LLM-Resilient Bibliometrics: Factual Consistency Through Entity Triplet Extraction*

Alexander Sternfeld[1], Andrei Kucharavy[2], Dimitri Percia David[2], Alain Mermoud[1] and Julian Jang-Jaccard[1]

[1]Cyber-Defence Campus, armasuisse, Science and Technology, Thun, Switzerland
[2]Institute of Entrepreneurship Management, HES-SO Valais-Wallis

Abstract

The increase in power and availability of Large Language Models (LLMs) since late 2022 led to increased concerns with their usage to automate academic paper mills. In turn, this poses a threat to bibliometrics-based technology monitoring and forecasting in rapidly moving fields. We propose to address this issue by leveraging semantic entity triplets. Specifically, we extract factual statements from scientific papers and represent them as *(subject, predicate, object)* triplets before validating the factual consistency of statements within and between scientific papers. This approach heavily penalizes blind usage of stochastic text generators such as LLMs while not penalizing authors who used LLMs solely to improve the readability of their paper. Here, we present a pipeline to extract such triplets and compare them. While our pipeline is promising and sensitive enough to detect inconsistencies between papers from different domains, the intra-paper entity reference resolution needs to be improved to ensure that triplets are more specific. We believe that our pipeline will be useful to the general research community working on the factual consistency of scientific texts.

Keywords

Bibliometrics, Entity Extraction, Machine Learning, Technological Forecasting, Quantum Computing

1. Introduction

For firms to make informed investment decisions, sound forecasts on the development of technologies are necessary. One prominent method for technology forecasting is bibliometrics, which uses the information in scholarly books and journals [1]. Modern bibliometric methods leverage the increase in available data by applying machine-learning methods. For example, Percia David et al. (2023) analyse arXiv pre-prints to evaluate the security development of information technologies [2].

While scientific publications are thus increasingly more important for technology forecasting, the quality of the papers must be evaluated critically. The publish-or-perish pressure led to a record growth in the number of scientific publications per author, often with minimal peer review [3, 4]. In such a setting, if LLMs can generate text that sufficiently resembles a scientific article to pass for one on a cursory reading, they are likely to be used to generate scores of articles. Unfortunately, this eventuality is already likely to be a reality, given that Majovsky et al. (2023) showed that ChatGPT can create an authentic-looking neurosurgery scientific article [5].

Recently, there has been a growing interest in identifying text generated by LLMs. As early as 2019, Zellers et al. showed that a GPT2-like LLM *Grover* could detect its own output. However, recent research suggests that, in general, LLM detectors either do not work or are easy to evade [6, 7]. Overall, for a minimally competent attacker who wants to evade detection, LLM detectors cannot be relied upon.

Unfortunately, the situation is serious enough for some of the most reputable providers of proxies of the impact of scientific articles to have modified their algorithms to only consider publications adhering to stringent criteria [8]. Due to the velocity of innovation and the reliance on preprint repositories, such an approach is not adapted to technology monitoring in the domains adjacent to cybersecurity and machine learning. Because of these factors, we investigate if factual consistency could be used for LLM-resilient bibliometrics instead.

Specifically, we represent facts as entity triplets of the form *(subject, predicate, object)* that are extracted from the claims of the paper. The entity triplet plays a crucial role as it serves as a proxy to understand the primary claims of the paper and subsequently validates factual consistency compared to other works in the domain. Our paper describes the workflow involved in entity triplet extraction and provides an overview of our initial findings regarding the effectiveness of the entity triplets and their relation to the number of clusters generated around the subject. The code of this project is available at https://github.com/technometrics-lab/0-Factual_Consistency_Through_Entity_Triplets, at commit c7b01e4.

2. Related work

Previous approaches for claim extraction can be categorized into heuristics and machine learning methods. The advantage of an approach based on heuristics is that no training data is required and the computational cost tends to be low. However, machine learning approaches can capture more complex patterns, leading to the extraction of triplets of higher quality. Such methods have been developed most prominently in the biomedical domain. For example, Li et al. (2021) use BiLSTMs to extract the factual statements presented in papers [9]. Although less labeled data is available, there has been work focusing on claim extraction from papers in different domains. For instance, Binder et al. (2022) use BiLSTMs for argumentative discourse unit recognition and argumentative relation extraction [10].

The majority of existing triplet extraction models use supervised training. Two notable examples are RECON and sPERT, which require labeled training data [11, 12]. The disadvantage of supervised methods is the need for training data and the dependency on the relations that are present in the dataset. In contrast, unsupervised methods do not need training data and use either heuristics or machine learning methods to extract triplets. One example of such a model is

Joint Workshop of the 5th Extraction and Evaluation of Knowledge Entities from Scientific Documents and the 4th AI + Informetrics (EEKE-AII2024), April 23-24, 2024, Changchun, China and Online
alex1.sternfeld@gmail.com (A. Sternfeld)

Stanford OpenIE, which extractsrelational tuples without the need to specify a schema in advance. However, it has been shown that OpenIE tends to extract too aggressively, resulting in the presence of non-useful relations [13]. We contribute by providing a method that can provide triplets from scientific papers with a high precision, while the user only needs to specify the desired research categories.

3. Methodology

The process of extracting informative triplets from raw PDFs consists of four main stages. First, PDFs are converted to text files, after which they are preprocessed to remove word breaks and citations, expand abbreviations and lemmatize words. Then, we extract the sentences from the paper that convey its core ideas, which we refer to as *claims*. From these claims, we can extract subject-predicate-object triplets. The last step is to process these triplets further so that they can be used in a comparative analysis. The entire pipeline is displayed in Figure 1. In the following subsections, we elaborate on each of the steps. While we focus on arXiv, the approach is generalizable to all scientific PDFs.

3.1. Preprocessing

To convert the pdf to text we use the PyMuPDF library [14]. We then further clean the text by removing bracketed citations and merging words that were split due to line breaks. We then expand abbreviations by using a rule-based algorithm introduced by Schwartz and Hearst (2003) [15]. In appendix 6.1 we show that the Schwartz-Hearst algorithm outperforms the scispaCy [16] and NLPRe [17] abbreviation detection methods, which are built on spaCy. Moreover, due to the rule-based nature of the algorithm, it is relatively fast. The example below shows the transition from a raw sentence to a preprocessed result.

Table 1
Illustration of the preprocessing steps, the parts in bold are altered during preprocessing.

Uncleaned	Preprocessed
Society has been affected by **artificial intelligence (AI)** and has become more **rel- iant** on **AI** products.	Society has been affected by **artificial intelligence** and has become more **reliant** on **artificial intelligent** products.

3.2. Claim and triplet extraction

After preprocessing the text, we identify the sentences that convey the authors' claims. Specifically, we use the *ClaimDistiller* framework developed by Wei et al. (2023) [18]. In their work, both CNN's and BiLSTM's are trained for claim extraction on the PubMED-RCT and SciARK datasets [19, 20]. Although the usage of supervised contrast training improves the performance of the model, it causes a computational overhead. We use the BiLSTM without supervised contrast learning to strike a balance between performance and computational efficiency. We choose to extract the claims from the papers such that the subsequent triplet extraction will have to be performed on fewer sentences.

Next, we want to reduce the claims to (*subject, predicate, object*) triplets, analogous to the Resource Description Framework (RDF) format commonly used in the representation of OWL ontologies. We choose this representation, as it will facilitate the comparison of claims across papers. We use the Python library `textacy`, which is built on spaCy [21] and has a built-in extraction method that does not require the specification of relations in advance.

3.3. Post-processing of triplets

Our goal is to compare triplets across papers. Therefore, we further process the triplets so that we can pair triplets from different papers that refer to the same subject. The following steps are followed:

1. Lowercase all words in the triplet
2. Remove triplets where either the subject or object contains more than 6 words
3. Remove stopwords from the triplets based on the list included in NLTK
4. Remove any character that is not text
5. Lemmatize verbs and nouns in the triplet
6. Remove words containing less than 3 characters
7. Filter the triplets non-specific to scientific work by comparison with a general book corpus
8. Filter the triplets characteristic of scientific works in general by comparison with arXiv articles from different categories

In the second step, we choose this cutoff, as we expect that phrases of over 6 words may contain nuances that cannot be captured in a simple subject-predicate-object relation. Words with less than 3 characters are removed, as we observed that such words were often noise. Moreover, as abbreviations are expanded we expect all informative terms to be at least of length 3.

We use the general-purpose Gutenberg book corpus to filter the triplets that carry little information. We define the number of times term i appears at least 5 times in a document in the book corpus and in the paper corpus as $f_{b,i}$ and $f_{p,i}$, respectively. We then assign a score s_i to each term i:

$$s_i = \begin{cases} -\infty & f_{p,i} < 10 \\ log(\frac{f_{p,i}}{N_p}) - log(\frac{f_{b,i}}{N_b}) & \text{if } f_{p,i} \geq 10 \text{ and } f_{b,i} > 0 \\ \infty & \text{if } f_{b,i} = 0 \text{ and } f_{p,i} \geq 10 \end{cases}$$

Terms that are not present in at least 10 papers, thus get a score of $-\infty$. If the term is present in at least 10 papers, the score increases when the frequency of the term in the book corpus is lower. We keep the triplets with subjects in the top 10% of the term scores.

In the last step, we aim to keep only the triplets that carry domain-specific information. Therefore, we sample a random subset of 1000 arXiv papers from December 2023 from different categories than our target papers. We then only keep the triplets with subjects present in a maximum of 15 papers.

3.4. Clustering

As we use the extracted triplets to compare papers, it is necessary to cluster them based on the subject and object. Both SciBERT [22] encodings and spaCy [21] embeddings were considered. Based on visual inspection, the resulting clusters are most coherent when using SciBERT, which is a language model based on BERT [23], pretrained on a large multi-domain corpus of scientific publications. After

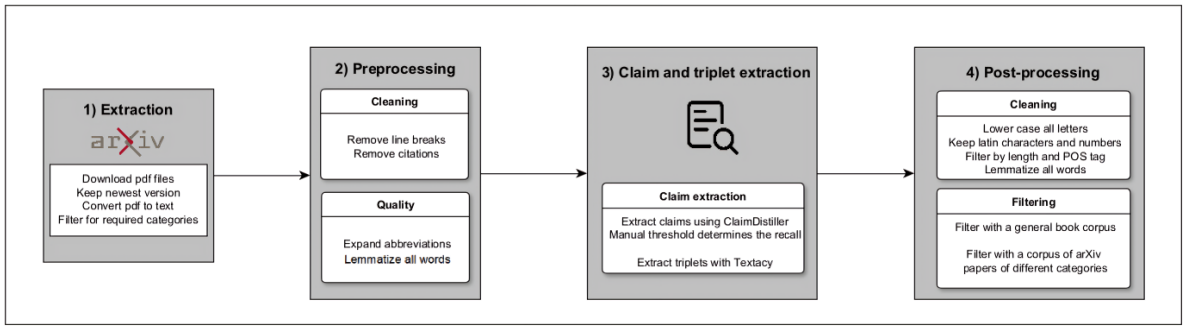

Figure 1: The complete pipeline for extracting entity triplets from the raw articles from the arXiv archive

encoding both the subjects and objects, we utilize an agglomeration hierarchical clustering algorithm from Scipy [24], which compares the average distance between clusters. By visually inspecting the dendograms, we set a cutoff to obtain the final clusters. Figure 5 in the appendix shows an example of the dendograms for a subset of the subjects and objects. The threshold is chosen at the height where the distance between clusters begins to noticeably increase.

3.5. Triplet comparison

After clustering, we compare the triplets within the same cluster based on the predicates. We take a first step in this direction by analysing embedding inversions, as simple vector arithmetic can provide valuable insights into word relationships, such as negation or gender variants [25]. Specifically, we subtract the spaCy embeddings of the predicates and study the tokens closest to the resulting vector. Although SciBERT encodings likely contain more semantic information, the input sequence is embedded along with its context, hence it cannot be easily inverted to a token.

4. Results

4.1. Data

We consider two different datasets to evaluate our method. First, to leverage in-house expertise in the domain of computer science and natural language processing (NLP), we focus on publications relevant to the domain and retrieve data from the arXiv categories `cs.AI`, `cs.CL` and `cs.LG`. Specifically, we retrieve all papers from December 2023, which amounts to a total of 4225 research articles.

Second, for a quantitative analysis of the usefulness of triplets for factual consistency evaluation, we consider two surveys. To validate our approach in an independent domain, we considered both a survey on LLMs and a survey on Quantum Computing. Specifically, we analyse a survey on LLMs by Zhao et al. (2019) [26] and a survey on quantum computing technologies by Gyongyosi and Imre (2019) [27]. We construct a dataset comprising these two surveys and the arXiv preprints cited by these surveys. We limit ourselves to the papers for which an arXiv ID was provided in the references of the survey, leading to a total of 188 papers. In the subsequent section, we refer to the papers related to the LLM survey as the *LLM data* and to the papers related to the quantum computing survey as the *quantum data*.

4.2. Cluster analysis

4.2.1. CS articles December 2023

After preprocessing the articles, we extract the triplets. For the triplet extraction, we used the hyperparameters displayed in Table 5 in the appendix. In total, 79,986 triplets were extracted from the research articles. We cluster these triplets based on both the subject and object embeddings, which resulted in 37,076 clusters. Figure 3 in the appendix shows the distribution of the number of triplets per cluster. This shows that most clusters contain less than 25 triplets, but that there are outliers that contain over 200 triplets.

4.2.2. LLM and quantum computing surveys

For a more in-depth analysis, we consider the triplets extracted from the LLM and quantum computing surveys and their cited papers. In total, 1895 triplets are extracted from the 188 papers. We cluster these triplets based on the subjects so we can evaluate the differences between the objects in a cluster. Figure 4 in the appendix shows that larger clusters often have triplets from multiple categories, whereas small clusters tend to have triplets from only one category.

Next, we make pairwise comparisons between the object embeddings within a cluster. Figure 2 shows the pairwise distances (L2 norm) between objects for each cluster size. We find that for clusters below size 8, the distance between objects from the LLM data and quantum data is larger than the distances of objects within a category. For larger cluster sizes, this effect disappears. This indicates that for smaller clusters with triplets from both categories, the objects are more diverse. Furthermore, we see that for clusters with triplets from one category, the distance between objects increases for larger sizes. This confirms that larger clusters are more domain-agnostic and contain more varied objects.

Table 2 shows manually selected clusters with sizes 2, 4 and 8. The column with mixed data clusters shows clusters that contain both triplets from the LLM data and the quantum data, where the triplets from the quantum data are displayed in italics. For smaller clusters, the triplets within a dataset tend to be similar and vary only slightly. In contrast, the objects differ more for the mixed data clusters as they are domain-specific. On the other hand, the results suggest that larger clusters contain more domain-agnostic subjects and objects, such as *lab*. Consequently, the distance between the objects from different datasets differs less than between objects from the same data. Further manual inspection supports this hypothesis with large clusters containing subjects such as *appendix* and *conclusion*.

Table 2
Manually selected clusters of sizes 2, 4 and 8. In the clusters with mixed data, the italic triplets belong to the quantum data and the regular triplets to the LLM data.

Cluster size	Pure LLM cluster	Pure quantum cluster	Mixed data cluster
2	(expert knowledge, enhance, data utility), (expert knowledge, employed, data utility)	(bundle map, determine, representation), (tangent map, analyse, map)	(minimizer, satisfies, argmin), *(minimizer, given, tetrahedron)*
5	(reinforcement learning, requires, language model), (reinforcement learning, based, sampling algorithm), (reinforcement learning, learns, reward model), (reinforcement learning, offer, evaluation), (reinforcement learning, inherits, drawback training instability)	(complement, express, failure), (complement, contains, function), (complement, must contain, part), (complement, not capture, complement), (complement, not represented, set)	(time duration, cover, feature), (spn value, shown, symbol), (duration value, have, spacing), *(dene value, introduce, reduction relation),* *(dene value, dene, progress relation)*
8	(neuron, receives, impulse), (neuron, displayed, difference), (neuron, reaching, average rate), (neuron, not, impact), (neuron, coupled, weight wji), (neuron, changed, activity), (neuron, described, pair), (neuron, sends, impulse), (neuron, make, decision)	(alice protocol, avoids, computing requirement), (alice protocol, ha, difference), (alice protocol, requires, bob), (alice protocol, consumes, network bandwith), (alice protocol, reduce, quantum computation), (alice protocol, ha, advantage), (alice protocol, provides, fault tolerance), (alice protocol, preserve, tolerance ability)	(lab, improved, learning), (lab, offered, value), (lab, had, benefit), *(lab, offer, introduction),* *(lab, present, environment),* *(lab, improve, performance),* *(lab, demonstrates, concept),* *(lab, outline, qml solution)*

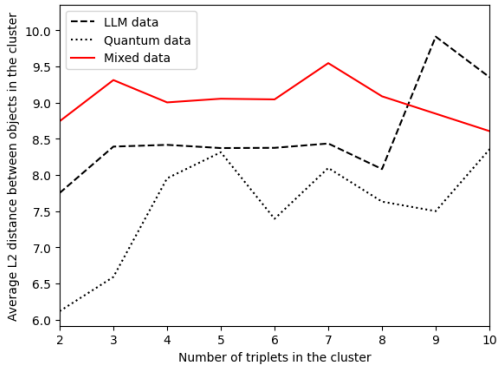

Figure 2: The average distance (L2 norm) between objects of triplets for different cluster sizes

Overall, we argue that it means that triplets as extracted by our pipelines can be used as proxies for factual consistency, but that additional refinement is needed to avoid extracting overly generic statements.

4.3. Factual consistency

4.3.1. Predicate comparisons

As a first step in evaluating the factual consistency between papers, triplets in the same cluster are compared based on the predicates. Specifically, two triplets are considered consistent when the predicates are synonyms, hypernyms or hyponyms. If the predicates are antonyms, the triplets are considered inconsistent. The VerbOcean and WordNet databases are used to label pairs of predicates [28, 29].

Table 3
Number of consistent and inconsistent triplet pairs.

	Inconsistent triplet pairs	Consistent triplet pairs
Across papers	2044	434362
Within papers	175	4549

Table 3 shows that the majority of triplets within a cluster are consistent. We see both within and across papers that there are inconsistent triplets present. However, manual inspection shows that, in some cases, an inconsistent pair of triplets can be caused by differing contexts. For example, the pair *(initialization, impede, optimization process)* is marked to be inconsistent with *(initialization, accelerate, optimization process)*. This again shows that refinement is needed to extract more specific triplets.

4.3.2. Embedding inversion

To do a more qualitative assessment, we invert the differences of the embeddings of predicates from the same cluster. Table 6 in the appendix shows a manual selection of 9 of these embedding inversions. The results show that an embedding inversion does not provide informative results in this context. In general, we do not find that there is a noticeable difference between embedding inversions of predicates that are consistent or inconsistent.

5. Conclusion

This paper presents an unsupervised method for the extraction of triplets from scientific work. Whereas previous methods either require labeled training data or the prespecification of entity relations, we allow for entity triplet extraction through domain specification.

The results show that the extracted triplets accurately reflect the domain from the corresponding scientific work. When we cluster triplets based on the subject, we find that smaller clusters tend to be domain-specific. In contrast, larger clusters are more generic and often contain triplets from different domains. We interpret it as extracted triplets being suitable for evaluating factual consistency, but requiring further refinement for a more specific extraction. We believe this is due to insufficient resolution of excessively general nouns (e.g. *lab, conclusion*). To compare the triplets, an embedding inversion was implemented on the difference of the verb embeddings for similar triplets. Our findings show that an embedding inversion does not allow us to discriminate between consistent and inconsistent triplets.

Our results suggest that the next steps for the usage of the extracted triplets for the development of LLM-resilient proxies should focus on better filtering of domain-agnostic subjects, for them to be informative about factual consistency. Then, a semantic network can be built based on the similarities between the triplets for the entirety of the scientific publications in a domain of interest. By leveraging this network, we can identify papers that are factually inconsistent or excessively consistent and use the remainder of the corpus for a bibliometric analysis.

References

[1] Y. Zhang, A. L. Porter, S. Cunningham, D. Chiavetta, N. Newman, Parallel or intersecting lines? intelligent bibliometrics for investigating the involvement of data science in policy analysis, IEEE Transactions on Engineering Management 68 (2020) 1259–1271.

[2] D. Percia David, L. Maréchal, W. Lacube, S. Gillard, M. Tsesmelis, T. Maillart, A. Mermoud, Measuring security development in information technologies: A scientometric framework using arxiv e-prints, Technological Forecasting and Social Change 188 (2023) 122316. URL: https://www.sciencedirect.com/science/article/pii/S004016252300001X. doi:https://doi.org/10.1016/j.techfore.2023.122316.

[3] M. A. Hanson, P. G. Barreiro, P. Crosetto, D. Brockington, The strain on scientific publishing, CoRR abs/2309.15884 (2023). URL: https://doi.org/10.48550/arXiv.2309.15884. doi:10.48550/ARXIV.2309.15884. arXiv:2309.15884.

[4] J. P. A. Ioannidis, R. Klavans, K. W. Boyack, Thousands of scientists publish a paper every five days, Nature 561 (2018) 167 – 169. URL: https://api.semanticscholar.org/CorpusID:52198631.

[5] M. Majovsky, M. Černý, M. Kasal, M. Komarc, D. Netuka, Artificial intelligence can generate fraudulent but authentic-looking scientific medical articles: Pandora's box has been opened, Journal of Medical Internet Research 25 (2023). doi:10.2196/46924.

[6] C. Chen, K. Shu, Can llm-generated misinformation be detected?, 2023. arXiv:2309.13788.

[7] D. S. G. Henrique, A. Kucharavy, R. Guerraoui, Stochastic parrots looking for stochastic parrots: Llms are easy to fine-tune and hard to detect with other llms, 2023. arXiv:2304.08968.

[8] Clarivate, 2024 journal citation reports, https://clarivate.com/blog/2024-journal-citation-reports-changes-in-journal-impact-factor-category-rankings-to-enhance-transparency-and-inclusivity/, 2024. Accessed: 2024-02-29.

[9] X. Li, G. A. Burns, N. Peng, Scientific discourse tagging for evidence extraction, in: P. Merlo, J. Tiedemann, R. Tsarfaty (Eds.), Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021, Association for Computational Linguistics, 2021, pp. 2550–2562. URL: https://doi.org/10.18653/v1/2021.eacl-main.218. doi:10.18653/V1/2021.EACL-MAIN.218.

[10] A. Binder, B. Verma, L. Hennig, Full-text argumentation mining on scientific publications, CoRR abs/2210.13084 (2022). URL: https://doi.org/10.48550/arXiv.2210.13084. doi:10.48550/ARXIV.2210.13084. arXiv:2210.13084.

[11] A. Bastos, A. Nadgeri, K. Singh, I. O. Mulang, S. Shekarpour, J. Hoffart, M. Kaul, Recon: Relation extraction using knowledge graph context in a graph neural network, in: Proceedings of the Web Conference 2021, WWW '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 1673–1685. URL: https://doi.org/10.1145/3442381.3449917. doi:10.1145/3442381.3449917.

[12] M. Eberts, A. Ulges, Span-based joint entity and relation extraction with transformer pre-training, in: G. D. Giacomo, A. Catalá, B. Dilkina, M. Milano, S. Barro, A. Bugarín, J. Lang (Eds.), ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020), volume 325 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2020, pp. 2006–2013. URL: https://doi.org/10.3233/FAIA200321. doi:10.3233/FAIA200321.

[13] L. Liu, A. Omidvar, Z. Ma, A. Agrawal, A. An, Unsupervised knowledge graph generation using semantic similarity matching, in: C. Cherry, A. Fan, G. Foster, G. R. Haffari, S. Khadivi, N. V. Peng, X. Ren, E. Shareghi, S. Swayamdipta (Eds.), Proceedings of the Third Workshop on Deep Learning for Low-Resource Natural Language Processing, Association for Computational Linguistics, Hybrid, 2022, pp. 169–179. URL: https://aclanthology.org/2022.deeplo-1.18. doi:10.18653/v1/2022.deeplo-1.18.

[14] Artifex, Pymupdf, https://pypi.org/project/PyMuPDF/, 2024. Accessed: 2024-02-29.

[15] A. S. Schwartz, M. A. Hearst, A simple algorithm for identifying abbreviation definitions in biomedical text, in: R. B. Altman, A. K. Dunker, L. Hunter, T. E. Klein (Eds.), Proceedings of the 8th Pacific Symposium on Biocomputing, PSB 2003, Lihue, Hawaii, USA, January 3-7, 2003, 2003, pp. 451–462. URL: http://psb.stanford.edu/psb-online/proceedings/psb03/schwartz.pdf.

[16] M. Neumann, D. King, I. Beltagy, W. Ammar, ScispaCy: Fast and robust models for biomedical natural language processing, in: D. Demner-Fushman, K. B. Cohen, S. Ananiadou, J. Tsujii (Eds.), Proceedings of the 18th BioNLP Workshop and Shared Task, Association for Computational Linguistics, Florence, Italy, 2019, pp. 319–327. URL: https://aclanthology.org/W19-5034. doi:10.18653/v1/W19-5034.

[17] H. B. Travis Hoppe, Nlpre, https://github.com/NIHOPA/NLPre, 2024. Accessed: 2024-04-15.

[18] X. Wei, M. R. U. Hoque, J. Wu, J. Li, Claimdistiller: Scientific claim extraction with supervised contrastive learning, in: C. Zhang, Y. Zhang, P. Mayr, W. Lu, A. Suominen, H. Chen, Y. Ding (Eds.), Proceedings of Joint Workshop of the 4th Extraction and Evaluation of Knowledge Entities from Scientific Documents (EEKE2023) and the 3rd AI + Informetrics (AII2023) co-located with the JCDL 2023, Santa Fe, New Mexico, USA and Online, 26 June, 2023, volume 3451 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023, pp. 65–77. URL: https://ceur-ws.org/Vol-3451/paper11.pdf.

[19] A. Fergadis, D. Pappas, A. Karamolegkou, H. Papageorgiou, Argumentation mining in scientific literature for sustainable development, in: K. Al-Khatib, Y. Hou, M. Stede (Eds.), Proceedings of the 8th Workshop on Argument Mining, Association for Computational Linguistics, Punta Cana, Dominican Republic, 2021, pp. 100–111. URL: https://aclanthology.org/2021.argmining-1.10. doi:10.18653/v1/2021.argmining-1.10.

[20] F. Dernoncourt, J. Y. Lee, PubMed 200k RCT: a dataset for sequential sentence classification in medical abstracts, in: G. Kondrak, T. Watanabe (Eds.), Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers), Asian Federation of Natural Language

Processing, Taipei, Taiwan, 2017, pp. 308–313. URL: https://aclanthology.org/I17-2052.

[21] M. Honnibal, I. Montani, S. Van Landeghem, A. Boyd, spaCy: Industrial-strength Natural Language Processing in Python (2020). doi:10.5281/zenodo.1212303.

[22] I. Beltagy, K. Lo, A. Cohan, SciBERT: A pretrained language model for scientific text, in: K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3615–3620. URL: https://aclanthology.org/D19-1371. doi:10.18653/v1/D19-1371.

[23] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), Association for Computational Linguistics, 2019, pp. 4171–4186. URL: https://doi.org/10.18653/v1/n19-1423. doi:10.18653/V1/N19-1423.

[24] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0 Contributors, SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, Nature Methods 17 (2020) 261–272. URL: https://doi.org/10.1038/s41592-019-0686-2. doi:10.1038/s41592-019-0686-2.

[25] K. Ethayarajh, D. Duvenaud, G. Hirst, Towards understanding linear word analogies, in: A. Korhonen, D. Traum, L. Màrquez (Eds.), Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 3253–3262. URL: https://aclanthology.org/P19-1315. doi:10.18653/v1/P19-1315.

[26] L. Gyongyosi, S. Imre, A survey on quantum computing technology, Comput. Sci. Rev. 31 (2019) 51–71. URL: https://doi.org/10.1016/j.cosrev.2018.11.002. doi:10.1016/J.COSREV.2018.11.002.

[27] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J. Nie, J. Wen, A survey of large language models, CoRR abs/2303.18223 (2023). URL: https://doi.org/10.48550/arXiv.2303.18223. doi:10.48550/ARXIV.2303.18223. arXiv:2303.18223.

[28] G. A. Miller, WordNet: A lexical database for english, Communications of the ACM 38 (1995) 39–41.

[29] T. Chklovski, P. Pantel, VerbOcean: Mining the web for fine-grained semantic verb relations, in: D. Lin, D. Wu (Eds.), Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Barcelona, Spain, 2004, pp. 33–40. URL: https://aclanthology.org/W04-3205.

[30] A. Kucharavy, Z. Schillaci, L. Maréchal, M. Würsch, L. Dolamic, R. Sabonnadiere, D. P. David, A. Mermoud, V. Lenders, Fundamentals of generative large language models and perspectives in cyber-defense, 2023. arXiv:2303.12132.

6. Appendix

6.1. Abbreviation detection algorithms

During the preprocessing of papers, we expand abbreviations and map them to their long form. To compare the performance of different abbreviation detection algorithms, we evaluate them on the paper *Fundamentals of Generative Large Language Models and Perspectives in Cyber-Defense*, of which we have a thorough understanding [30].

Table 4 shows the performance of the Schwartz-Hearst [15], scispaCy [16] and NLPRe [17] abbreviation detection methods. The results show that the Schwartz-Hearst algorithm performs the best, though the scispaCy implementation has a similar performance. However, the Schwartz-Hearst algorithm is much faster, so we chose this approach.

Table 4
Performance of three abbreviation detection algorithms on the paper *Fundamentals of Generative Large Language Models and Perspectives in Cyber-Defense* [30].

	Correctly detected abbreviations	Falsely detected abbreviations	Processing time
Schwartz-Hearst	11	1	0.04 s
scispaCy	11	4	8.73 s
NLPRe	0	0	0.01 s

6.2. Triplet extraction

Figure 3 below illustrates the number of triplets extracted per paper. The results indicate that for most papers, less than 30 triplets are extracted. However, the right tail is long, which shows that there are outliers for which over 200 triplets are extracted. Table 5 shows the hyperparameters for the extraction of triplets for the arXiv papers with categories `cs.AI`, `cs.CL` and `cs.LG` from December 2023.

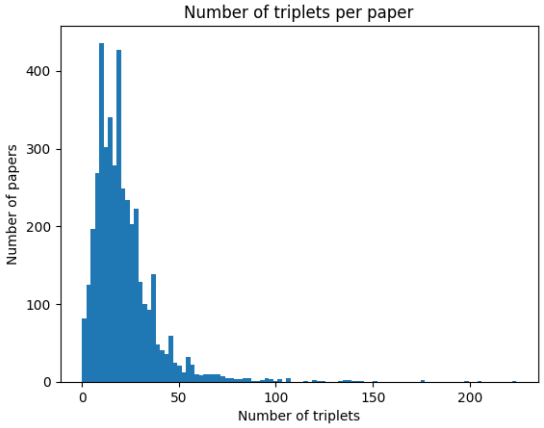

Figure 3: Histogram of the number of triplets extracted per paper from the categories cs.AI, c.CL and cs.LG in December 2023

Table 5
Hyperparameter setting for the triplet extraction

	Parameter value
Maximum length triplet component	6
Threshold claim extraction	0.05
Threshold for book corpus filtering	0.10
Threshold for arXiv corpus filtering	10
Threshold subject clustering	0.1
Threshold object clustering	0.1

6.3. Triplet clustering

Figure 4 shows for each cluster size the fraction of the clusters that only contains triplets from one category of data and the fraction that has triplets from both categories. The results show that smaller clusters more often tend to have triplets from one category, whereas larger clusters are more often mixed.

Figure 5 shows the dendograms for the clustering of the subjects and objects. We have chosen a cutoff of 0.10 for both, as this maintained a high similarity for the subjects and objects within the same cluster.

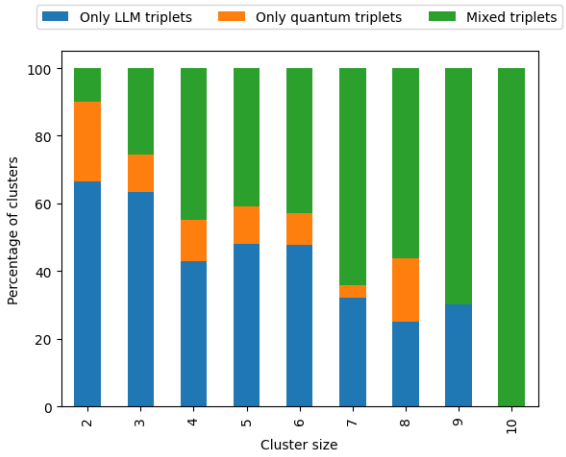

Figure 4: Fraction of the clusters with all triplets belonging to one category and fraction of clusters with triplets from both categories.

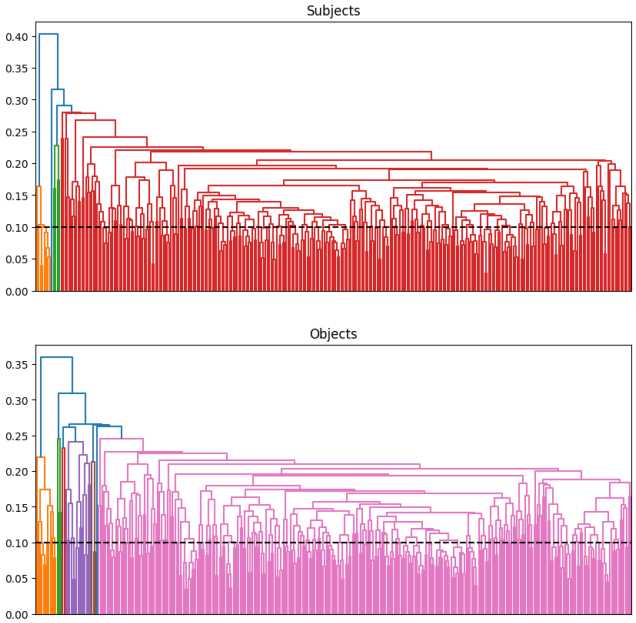

Figure 5: Dendrogram of the clustering of a subset of 350 subjects and 350 objects from the arXiv papers from the categories cs.AI, c.CL and cs.LG in December 2023

6.4. Embedding inversion

Table 6 shows 8 examples of embedding inversions, where the 10 most similar tokens are presented. Furthermore, the distance between the verbs (1 minus the cosine similarity) is shown. We find that the embedding inversions are not clearly interpretable, as the top 10 embedding inversions do not reflect the differences between the predicates. The embedding inversions are either similar to one of the two predicates, or are seemingly unrelated to both. Therefore, the embedding inversion cannot be used to assess whether triplets are aligned or contradictory.

Table 6
Manually selected triplets from the arXiv categories cs.AI, cs.CL, cs.LG from December 2023. The distance is defined as 1 minus the cosine similarity between the verb embeddings.

Original triplets	Verb 1		Verb 2	Top 10 embedding inversions	Distance
(example, illustrates, behavior), (example, mimic, behavior)	illustrates	-	mimic	ILLUSTRATES, ILLUSTRATING, schematically, EXEMPLARY, SUMMARIZES, DEPICTS, EMBODIMENT, ILLUSTRATED, ILLUSTRATIVE, DESCRIBES	0.75
(architecture, accomplishes, score), (architecture, achieves, score)	accomplishes	-	achieves	rigamarole, ERRAND, busywork, AFTERWORDS, canvasing, thigns, harrasing, forementioned, explaning, Busy-Work	0.22
(type rnns, perform, baseline model), (type rnns, outperform, baseline model)	performs	-	outperform	PERFORMS, PERFORMING, PERFORMED, PERFORM, SINGS, CONCERT, ACTs, SONG, RENDITION, PLAYS	0.32
(subgradient method, not ensure, convergence), (gradient algorithm, enjoy, convergence)	not ensure	-	enjoy	COMPLIANCE, COMPLY, INSUFFICIENT, IDENTIFIED, ENSURE, AUDIT, INDICATED, NON-COMPLIANCE, DETERMINES, IMPROPERLY	0.68
(image representation, extract, concept), (image representation, capture, concept)	extract	-	capture	EXTRACT, EXTRACTS, DECOCTION, TINCTURE, GINSENG, TURMERIC, Comfrey, KOLA, ALOE, Stevia	0.49
(language model, incurs, cost), (language model, slash, cost)	incurs	-	slash	INCURS,Accrues, ASCERTAINS, INCUR, INCURRING, incure, howsoever, INCURRED, internalizes, Indemnified	0.79
(text, represents, knowledge), (text, requires, knowledge)	represents	-	requires	REPRESENTS, REPRESENTED, REPRESENTING, RepresENT, ABSCISSA, symbolises, PERSONIFIES, DEPICTS, symbolised, Respresents	0.45
(knowledge transfer, demonstrates, improvement), (knowledge transfer, not maintain, improvement)	demonstrates	-	not maintain	DEMONSTRATES, demonstates, demostrates, EXEMPLIFIES, Dissects, Elucidates, DECONSTRUCTS, ILLUSTRATES, explicates, EXPLORES	0.49