SOFTWARE TOOL ARTICLE

REVISED PubRunner: A light-weight framework for updating text mining results [version 2; referees: 1 approved, 2 approved with reservations]

Kishore R. Anekalla [1], J.P. Courneya [2], Nicolas Fiorini [3], Jake Lever[4], Michael Muchow[5], Ben Busby [6]

[1]Northwestern University, Chicago, IL, 60611, USA
[2]Health Sciences and Human Services Library, University of Maryland, Baltimore, MD, 21201, USA
[3]National Center for Biotechnology Information, National Institutes of Health, Bethesda, MD, 20894, USA
[4]Canada's Michael Smith Genome Sciences Centre, University of British Columbia, Vancouver, BC, V5Z 4S6, Canada
[5]National Institute of Standards and Technology, Gaithersburg, MD, 20899, USA
[6]National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health, Bethesda, MD, 20894, USA

v2 **First published:** 02 May 2017, **6**:612 (doi: 10.12688/f1000research.11389.1)
Latest published: 13 Oct 2017, **6**:612 (doi: 10.12688/f1000research.11389.2)

Abstract

Biomedical text mining promises to assist biologists in quickly navigating the combined knowledge in their domain. This would allow improved understanding of the complex interactions within biological systems and faster hypothesis generation. New biomedical research articles are published daily and text mining tools are only as good as the corpus from which they work. Many text mining tools are underused because their results are static and do not reflect the constantly expanding knowledge in the field. In order for biomedical text mining to become an indispensable tool used by researchers, this problem must be addressed. To this end, we present PubRunner, a framework for regularly running text mining tools on the latest publications. PubRunner is lightweight, simple to use, and can be integrated with an existing text mining tool. The workflow involves downloading the latest abstracts from PubMed, executing a user-defined tool, pushing the resulting data to a public FTP or Zenodo dataset, and publicizing the location of these results on the public PubRunner website. We illustrate the use of this tool by re-running the commonly used word2vec tool on the latest PubMed abstracts to generate up-to-date word vector representations for the biomedical domain. This shows a proof of concept that we hope will encourage text mining developers to build tools that truly will aid biologists in exploring the latest publications.

This article is included in the Container Virtualization in Bioinformatics collection.

Open Peer Review

Referee Status: ? ? ✓

	Invited Referees 1	2	3
REVISED version 2 published 13 Oct 2017			✓ report
version 1 published 02 May 2017	? report	? report	? report

1 **Jin-Dong Kim**, Research Organization of Information and Systems, Japan

2 **Fabio Rinaldi**, University of Zurich , Switzerland

3 **Julien Gobeill**, University of Applied Sciences and Arts of Western Switzerland (HES-SO, HEG (Geneva School of Management)), Switzerland
Swiss Institute of Bioinformatics, Switzerland

Discuss this article

Comments (1)

This article is included in the Hackathons collection.

Corresponding author: Ben Busby (ben.busby@nih.gov)

Author roles: Anekalla KR: Conceptualization, Software; **Courneya JP**: Conceptualization, Software, Writing – Original Draft Preparation; **Fiorini N**: Conceptualization, Software; **Lever J**: Conceptualization, Methodology, Project Administration, Writing – Original Draft Preparation, Writing – Review & Editing; **Muchow M**: Conceptualization, Software, Writing – Original Draft Preparation; **Busby B**: Conceptualization, Resources, Supervision, Writing – Original Draft Preparation, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

How to cite this article: Anekalla KR, Courneya JP, Fiorini N *et al.* **PubRunner: A light-weight framework for updating text mining results [version 2; referees: 1 approved, 2 approved with reservations]** *F1000Research* 2017, **6**:612 (doi: 10.12688/f1000research.11389.2)

Grant information: This research was supported by the Intramural Research Program of the NIH, National Library of Medicine. JL is supported by a Vanier Canada Graduate Scholarship.
The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

First published: 02 May 2017, **6**:612 (doi: 10.12688/f1000research.11389.1)

REVISED **Amendments from Version 1**

We have read the reviewers comments carefully and made the following large changes to PubRunner and the associated paper. Firstly, we have altered the introduction section to better illustrate the challenge faced in the biomedical text mining community by out-dated results and tools. We hope this better illustrates the need for the PubRunner framework.

We have made two large changes to the codebase. PubRunner can now upload data to Zenodo which is a data repository designed for very large datasets to encourage open science. This will allow the output of text mining tools to be kept publicly available permanently. Furthermore Zenodo gives each dataset a Digital Object Identifier (DOI) making it easier for other researchers to cite exactly which datasets that they used.

We also added one extra use-case which is a commonly used text mining tool. Word2vec (Mikolov *et al.*) creates word representation vectors for terms in a corpus. This data can be used for interesting analysis on term similarity or as a useful input to other machine learning algorithms (Mehryary *et al.*). This resource is valuable to the biomedical community, requires substantial compute and storage to create (which may be outside the capability of smaller research groups), and is a good example of a resource that should be kept up-to-date. We hope this shows that PubRunner can be used with real text mining tools and not only the test cases that we had previously shown.

The option to upload to Zenodo has been added to the figure. The figure has also been simplified by removing text that describes possible text mining tools.

See referee reports

Introduction

The National Library of Medicine's (NLM) PubMed database contains over 27 million citations and is growing exponentially (Lu, 2011). Increasingly, text mining tools are being developed to analyze the contents of PubMed and other publicly searchable literature databases. These tools fall into three main categories based on the potential users. The first group of tools is aimed at other text mining researchers to help them solve problems. These tools can assist in parsing (e.g. Stanford CoreNLP - Manning *et al.*), entity recognition (e.g. DNorm - Leaman *et al.*) and other tasks (e.g. Word2Vec - Mikolov *et al.*). The second group of tools is aimed at expert curators to aid their creation of well-maintained biological databases. The third group of tools are aimed directly at biologists and provide automatically generated databases (such as miRTex - Li *et al.*), knowledge discovery capabilities (such as FACTA+ - Tsuruoka *et al.*) and many other uses.

Several challenges face researchers when trying to reuse the biomedical text mining methods and data of other researchers. Firstly, data and particularly code is rarely shared publicly. This is detrimental to the community and makes replicability and reproducibility very challenging. Furthermore, annotation formats and policies vary widely among research groups and specific biological domains. These problems are exacerbated by different ontology usages. These annotation issues limit the interoperability of different research tools and datasets. Finally, the data that is released is often static, as the text mining tool is only executed once and not rerun as new publications are released. This is commonly due to the goal of publishing a paper on the tool after which the tool is forgotten, the graduate student leaves the group and the project is abandoned.

As an example of a tool that would benefit from updated data, the FACTA+ tool (Tsuruoka *et al.*), which is aimed directly at biologists interested in understanding the associations of a biomedical concept, has not been updated since 2010. Given that it has not been updated, it misses a lot of important data, such as all recent information about Zika outbreaks. Many other tools (e.g. miRTex - Li *et al.*) have been run on a static set of Medline abstracts. Their results are incredibly useful but would prove more valuable if they were updated with the latest publications.

The open science movement has gained momentum in many areas of science. Studies (McKiernan *et al.*) show that science is stifled by researchers not sharing their data. Efforts such as Zenodo gives researchers an easy-to-use permanent and citable archive in which to store very large datasets. In fact, each dataset can be up to 50GB and is stored on the same robust servers as are used for data from the Large Hadron Collider. The challenge of maintaining up-to-date results requires additional engineering, which often goes beyond a basic research project. Some research is beginning to look at methods to maintain updated analysis on PubMed (Hakala *et al.*), but a general framework is needed.

To encourage biomedical text mining researchers to widely share their results and code, and keep analyses up-to-date, we present PubRunner. PubRunner is a small framework created during the National Center of Biotechnology Information Hackathon in January 2017. It wraps around a text mining tool and manages regular updates using the latest publications from PubMed. On a regular schedule, it downloads the latest Pubmed files, runs the selected tool(s), and outputs the results to an FTP directory or Zenodo archive. It also updates a public website with information about where the latest results can be located. We feel that this is a small but valuable step to help the text mining community produce robust and widely used tools and will encourage discussion about open data and open source development.

Methods

PubRunner manages monthly runs of text mining analyses using the latest publications from PubMed without requiring human intervention. The PubRunner framework has several key steps, outlined in Figure 1. First, it queries the PubMed FTP server to identify new XML files and downloads them. It currently downloads the Baseline dataset and then updates with the Daily Updates files (https://www.nlm.nih.gov/databases/download/pubmed_medline.html). It tracks which files are new and downloads the minimal required set to be up-to-date. Second, it executes the text mining tool(s) on the latest downloaded PubMed files. These tools are then run as Python subprocesses and monitored for exit status. Furthermore, PubRunner uses a timeout parameter to kill processes that exceed a time limit. PubRunner runs on the same private server used for the text mining analysis but moves results to a publicly visible FTP or permanent archiving on Zenodo after the analysis is complete. It requires FTP login information or Zenodo authentication token to be able to upload files.

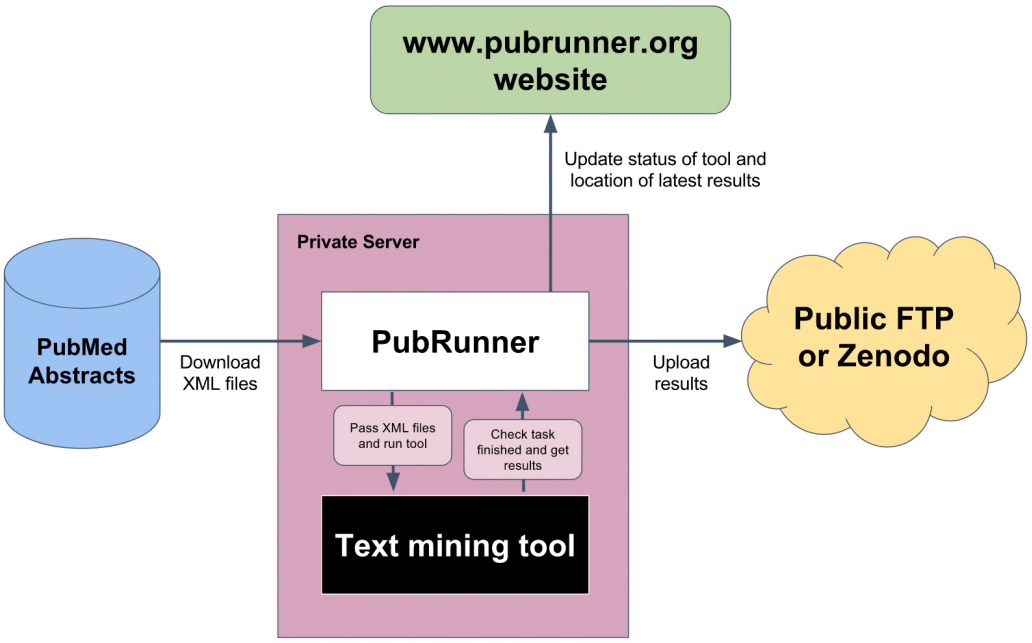

Figure 1. Overview of PubRunner. Overview of PubRunner. PubMed abstract files in XML format are downloaded to the PubRunner framework, processed by the text-mining tools, the output pushed to a public FTP/Zenodo site and an update sent to the central PubRunner website.

With this first step, PubRunner requires the tools that it executes to take a set of PubMed XML files. PubRunner does not guarantee any quality measure of the tools that are run. The users of the data generated by a PubRunner run should refer to the publication associated with any tools to understand the quality metrics that the original authors used to gauge the expected performance of the tool.

A central website was developed to track the status of different text mining analyses that are managed by PubRunner. These analyses may be executed on a variety of different researchers' computers with results hosted on different FTPs. The website lists the tools with information about their latest run and where their code and results can be found. This allows text mining users to more easily find robust and up-to-date analyses on PubMed.

A key design goal of PubRunner is to make installation as straightforward as possible. This is to encourage widespread use of the framework and release of both tool code and results data. Accordingly, a Docker image containing PubRunner has been produced, and installation from the Github code is also very straightforward. Also, each PubRunner component (server, website, and FTP) can be built by using the Docker file available for each in the GitHub repository. Deploying a specific component is thus made easy. Notably, there is not one central PubRunner FTP server. The output of PubRunner can be transferred to a pre-existing FTP server (e.g. an institution's FTP server) or a new FTP server can be set up using the Docker image. After PubRunner is installed, configuration involves setting the paths to the tools to be run and the login information for the FTP/Zenodo.

PubRunner currently has two dependencies: Python and R. The Docker file manages installation of these tools. The CPU and memory requirements required to run PubRunner depend on the associated text mining tools to be executed. PubRunner does require a reasonable amount of disk space, approximately 185GB, in order to download the full set of PubMed XMLs.

For a text mining tool developer to start using PubRunner, they first register their tool with the central website (http://www.pubrunner.org). Each tool should accept a set of Medline XML files as input and generate output files in a specific directory. The website gives them instructions on the necessary configuration settings (including an authentication token) so that their PubRunner instance can communicate with the central website. After each scheduled run of PubRunner on their remote server, an update message is sent to the website. This is implemented as an HTTP POST request to a PHP script on the PubRunner website. The request contains a JSON packet of information with an authentication token so that only submissions from authorized users are allowed. The JSON packet includes success status for the tools with URLs to the appropriate data. A potential extension to the website would hide tools that have failed for over three months and send notifications to the maintainers of each failed tool.

Use case

PubRunner was tested using three test-case text mining tools that were developed specifically for testing the framework and one

real-world text mining tool. These tools are also included in the Github repository.

The first of the test-case tools, named CountWords, generated basic word counts for each abstract in a PubMed XML file. It takes as input a list of PubMed XML files, parses the XML for the AbstractText section, splits the text by whitespace and counts the resulting tokens to give a naïve word count. It then outputs the set of word counts along with the corresponding PubMed IDs to a tab-delimited file. In order to test the robustness of the process management, two other tools that would fail were developed. The second tool, simply named Error, consistently failed. The third, named CountWordsError, uses the same code to calculate word counts as the first tool but would fail with a probability of 0.5. PubRunner successfully managed new runs of these test tools using updates from PubMed.

The real-world text mining tool was word2vec (Mikolov *et al.*). It is a commonly used tool to generate vector representations of individual words. These vector representations are a very commonly used resource in general NLP research and have been used in biomedical text mining (Mehryary *et al.*). Pyysalo *et al.* created vectors specifically for the biomedical domain in 2013 which are available at. New terms appear frequently and new relations form between biomedical concepts so it is important to update these vectors. We, therefore, built a small pipeline that takes in Pubmed XML files and feeds the raw text of the titles and abstracts from Pubmed citations into the word2vec tool.

At the time of publication, all four tools are deployed using PubRunner on a server hosted by the British Columbia Cancer Agency. PubRunner reruns the tools monthly and updates the results and status posted to the PubRunner website.

Conclusions and next steps

The PubRunner prototype reduces the additional engineering required for a text mining tool to be run on the latest publications. It will encourage the sharing of tool code and analysis data. At the moment, it can manage text mining runs using the latest Pubmed data. Future versions of the software will add additional corpora sources, such as PubMed Central, allow easier integration of ontologies and other bioinformatics resources and will include the ability to process only a subset of MEDLINE. While this is only the first step towards making biomedical tools easier to use, we hope that it will encourage discussion about how researchers can improve data and code sharing.

Data and software availability

PubRunner central website: http://www.pubrunner.org

Latest source code for the pipeline is publically available on GitHub: https://github.com/NCBI-Hackathons/PubRunner.

Archived source code as at time of publication: 10.5281/zenodo.892384 (Lever *et al.*, 2017)

License: MIT

The Docker image is available at https://hub.docker.com/r/ncbihackathons/pubrunner/.

Author contributions

All the authors participated in designing the study, carrying out the research, and preparing the manuscript. All authors were involved in the revision of the draft manuscript and have agreed to the final content.

Competing interests

No competing interests were disclosed.

Grant information

This research was supported by the Intramural Research Program of the NIH, National Library of Medicine. JL is supported by a Vanier Canada Graduate Scholarship.

The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Acknowledgements

We would like to thank Lena Pons and Lisa Federer for their valuable input.

References

Hakala K, Kaewphan S, Salakoski T, *et al.*: **Syntactic analyses and named entity recognition for PubMed and PubMed Central—up-to-the-minute.** ACL 2016, 2016; 102–107.
Publisher Full Text

Leaman R, Islamaj Doğan R, Lu Z: **DNorm: disease name normalization with pairwise learning to rank.** *Bioinformatics.* 2013; **29**(22): 2909–2917.
PubMed Abstract | Publisher Full Text | Free Full Text

Lever J, Fiorini N, Anekalla KR, *et al.*: **NCBI-Hackathons/PubRunner: Updated release for F1000 paper [Data set].** *Zenodo.* 2017.
Data Source

Li G, Ross KE, Arighi CN, *et al.*: **miRTex: A Text Mining System for miRNA-Gene Relation Extraction.** *PLoS Comput Biol.* 2015; **11**(9): e1004391.
PubMed Abstract | Publisher Full Text | Free Full Text

Lu Z: **PubMed and beyond: a survey of web tools for searching biomedical literature.** *Database (Oxford).* 2011; **2011**: baq036.
PubMed Abstract | Publisher Full Text | Free Full Text

Manning CD, Surdeanu M, Bauer J, *et al.*: **The stanford corenlp natural language processing toolkit.** ACL (System Demonstrations). 2014; 55–60.
Publisher Full Text

Mehryary F, Hakala K, Kaewphan S, *et al.*: **End-to-End System for Bacteria Habitat Extraction.** *BioNLP.* 2017; **2017**: 80–90.
Publisher Full Text

McKiernan EC, Bourne PE, Brown CT, *et al.*: **How open science helps researchers succeed.** *eLife.* 2016; **5**: pii: e16800.
PubMed Abstract | Publisher Full Text | Free Full Text

Mikolov T, Sutskever I, Chen K, *et al.*: **Distributed representations of words and phrases and their compositionality.** *Adv Neural Inf Process Syst.* 2013.
Reference Source

Pyysalo S, Ginter F, Moen H, *et al.*: **Distributional semantics resources for biomedical text processing.** *LBM.* 2013.
Reference Source

Tsuruoka Y, Miwa M, Hamamoto K, *et al.*: **Discovering and visualizing indirect associations between biomedical concepts.** *Bioinformatics.* 2011; **27**(13): i111–i119.
PubMed Abstract | Publisher Full Text | Free Full Text

Open Peer Review

Current Referee Status: ? ? ✓

Version 2

Referee Report 31 October 2017

doi:10.5256/f1000research.13740.r26975

✓ **Julien Gobeill** [1,2]
[1] Information Science Department, BiTeM Group, University of Applied Sciences and Arts of Western Switzerland (HES-SO, HEG (Geneva School of Management)), Carouge, Switzerland
[2] Text Mining group, Swiss Institute of Bioinformatics, Geneva, Switzerland

The authors made efforts to improve the article. Regarding the application of biomedical text mining and its potential users, they present admissible and well-founded arguments. The description if the software is now more technically sound. Thus, let's PubRunner find its potential users.

Competing Interests: No competing interests were disclosed.

Referee Expertise: Text mining, natural language processing, bioinformatics

I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Version 1

Referee Report 07 August 2017

doi:10.5256/f1000research.12294.r24597

? **Julien Gobeill** [1,2]
[1] Information Science Department, BiTeM Group, University of Applied Sciences and Arts of Western Switzerland (HES-SO, HEG (Geneva School of Management)), Carouge, Switzerland
[2] Text Mining group, Swiss Institute of Bioinformatics, Geneva, Switzerland

The authors' assumption is that many text mining tools are underused because their results are static (i.e. not updated with latest publications), and that biological experts could benefit from a centralized platform that would aggregate up-to-date outputs from various text mining tools. In this perspective, they propose and describe a framework: PubRunner.

The basic idea is simple and naive. Sometimes great things come from simple and naive ideas. I could imagine a centralized platform, where biological experts could explore up-to-date results of different tools,

and occasionally download a dataset: it would be quite like Commoncrawl, but for text mining. This idea is quite interesting, as the authors say: "an ecosystem of text mining tools running on the latest publications". Moreover, this could help these developed tools to have a bigger visibility and a longer life.

Yet, I cannot imagine that this kind of platform could be integrated in a real curation workflow.

First of all, the authors twice deal with the importance of sharing text mining results: "molecular biologists rely on text mining experts to run these tools on the latest publications and openly share their results". As a text miner, in contact with curators, this is not my life :) In my world, we text miners do not produce results, but tools and/or systems. Curators do not see us as data/results providers: they rather ask for quality-controlled and embedded assist in their curation workflow.

In a technical perspective, the presented idea has a lot of drawbacks.

Dealing with data format: what if the tool is not made for working from MEDLINE XML files as input? It seems that each tool would be free to deliver dataset in its own format (xml, json, csv...). How to describe the data content and format for the interested user? Moreover, computing data for all MEDLINE will result in huge dataset. Currently, there is a 260 Mo file just for word counts: would the dataset size still be manageable for more complex output? How to manage systems that are only relevant on a fraction of MEDLINE? (e.g. functional curation)

Dealing with data quality: how to know the quality of the dataset? This is extremely important for curators.

Dealing with computation: I don't think it is the job of Pubrunner to download MEDLINE updates (especially all the baseline, which will take days in a good server). "The PubRunner prototype reduces the additional engineering required for a text mining tool to be run on the latest publications": it can be true for text miners that are not used to work in biomedicine. For others, I assume they manage MEDLINE updates by their own. Moreover, I don't think that server managers will fully accept to have a third party platform running, especially on a production server.

As a conclusion, I think that this work could be the first step to a useful "thing", but it is far from being useful for the moment. The authors conclude that "this will certainly benefit biomedical researchers by allowing easier analysis of the latest publications", and this is not my opinion.

Is the rationale for developing the new software tool clearly explained?
Partly

Is the description of the software tool technically sound?
No

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Partly

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
No

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?
No

Competing Interests: No competing interests were disclosed.

Referee Expertise: Text mining, natural language processing, bioinformatics

I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response (*Member of the F1000 Faculty*) 13 Sep 2017

Ben Busby, National Center for Biotechnology Information (NCBI), National Institutes of Health, USA

We would like to thank you for taking the time to review this hackathon paper. We have read your comments carefully and have factored them into the changes.
You provided your perspective on the application of biomedical text mining and its potential users. We would suggest that there are broadly three groups of users for text mining tools. Biocurators (those that curate biological databases) are the one that you discussed. They require customised tools that fit their workflows. While some groups do have a text mining expert on staff who is able to provide assistance through the lifetime of the project, other projects may only have an expert for a short-time to start up the initiative. The text mining tools used by these curators provide suggested papers (or more) to the curators. It would seem useful to those groups that their text mining expert could fit the tool in the PubRunner framework so that continued curation suggestions could be provided after the expert has left the project.
However there are two other groups who benefit from text mining tools. There are biologists who use the output of text mining tools directly. These tools can be thought of as fully automated as they are not expertly curated. Understandably, these tools likely provide lower quality information but can be very useful in areas where expert curators cannot be found. Lastly there are text mining tools developed to be used by other text mining researchers (Hakala et al, Mikolov et al). We would suggest that these two groups would benefit greatly from automated updates using the latest publications.
You raise a very valid point with the varied input formats used by different tools. PubRunner, currently expects a tool to accept the Medline XML files. However we have included code that extracts the title and abstract text from Medline XML files. Individual pipelines can make use of this code.
The output data sizes of some text mining tools could be substantial. We have implemented an additional output hosting option with Zenodo. This allows datasets of up to 50GB to be hosted indefinitely. As an example of the ease of this, we added an additional tool (word2vec) that generates files of approximately 5GB.
We agree that some text mining tools need only be run on part of Pubmed. This could be by filtering using MeSH terms, publication dates or other factors. We intend to add this functionality as a future feature and have added it to the paper.
The quality of the output of the data is entirely dependent on the tool. PubRunner isn't able to make evaluations on the quality of a tool. Users of any data output by a tool would need to refer to the tool's documentation or associated publication in order to understand the expected quality of

output data.
You note that many biomedical text mining groups will already manage the download of MEDLINE baseline and update files. This exemplifies that each group has written their own code for this problem and shows the redundancy between groups. For new researchers, and particularly groups whose main focus is not biomedical text mining, they may not regularly download Pubmed. For a group that already manages the download of PubMed, we intend for a future version of PubRunner to accept the location of an existing download of Pubmed. It will then double check that all files are up-to-date and use that resource (thereby not duplicating downloading efforts).
As a product of a hackathon, we understand that further development is needed in order to make PubRunner production-ready. We have noted this in the conclusion. Many server managers happily run third-party tools on a regular basis and with improved testing and documentation, we feel that a manager could happily run this framework on their own server.

Competing Interests: No competing interests were disclosed.

Referee Report 07 August 2017

doi:10.5256/f1000research.12294.r24264

? **Fabio Rinaldi**
Swiss Institute of Bioinformatics, University of Zurich , Zürich, Switzerland

The idea presented in this short paper is clear: PubRunner is a software framework that might help to run text mining pipelines at regular intervals. Besides, Pubrunner is also capable of publishing their results on a reference web site.

Although PubRunner does probably have some utility, the paper fails to address or even discuss other problems that hinder the reusability of text annotations produced by different text mining systems. On the most basic level, different annotation schemas might be used (e.g. different XML formats, or Json). On an higher levels, different annotation labels might be used for the same entity types, or there might even be partial overlappings between annotation categories used by different systems. Another problems is how to deal with errors and noise introduced by each system. And the list of potential obstacles to integration certainly does not end there.

Another severe limitation of the paper is that the framework has been tested using only three "toy" scenarios: a word counter, and two error-generating systems. It would have been more interesting to pick at least one real-world text mining system, which perhaps might have prompted the authors to consider other potential problems that the toy systems did not present.

Is the rationale for developing the new software tool clearly explained?
Partly

Is the description of the software tool technically sound?
Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?
No

Competing Interests: No competing interests were disclosed.

Referee Expertise: Biomedical text mining, ontologies, terminology

I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response (*Member of the F1000 Faculty*) 13 Sep 2017

Ben Busby, National Center for Biotechnology Information (NCBI), National Institutes of Health, USA

You raise one of the large challenges in biomedical text mining: the different annotation schemes used by different groups. We agree that is one of the limiting factors for the re-use of annotated data. However we argue that text mining tools are not reused because code is not often released, and any results from the tool are left static. We have added a section during the introduction discussing the challenge of reusing others research in biomedical text mining.
We have also added an extra tool, word2vec, which is commonly used to generate word representation vectors for terms. Pyssalo et al (2013) created word vectors using the PubMed corpus, and these have been used by other researchers. However these vectors are now out-of-date. We added the word2vec tool to PubRunner. Furthermore we have added Zenodo as an option for hosting the resulting data. This allows the large files generated by word2vec to be hosted indefinitely.

Competing Interests: No competing interests were disclosed.

Referee Report 26 May 2017

doi:10.5256/f1000research.12294.r22433

? **Jin-Dong Kim**
Database Center for Life Science (DBCLS), Research Organization of Information and Systems, Tokyo, Japan

Overall, I like the idea of PubRunner. However, as a potential user of the tool, I have three problems described below.

1. While the motivation of the work is clear, what was not clear to me was the benefit of using a specialized tool like PubRunner instead of utilizing a more general scheduling tool like cron. I guess the authors know the benefits, but as it was not clear in the manuscript, I suggest the authors to explain it more clearly in the manuscript to better motivate potential users actually to choose to use PubRunner over alternatives.

2. In the manuscript, the protocol between an instance of PubRunner and the "central website" is not clearly described. It is even completely missing in Figure 1. I think designing such a protocol is not a trivial work, and it has be to clearly explained in the manuscript.

3. For the PubRunner to be more useful, I also think the method of publishing the results has to be more generalized. In fact, I have a big question about requiring the results to be uploaded to a FTP server. Can't it be a HTTP server? If PubRunner allows its user to choose the way of publishing the text mining results, it will become much more useful. This problem is actually related to the problem 1, e.g., if I choose to utilize cron, I can freely choose how to publish the result of my text mining pipeline.

Is the rationale for developing the new software tool clearly explained?
Partly

Is the description of the software tool technically sound?
Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Partly

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?
Partly

Competing Interests: No competing interests were disclosed.

Referee Expertise: text mining, database, question-answering

I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response (*Member of the F1000 Faculty*) 13 Sep 2017

Ben Busby, National Center for Biotechnology Information (NCBI), National Institutes of Health, USA

The reason to use the PubRunner framework over a simple CRON job is that PubRunner manages the download of files and upload of results files to a publicly available location. While many research groups have previously written code to manage the download of the PubMed baseline, there is incredible redundancy across these groups. It is worthwhile to create one implementation that can be used by existing groups or new groups as they start work in the biomedical text mining group.
We have added further details of the communication protocol between the PubRunner tool and the website (http://www.pubrunner.org) which is used to update the current status and information about data location.
You had also questioned the ability to only upload to FTP sites. Most HTTP websites will have an FTP access point which could be used by PubRunner. Furthermore, we have added functionality to upload to Zenodo which is a permanent data repository. It allows datasets up to 50GB, is hosted on the same infrastructure as the Large Hadron Collider and provides Digital Object Identifiers (DOIs) to allow PubRunner produced data to be easily citable.

Competing Interests: No competing interests were disclosed.

Discuss this Article

Version 1

Reader Comment 06 May 2017

Alexander Garcia-Castro, Florida State University, USA

Useful. And you got your first citation.

Competing Interests: No competing interests were disclosed.