview, relectures, révision de textes.

KEYWORDS: corpus, dataset, scientific articles, openreview, reviews, text revision.

1 Introduction

Le processus d'écriture d'un article scientifique est une tâche complexe et difficile, particulièrement pour les jeunes chercheurs qui doivent apprendre les conventions de l'écriture scientifique. C'est à fortiori vrai pour les chercheurs non natifs anglophones qui doivent également faire face à la barrière de la langue. Juniors ou seniors, tous doivent prêter une attention particulière à la qualité de leur écriture afin de transmettre efficacement leurs idées au lecteur.

L'écriture scientifique est un genre à part avec ses propres codes et spécificités : structure de l'article (format IMRaD : *Introduction, Methods, Results and Discussion* (Swales, 1990)), style concis et précis, usages des temps, des pronoms et de la terminologie (Kallestinova, 2011; Bourekkache, 2022).

Plusieurs propositions ont été faites à travers la littérature pour décrire le processus d'écriture d'un article scientifique (Silveira *et al.*, 2022; Laksmi, 2006; Bailey, 2014; Seow, 2002; Du *et al.*, 2022a). Toutes ces propositions partagent des étapes communes et peuvent être résumées comme : *Étape 1 : Pré-écriture*, *Étape 2 : Brouillon*, *Étape 3 : Révision* et *Étape 4 : Relecture de finitions*. Le

présent corpus a vocation à être une ressource pour l'étape de *Révision* qui est réalisée en amont de la soumission par l'auteur lui-même, puis à la suite de la phase de relecture par les pairs, basée sur leurs suggestions. Elle consiste à faire des changements en profondeur dans le texte, sur le fond, la structure des phrases, la façon de connecter les idées. Cette étape est itérative (on la répète jusqu'à obtenir un résultat satisfaisant) ([Du et al., 2022a](#)) et 1-vers-N (une section de texte peut avoir plusieurs révisions correctes ([Ito et al., 2019](#))). Apporter une aide automatique à cette étape du processus d'écriture permettrait aux auteurs d'améliorer plus rapidement et efficacement leurs textes.

Il existe actuellement peu de corpus pour cette tâche dont les plus semblables à notre travail sont donnés ci-après. 1- PeerRead ([Kang et al., 2018](#)) est un corpus de 14 784 brouillons d'articles accompagnés de la décision de publication acceptée/rejetée. Ce corpus ne contient pas les versions finales des articles. 2- IteraTeR ([Du et al., 2022b](#)) un corpus de 31 631 documents (toutes les versions incluses) dont 11 443 résumés provenant de ArXiv, le reste provenant de Wikipedia et Wikinews. Il contient toutes les révisions pour un même résumé, alignées phrase à phrase, mais il n'inclut pas les articles dans leur intégralité. 3- arXivEdits ([Jiang et al., 2022](#)) composé de 751 articles provenant de ArXiv et leurs différentes versions, alignées phrase à phrase, pour un total de 1 790 documents.

Dans cet article, nous présentons la première étape de la création de CASIMIR, un corpus d'articles scientifiques en anglais accompagnés de leurs différentes versions et relectures faites par les pairs, collectés à partir de [OpenReview](#)¹. Il aura une taille supérieure aux précédents corpus et proposera un alignement au niveau des paragraphes en plus de l'alignement au niveau phrase. Cette nouvelle ressource pourra par exemple être mise à profit dans l'entraînement de modèles répondant aux diverses tâches de l'assistance à l'écriture scientifique.

la révision de textes ([Du et al., 2022a](#)), la correction orthographique, la prédiction d'acceptation/rejet d'un papier ([Kang et al., 2018](#)), etc.

2 Crédit du corpus

Les articles sont collectés depuis OpenReview, une plateforme ouverte de relecture par les pairs qui permet d'héberger les différentes versions d'un même article au format PDF ainsi que ses relectures. Son avantage est de proposer des premières versions peu relues et donc d'avoir des révisions plus importantes au fil des re-soumissions (exemple en annexe A). De plus, le contenu des relectures par les pairs est un guide sur la qualité des articles associés et les intentions sous-jacentes aux révisions effectuées. Toutefois, OpenReview présente un inconvénient majeur : les articles ne sont disponibles qu'en version PDF et non en format LaTeX et devront être convertis vers un autre format, ici le XML.

On considérera les termes suivants issus de la terminologie OpenReview :

- Un **forum** désigne l'espace de discussion et de dépôt attribué à un article.
- Une **relecture** désigne un commentaire écrit sur l'article : une relecture complète par un pair, une réponse de l'auteur ou du relecteur ou la décision finale sur la publication de l'article.

1. <https://openreview.net/>

2.1 Méthodologie de collecte et première étape de filtration

L'objectif est de collecter exhaustivement l'ensemble des documents disponibles sur OpenReview au 10/03/2023. Pour y parvenir, nous utilisons l'[API²](#) fournie par la plateforme.

Le processus de collecte est présenté ci-dessous :

1. Collecte de la liste des événements (ateliers, conférences, etc) hébergés sur la plateforme.
2. Collecte, grâce à cette liste, de l'ensemble des identifiants de forums liés à chaque événement.
3. Collecte, grâce aux identifiants des forums, des métadonnées associées aux versions d'un même article et création d'un fichier de correspondance associant l'identifiant du forum (identifiant du papier final) à l'identifiant de ses versions antérieures.
4. Collecte, grâce aux identifiants des forums, des relectures (messages sur le forum) et création d'un fichier de correspondance associant l'identifiant du forum et celui de ses relectures.
5. Collecte, des PDF disponibles sur le site des différentes versions des articles.

390 Go de données sont collectées dont 730 invitations et 121 492 PDF pour 29 504 articles.

2.2 Conversion des PDF

Les fichiers PDF ne sont pas directement utilisables pour l'entraînement de modèles, il est nécessaire de les convertir vers un format approprié, ici le XML. Pour extraire le contenu des fichiers PDF tout en conservant leur structure, on utilise l'outil état de l'art Grobid([GRO, 2008 2023](#)).

Après la conversion en XML, les citations, formules, figures et bibliographies seront retirées. Les fichiers PDF qui ne peuvent être convertis seront exclus du corpus et les articles qui n'ont plus qu'une seule version seront à nouveau filtrés.

Une première observation de la qualité de la conversion a été réalisée sur un sous-ensemble de documents déjà convertis. Ils comportent des erreurs telles que la mauvaise détection des tables et figures, la détection incorrecte de paragraphes en tant que figures, la suppression de portions de phrases, la mauvaise détection de sections, la retranscription des formules incluses dans les paragraphes, etc. Ces erreurs rendent l'alignement entre les différentes versions des articles plus complexe, car elles génèrent des différences supplémentaires entre les versions d'un article qui n'existent pas initialement.

3 Filtrage et description du corpus

Seuls les articles ayant au moins deux versions et dont les PDF et les métadonnées ont pu être collectés sont conservés. Le corpus résultant de cette première étape de filtrage comprend 26 355 articles et leurs versions antérieures (89.33% du nombre d'articles initial), pour un total de 118 415 documents (97.46% du nombre de documents initial). Il comprend également les métadonnées de chaque version

2. <https://openreview-py.readthedocs.io/en/latest/api.html>

et les relectures associées. 37 conférences sont représentées dans les données (hors soumissions et challenges indépendants). Parmi les domaines les plus représentés, nous retrouvons l'apprentissage automatique (ICLR, ICML, NeurIPS), la robotique (RSS, CoRL), le traitement automatique des langues (ACL), la vision par ordinateur (ECCV), etc.

La distribution du nombre de versions antérieures et de relectures par article est présentée dans la Figure 1. La majorité des articles ont moins de 10 versions. Toutes les soumissions de l'auteur sur la plateforme sont comptabilisées comme versions, plusieurs peuvent donc être effectuées avant même les relectures ou présenter des différences mineures. Pour les relectures, tous les échanges sur le forum lié à un article sont comptabilisés, expliquant le nombre élevé de relectures pour certains articles. Cependant, il est possible de différencier les "réelles" relectures des autres messages en utilisant les attributs des métadonnées.

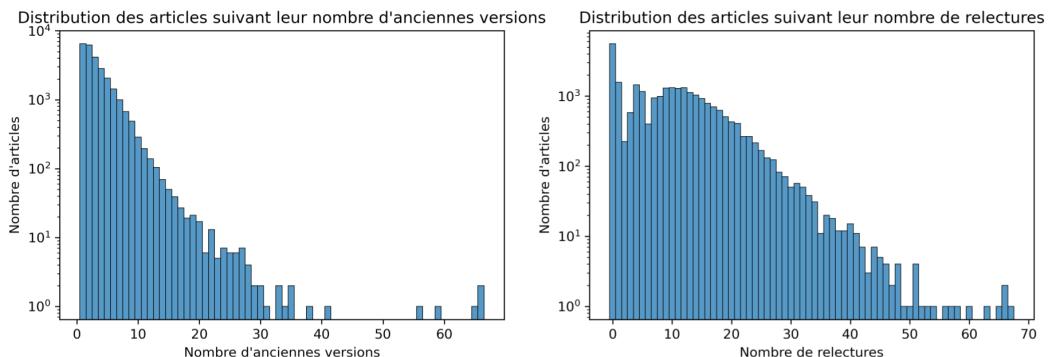


FIGURE 1 – Distribution du nombre de versions (gauche) et de relectures (droite) par article collectés

4 Discussion

Dans cet article, nous avons présenté la première étape de création du corpus de révisions d'article scientifiques CASIMIR. Lors de ces travaux préliminaires, plusieurs problèmes ont été rencontrés. Tout d'abord, des difficultés liées à OpenReview, certains fichiers PDF étaient manquants et certaines conférences ne contenaient aucun papier. Le corpus a donc été limité aux données disponibles au moment de la collecte.

Une autre difficulté rencontrée concerne la conversion des PDF évoquée en Section 2.2. Pour pallier ce problème, trois articles ont été sélectionnés aléatoirement, avec leur première et dernière version, soit six documents. Les versions XML générées par Grobid ont été corrigées manuellement pour ces six documents, avec un temps approximatif de 75 minutes par article. Ces articles annotés manuellement serviront de références pour évaluer la qualité de la conversion en XML générée par Grobid, ainsi que la dégradation de la qualité de l'alignement automatique sur les fichiers convertis automatiquement.

Pour poursuivre la création du corpus, l'ensemble des documents doit être converti en XML, puis les différentes versions des articles doivent être alignées paragraphe à paragraphe et phrase à phrase. Pour cela, nous pourrons nous reposer sur le modèle d'alignement de phrases proposé par (Jiang

et al., 2022) ainsi que leur algorithme d’alignement des paragraphes en l’améliorant pour tenir compte des fusions et divisions de paragraphes. Cela permettra d’extraire les révisions entre les différentes versions d’un article. Les trois articles de référence corrigés manuellement seront également alignés manuellement.

Enfin, les documents seront annotés selon une taxonomie de révisions à définir (exemples : clarté, grammaire, langage, style, etc) pour être utilisés pour l’entraînement de modèles de révision de texte.

Références

- (2008–2023). Grobid. <https://github.com/kermitt2/grobid>.
- AGLIONBY G. & TEUFEL S. (2022). Identifying relevant common sense information in knowledge graphs. In *Proceedings of the First Workshop on Commonsense Representation and Reasoning (CSRR 2022)*, p. 1–7, Dublin, Ireland : Association for Computational Linguistics. DOI : [10.18653/v1/2022.csrr-1.1](https://doi.org/10.18653/v1/2022.csrr-1.1).
- BAILEY S. (2014). *Academic writing : A handbook for international students*. Routledge.
- BOUREKKACHE S. (2022). English for specific purposes : writing scientific research papers. case study : Phd students in the computer science department. Mémoire de master, University of Biskra, Algeria.
- DU W., KIM Z. M., RUNDERSTANDAHEJA V., KUMAR D. & KANG D. (2022a). Read, revise, repeat : A system demonstration for human-in-the-loop iterative text revision. In *Proceedings of the First Workshop on Intelligent and Interactive Writing Assistants (In2Writing 2022)*, p. 96–108, Dublin, Ireland : Association for Computational Linguistics. DOI : [10.18653/v1/2022.in2writing-1.14](https://doi.org/10.18653/v1/2022.in2writing-1.14).
- DU W., RAHEJA V., KUMAR D., KIM Z. M., LOPEZ M. & KANG D. (2022b). Understanding iterative revision from human-written text. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 3573–3590, Dublin, Ireland : Association for Computational Linguistics. DOI : [10.18653/v1/2022.acl-long.250](https://doi.org/10.18653/v1/2022.acl-long.250).
- ITO T., KURIBAYASHI T., KOBAYASHI H., BRASSARD A., HAGIWARA M., SUZUKI J. & INUI K. (2019). Diamonds in the rough : Generating fluent sentences from early-stage drafts for academic writing assistance. In *Proceedings of the 12th International Conference on Natural Language Generation*, p. 40–53, Tokyo, Japan : Association for Computational Linguistics. DOI : [10.18653/v1/W19-8606](https://doi.org/10.18653/v1/W19-8606).
- JIANG C., XU W. & STEVENS S. (2022). arxivedits : Understanding the human revision process in scientific writing. In *Proceedings of EMNLP 2022*.
- KALLESTINOVA E. D. (2011). How to write your first research paper. *The Yale journal of biology and medicine*, **84**(3), 181.
- KANG D., AMMAR W., DALVI B., VAN ZUYLEN M., KOHLMEIER S., HOVY E. & SCHWARTZ R. (2018). A dataset of peer reviews (PeerRead) : Collection, insights and NLP applications. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long Papers)*, p. 1647–1661, New Orleans, Louisiana : Association for Computational Linguistics. DOI : [10.18653/v1/N18-1149](https://doi.org/10.18653/v1/N18-1149).
- LAKSMI E. D. (2006). " scaffolding" students'writing in efl class : Implementing process approach. *TEFLIN Journal*, **17**(2), 144–156.

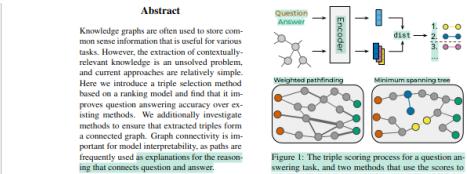
SEOW A. (2002). The writing process and process writing. *Methodology in language teaching : An anthology of current practice*, 315, 320.

SILVEIRA E. A., DE SOUSA ROMEIRO A. M. & NOLL M. (2022). Guide for scientific writing : how to avoid common mistakes in a scientific article. *Journal of Human Growth and Development*, 32(3), 341–352.

SWALES J. M. (1990). *Genre Analysis : English in academic and research settings*. The Cambridge applied linguistics series. The press syndicate of the University of Cambridge.

A Exemple des différences entre deux versions d'un même article issues de OpenReview

Abstract	
001	Knowledge graphs are often used to store com- 002 mon knowledge and for solving various tasks. 003 However, the extraction of contextually- 004 relevant knowledge is an unsolved problem, 005 and current approaches are relatively simple. 006 Here we introduce a triple selection method 007 based on a ranking model and show that it im- 008 proves question answering accuracy over ex- 009 isting methods. We additionally investigate 010 methods for which extracted triples form 011 a connected graph. Graph connectivity is es- 012 sential for model interpretability, as paths are 013 frequently used to understand reasoning from 014 question to answer. We make our code and 015 data available at https://github.com/ .
016	answering
1 Introduction	
018	For models to be able to reason about situations 019 that arise in everyday life, they must have access to 020 contextually appropriate common sense informa- 021 tion. This information is commonly stored as 022 a large set of facts from which the model must 023 identify relevant ones. In this paper, we start char- 024 terizing this facts as a knowledge graph. Here, 025 nodes represent high-level concepts, and relation- 026 ships are expressed via typed edges joining two 027 nodes. Each edge type represents a different kind 028 of conceptual relationship between concepts. The 029 concept of schema graph (Schemagraph) is a graph 030 that are extracted from these graphs. They are often 031 encoded using neural models, which are trained for 032 tasks including question answering or natural lan- 033 guage inference.
034	Prior work has focused on different ways to en- 035 code this information, including using it as input 036 directly to a model or using a graph neural network 037 (GNN) (Feng et al., 2020; Li et al., 2021; Liu et al., 038 2021). However, the question of how to identify 039 useful information has been under-explored, par- 040 ticularly in work that uses GNN encoders. This
simple retrieval methods that are used could limit 041 performance on tasks where contextually important in- 042 formation is not retrieved.	
043	In this paper we explore methods to con- 044 struct high-quality schema graphs containing 045 contextually-relevant information. We approach this 046 as a ranking task across triples in a knowledge 047 graph, and select the highest scoring for the schema 048 graph. However, simply using the most relevant 049 triples as edges in a GNN does not work as well as 050 resulting subgraphs tend to have low connectivity. 051 This is problematic for two reasons. First, it 052 inherently limits the power of the GNN, as nodes 053 in different graph components will not be updated 054 with information from each other. Second, paths 055 of reasoning through the schema graphs are often 056 used in explanations for model behaviour (Feng 057 et al., 2020; Li et al., 2021; Wang et al., 058 2020). If the graph consists of multiple separate 059 components, this becomes impossible.
060	The issue of graph disconnectedness is com- 061 pounded when certain nodes are required to be 062 included. For example, in question answering, con- 063 cepts mentioned in the question and in a candidate 064 answer should be identified and included. A path 065 starting from a question concept and then evaluated 066 against a plan of reasoning is an aggressive 067 search for answer concepts. We therefore apply a 068 graph algorithm to ensure that the schema graph is 069 connected, taking into account the identified edges 070 and desired nodes, and use an embedding-based 071 method to identify concepts mentioned.
072	Our contributions are summarised as follows:
073	• Apply a ranking model to identify common 074 sense triples that are in the same context. 075 Identify and thoroughly investigate methods 076 to ensure schema graph connectivity.
077	• Compare existing lexical approaches to entity linking to a simple embedding-based method.



Abstract

Knowledge graphs are often used to store common sense information in order to solve various tasks. However, the extraction of contextually-relevant knowledge is an unsolved problem, and current approaches are relatively simple. Here we introduce a triple selection method based on a ranking model and show that it improves question answering accuracy over existing methods. We additionally investigate methods for which extracted triples form a connected graph. Graph connectivity is essential for model interpretability, as paths are frequently used to understand reasoning from question to answer. We make our code and data available at <https://github.com/>.

1 Introduction

For models to be able to reason about situations that arise in everyday life, they must have access to contextually appropriate common sense information. This information is commonly stored as a large set of facts from which the model must identify a relevant subset. One approach to structuring these facts is as a knowledge graph. Here, nodes represent high-level concepts, and relationships are of different kinds of relationship between concepts. In practice, a subset of facts that are thought to be contextually relevant are extracted from the graph as using all facts in each instance is unnecessary, noisy, and computationally expensive.

Prior work has focused on different ways to encode this information, including using it as input directly to a model or using a graph neural network (GNN) (Feng et al., 2020; Li et al., 2021). However, the question of how to identify useful information has been under-explored, particularly in work that uses GNN encoders. If contextually important information is not selected then performance could be diminished or reduced. The final result of the use of overly simplistic retrieval methods.

In this paper we explore methods to extract high-quality subgraphs containing contextually relevant

information. We approach this as a ranking task across triples in a knowledge graph, and propose two methods that use the scores to extract a subgraph. The first is a weighted pathfinding approach which extends prior work (Lin et al., 2019), while the second builds a minimum spanning tree that connects all nodes in the graph.

Both approaches ensure that all or most nodes in the subgraph are reachable from each other, which is important for two reasons. First, it means that the GNN can update node embeddings with information from most other nodes, which would not be possible if the graph was disconnected. Second, it allows paths of reasoning to be extracted from the subgraphs, which are often used as explanations for model behaviour (Feng et al., 2020; Wang et al., 2020; Yasunaga et al., 2021).

There are also situations when specific concepts need to be included in order for a subgraph to be of high enough quality. For example, in question answering, a full explanation must include one

"We call these "relevant subgraphs" or "extracted subgraphs", noting that others use "schema graphs" (Lin et al., 2019).

FIGURE 2 – Visualisation des différences entre la première, la dernière version de (Aglionby & Teufel, 2022)