Automatizing biocurators' intuition: filtering scientific papers by analyzing titles and short summaries

Ralf Stephan

Institute for Globally Distributed Open Research and Education (IGDORE)

Address correspondence to: Ralf Stephan, Institute for Globally Distributed Open Research and Education (IGDORE), https://igdore.org. E-mail: ralf.stephan@igdore.org

September 29, 2023

Abstract

We present a text classification task arising in the biocuration of cellular chemical reactions when searching for curatable literature. We explore the suitability of various NLP and ML methods for this task. In summary, while fine-tuned domain-specific language models show the best results, random forests are nearly as good, with a much lighter computational footprint.

Introduction

The natural strategy to automatize biocuration[1] is to apply natural language processing (NLP) methods on the full text of papers (e.g. [13]). This way, the claims and the type of evidence supporting them can be extracted. The problem is that only a certain percentage of papers are published in a way that allows bulk download of their full text. When we write this, even abstracts are available in bulk only from about 60 percent of molecular biology papers. This lamentable situation renders automatic curation efforts incomplete and causes the loss of essential information. Manual curation, on the other hand, has to deal with shortcomings of literature search engines, like not being able to filter for papers from specific fields and incomplete results because of missing entity synonyms. To cope with information overflow, biocurators develop the intuition to see from the title alone if a paper is, e.g., an experimental molecular biology paper, when confronted with search engine results. Sifting through thousands of titles in the course of a project still needs to be more sustainable and should be automatized.

The Reactome Knowledgebase[7] provides manually curated molecular details across a broad range of physiological and pathological biological processes in humans, including hereditary and acquired disease processes. Reactome biocurators look for a specific subset of scientific papers to use as references to their work. When using a search engine for scientific papers, searching for a pathogen finds epidemiological, clinical, pharmacological, and molecular/structural biology works. The latter also include evolutionary biology and pure computational papers, which are "uninteresting" from a curator's standpoint. Clinical and pharmacological works get mixed in the results even when searching for specific proteins. Such classification tasks could be solved using domain identification methods. Domain identification of scientific texts has been attempted frequently, most successfully using large language models (e.g. [16]).

Perspective of Reactome curators: what is an interesting paper?

Reactome annotation of molecular reactions is evidence-based[7]. Evidence comes from molecular biology and biochemistry lab experiments and observations. Scientists communicate such results through scientific articles, which Reactome curators then reference. However, there is no easy way to know from the title if a paper belongs to the experimental (our label: EXP) type. Many scientific papers focus on pharmacology (which we label as DRUG) and include antibodies that Reactome does not curate. Finally, review articles on experimental results are also 'interesting' to curators. To label a paper as 'interesting/'uninteresting' an algorithm doesn't need to find the actual research theme or scientific domain, just whether the paper is 'interesting,' which combines domain inclusions and exclusions. Since there is no database associating all existing and new articles to various academic domains, direct classification of membership to the 'interesting' set is more efficient than the implementation of a general domain identifier. In this report, we explore this task by using different natural

language processing (NLP) and machine learning (ML) methods on the title and a summary (so-called "TLDR"[3]), both of which are available from Semantic Scholar for >90 percent of biomedical articles.

Methods

Annotating the data

We annotated 32,837 academic papers that were selected using citation snowballing[11][21] focused on rotavirus (ROTV), respiratory syncytial virus (RSV), and dengue virus (DENV). A certain percentage of papers were about different, adjacent subjects, enriching the sets with other-viral and non-viral topics. The collection includes review articles. Annotation consists of a label with three possibilities: EXP, DRUG, OTHER. Annotated data is available as lists of maps in JSON format; see the Open Data section.

In the later preprocessing, we reduce the label to a binary decision between EXP and DRUG + OTHER. We merged the ROTV and RSV sets into the training set, and the DENV set served as the test set. Table 1 shows statistics of training and test data.

Dataset	Size	#TLDR	#Reviews	#EXP
ROTV	10,222	92%	23%	40%
RSV	13,139	94%	26%	39%
ROTV+RSV	22,995	93%	24%	39%
DENV	8,678	94%	16%	24%

Table 1: Dataset stats.

Preprocessing

Using CoreNLP[14], we tokenized, lemmatized, and part-of-sentence tagged all text (title and TLDR), discarding determiners, prepositions, and coordinating conjunctions. Our software concatenated Title and TLDR (if available), and in the case of review articles, appended the string "Review." Table 2 shows three random examples.

Before preprocessing	After preprocessing
A multifunctional nanoparticle as a prophylactic and therapeutic approach targeting respiratory syncytial virus	*multifunctional nanoparticle prophylactic therapeutic approach target respiratory syncytial virus*
A murine model for oral infection with a primate rotavirus (simian SA11). Simian rotavirus SA11 was shown to replicate in the gastrointestinal tracts of infant mice after oral inoculation, and clinical symptoms, histopathological changes in the small intestinal mucosa, and the type-specific humoral immune response were all characteristic of rotav virus-induced gastroenteritis.	*murine model oral infection primate rotavirus (simian sa11)* simian rotavirus sa11 be show replicate gastrointestinal tract infant mouse oral inoculation , clinical symptom , histopathological change small intestinal mucosa , type - specific humoral immune response be characteristic rotav virus - induce gastroenteritis
A murine model of dengue virus infection in suckling C57BL/6 and BALB/c mice. Three-day-old C57BL/6 mice intraperitoneally infected with DENV-2 NGC were more susceptible to infection than BALB/c mice, showing increased liver enzymes, extended viremia, dissemination to organs and histological alterations in liver and small intestine.	*murine model dengue virus infection suckling c57bl / 6 balb / c mouse* three - day - old c57bl / 6 mouse intraperitoneally infect DENV - 2 NGC be more susceptible infection balb / c mouse , show increase liver enzyme , extend viremia , dissemination organ histological alteration liver small intestine

Table 2: Preprocessing examples. The title is set in italics. All three examples fall in the 'uninteresting' category.

ML methodology

One-hot encoding as input to random forests

Training and test sets served as input for ML methods. To establish a baseline, we first trained random forest decision trees on raw bags of words, using the standard implementation from Yggdrasil[9] with a maximum number *N* of 300 trees. We used automatic tuning to optimize random forest models. In other attempts, we added the most common bigrams to the bags, also the default preprocessing style for LDA embeddings (see below).

Similarity measures

We saved sentence embeddings of the training set from BioSentVec[5], and compared embeddings of the test set by cosine distance, with the label of the most similar training embedding becoming the prediction. This method was also applied to just the paper titles.

Vector embeddings as input to random forests

Sentence embeddings take the whole text (which consists of the paper title and, in most cases, the TLDR, see above) as input and produce an *N*-dimensional vector as output. We applied a version of the Universal Sentence Encoder[4] for embedding the whole text as input to the already mentioned random forest. In addition, we tried the specialized biomedical sentence embeddings PubMedBERT[6][8] and BioSentVec. With PubMedBERT, we used the original texts and preprocessed strings for comparison.

Another type of embedding of whole texts used in NLP is Latent Dirichlet Allocation (LDA, [2][10]). We explored this method with its implementation in Python as part of the *gensim* package[18]. LDA constructs several topics from the corpus and associates test input with these topics through probabilities. The output of LDA applying test text to 10 to 100 topics of the training set served as input to random forests.

Fine-tuning of language models

Instead of just adding a random forest backend to sentence embeddings, we used the SetFit[20] methodology to fine-tune embeddings and to train a neural net classification layer simultaneously. Sentence transformers fine-tuned included `paraphrase-MiniLM-L3-v2`[19] and `menadsa/S-BioELECTRA`[12].

ML pipeline frameworks and hardware

The Yggdrasil random forest implementation requires TensorFlow[15], while BioSentVec and SetFit use spaCy[17]. The embedding plus random forest pipelines were easily trained within an hour on a quad-CPU Intel machine with 32 GB of RAM, without a graphics card. Fine-tuning pipelines, however, needed a machine helped by a V100 GPU.

Results

The table below summarizes the results of applying various methodologies. The optimal cutoff was found by maximizing the F1-value.

Discussion

The literature shows that domain-specific sentence embeddings are superior to general-purpose embeddings. We started experimenting with one-hot encoding our text as input to random forests and expected to get a base result on which sentence embeddings would improve. The results show a clear improvement with increasing complexity of the models used. However, only the most complex BioElectra model could improve slightly on our baseline random forest, which is surprising and needs explanation. It is plausible that the human curator applies simple rules when categorizing text on-the-fly, and that a decision tree can pick up such rules easier from a one-hot encoding than from a sentence embedding. Only with sufficient depth can neural nets model such inherently logical knowledge. Since the random forest implementation by Yggdrasil is also very fast, we see no gain in using large language models in the implementation of our forthcoming biocuration literature search application.

	Cutoff	Acc.	Prec.	Recall	F1
RF one-hot, 300 trees	0.55	92.3	**86**	84	85
same, auto-tuned	0.55	91.8	85	83	84
same, with bigrams, 1000 trees	0.55	91.6	83	85	84
Universal Sentence Encoder + RF 500 trees	0.5	84.9	69	76	72
PubMedBERT + RF 500 trees	0.4	89	76	81	78.5
BioElectra + RF 1000 trees	0.5	90.7	80	86	83
BioSentVec + cosine			58	80	67
BioSentVec + RF 500 trees	0.5	89.6	79	81	80
LDA + RF 1000 trees	0.6	85	70	73	71.5
`paraphrase-MiniLM-L3-v2` fine-tuned	0.85	89.5	76	88	81.3
`S-BioELECTRA` fine-tuned	0.6	**92.8**	84	**89**	**86.4**

Open data

All project files are available from `https://osf.io/hpf24/` and [https://github.com/rwst/LitBall-training].

References

[1] International Society for Biocuration. "Biocuration: Distilling data into knowledge". In: *PLOS Biology* 16.4 (Apr. 2018), pp. 1–8. DOI: 10.1371/journal.pbio.2002846. URL: https://doi.org/10.1371/journal.pbio.2002846.

[2] David M Blei, Andrew Y Ng, and Michael I Jordan. "Latent dirichlet allocation". In: *Journal of machine Learning research* 3.Jan (2003), pp. 993–1022.

[3] Isabel Cachola et al. "TLDR: Extreme Summarization of Scientific Documents". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, Nov. 2020, pp. 4766–4777. DOI: 10.18653/v1/2020.findings-emnlp.428. URL: https://aclanthology.org/2020.findings-emnlp.428.

[4] Daniel Cer et al. *Universal Sentence Encoder*. 2018. arXiv: 1803.11175 [cs.CL].

[5] Qingyu Chen, Yifan Peng, and Zhiyong Lu. *BioSentVec: creating sentence embeddings for biomedical texts*. 2018. arXiv: 1810.09302 [cs.CL].

[6] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL].

[7] Marc Gillespie et al. "The reactome pathway knowledgebase 2022". In: *Nucleic acids research* 50.D1 (2022), pp. D687–D692.

[8] Google. *experts/bert/pubmed: BERT trained on MEDLINE/PubMed*. URL: https://tfhub.dev/google/experts/bert/pubmed/2.

[9] Mathieu Guillame-Bert et al. *Yggdrasil Decision Forests: A Fast and Extensible Decision Forests Library*. 2022. arXiv: 2212.02934 [cs.LG].

[10] Matthew Hoffman, Francis Bach, and David Blei. "Online learning for latent dirichlet allocation". In: *advances in neural information processing systems* 23 (2010).

[11] Samireh Jalali and Claes Wohlin. "Systematic Literature Studies: Database Searches vs. Backward Snowballing". In: *Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*. ESEM '12. Lund, Sweden: Association for Computing Machinery, 2012, pp. 29–38. ISBN: 9781450310567. DOI: 10.1145/2372251.2372257. URL: https://doi.org/10.1145/2372251.2372257.

[12] Kamal raj Kanakarajan, Bhuvana Kundumani, and Malaikannan Sankarasubbu. "BioELECTRA:Pretrained Biomedical text Encoder using Discriminators". In: *Proceedings of the 20th Workshop on Biomedical Language Processing*. Online: Association for Computational Linguistics, June 2021, pp. 143–154. DOI: 10.18653/v1/2021.bionlp-1.16. URL: https://aclanthology.org/2021.bionlp-1.16.

[13] Martin Krallinger et al. “Evaluation of text-mining systems for biology: overview of the Second BioCreative community challenge”. In: *Genome Biology* 9.2 (Sept. 2008), S1. ISSN: 1474-760X. DOI: 10.1186/gb-2008-9-s2-s1. URL: https://doi.org/10.1186/gb-2008-9-s2-s1.

[14] Christopher D Manning et al. “The Stanford CoreNLP natural language processing toolkit”. In: *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*. 2014, pp. 55–60.

[15] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/.

[16] Óscar E Mendoza et al. “Benchmark for research theme classification of scholarly documents”. In: *Proceedings of the Third Workshop on Scholarly Document Processing*. 2022, pp. 253–262.

[17] Ines Montani et al. *explosion/spaCy: v3.6.0: New span finder component and pipelines for Slovenian*. Version v3.6.0. July 2023. DOI: 10.5281/zenodo.8123552. URL: https://doi.org/10.5281/zenodo.8123552.

[18] Radim Řehůřek and Petr Sojka. “Software Framework for Topic Modelling with Large Corpora”. English. In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. http://is.muni.cz/publication/884893/en. Valletta, Malta: ELRA, May 2010, pp. 45–50.

[19] Nils Reimers and Iryna Gurevych. “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Nov. 2019. URL: http://arxiv.org/abs/1908.10084.

[20] Lewis Tunstall et al. *Efficient Few-Shot Learning Without Prompts*. 2022. DOI: 10.48550/ARXIV.2209.11055. URL: https://arxiv.org/abs/2209.11055.

[21] Claes Wohlin et al. “Successful combination of database search and snowballing for identification of primary studies in systematic literature studies”. In: *Information and Software Technology* 147 (2022), p. 106908. ISSN: 0950-5849. DOI: https://doi.org/10.1016/j.infsof.2022.106908. URL: https://www.sciencedirect.com/science/article/pii/S0950584922000659.