searchrefiner: A Query Visualisation and Understanding Tool for Systematic Reviews

Harrisen Scells
University of Queensland [†]
Brisbane, Australia
harrisen.scells@hdr.qut.edu.au

Guido Zuccon
University of Queensland [†]
Brisbane, Australia
g.zuccon@uq.edu.au

ABSTRACT

We present an open source tool, searchrefiner, for researchers that conduct medical systematic reviews to assist in formulating, visualising, and understanding Boolean queries. The searchrefiner web interface allows researchers to explore how Boolean queries retrieve citations in existing, popular query syntaxes used in systematic review literature search. The web interface allows researchers to perform tasks such as using validation citations to ensure queries are retrieving a minimum set of known relevant citations, and editing Boolean queries by dragging and dropping clauses in a structured editor. In addition, the tools provided by the searchrefiner interface allow researchers to visualise why the queries they formulate retrieve citations, and ways to understand how to refine queries into more effective ones. searchrefiner is targeted at both experts and novices, as a tool for query formulation and refinement, and as a tool for training users to search for literature to compile systematic reviews.

The searchrefiner website located at https://ielab.io/searchrefiner contains information about how to download, install, and use the tool, as well as a link to an online hosted version for demonstration purposes.

ACM Reference Format:
Harrisen Scells and Guido Zuccon. 2018. searchrefiner: A Query Visualisation and Understanding Tool for Systematic Reviews. In *The 27th ACM International Conference on Information and Knowledge Management (CIKM '18), October 22–26, 2018, Torino, Italy*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3269206.3269215

1 PROBLEM AND TARGET USERS

Formulating queries for conducting systematic reviews is a lengthy and complex task. This process is often not methodical, and there currently exists no standard process for formulating systematic review queries [1, 4]. To aid in this process, an information specialist is often used to assist in the query formulation process [6] through interfaces such as PubMed.[1] These existing search interfaces do little in the way of allowing users to visualise or understand the impact of modifying a query. Prior work in other domains [2, 8, 9], however, has shown that visualisations can assist users with further refining queries to be more effective.

In this work, we present a domain-specific tool, *searchrefiner*, to assist researchers performing medical systematic reviews with formulating and refining Boolean queries. This tool is to be used in the phase of systematic review creation prior to screening literature for inclusion in the final review.

Formulating queries that capture all relevant literature for a particular systematic review while minimising the total number of retrieved citations is a complex and difficult task, even for expert searchers. The core problem is that a systematic review may have between, e.g., 1 and 100 relevant studies that comprise the synthesis of the review, however thousands or even tens of thousands of studies may be retrieved by a query. Currently, queries are typically formulated with the use of known-to-be-relevant citations (seeds) and examining the total number of retrieved citations. As the complexity (in Boolean clauses) of the query rises, it becomes less obvious, however, which particular clauses are discriminative or which clauses retrieve an overlapping set of citations.

It is these factors of query formulation and refinement that we envision searchrefiner to be used for. For example, one use for searchrefiner is to help refine the total number of relevant citations retrieved by a Boolean query. The visualisation of the query assists researchers to understand how many citations are retrieved by each clause, and how combining these clauses with logical operators affects the resulting set size. Another use case is to update an existing systematic review: by using the structured query editor, users may modify existing queries without needing to rewrite the entire query by hand. searchrefiner currently supports two of the most popular syntaxes for formulating queries for medical systematic reviews: Ovid MEDLINE and PubMed. This is made possible by a query translation component, that converts across query languages, integrated from an existing domain-specific experiment framework [7].

Furthermore, searchrefiner can be useful for training purposes. Enabling beginners to visually see how a query retrieves citations by combining clauses with logical operators can assist with the training process. In addition, augmenting the visualisation with relevance information (feedback) can assist novice users gain an intuition for how to retrieve all relevant citations while minimising the total number of retrieved citations. For example, trainees can load all citations included for synthesis in a particular systematic review as relevance information.

[†] This work was conducted while the authors were at Queensland University of Technology.

[1] https://www.ncbi.nlm.nih.gov/pubmed/

CIKM '18, October 22–26, 2018, Torino, Italy

ACM ISBN 978-1-4503-6014-2/18/10.
https://doi.org/10.1145/3269206.3269215

2 INNOVATIVE ASPECTS

Tools for assisting systematic review creation have seen increasing success and adoption in the medical systematic review community. These tools primarily focus on assisting reviewers synthesise data into a publication by automatically generating text [10], automatically assessing bias in clinical trials [3], and annotating published studies for the purpose of ontology population [5]. While tools outside of this domain do exist for query visualisation and understanding, they do not align with the needs of the researchers and information specialists undertaking systematic reviews. searchrefiner specifically addresses pain points of query formulation that creators of systematic reviews have, and neatly fits into an existing ecosystem of tools for assisting researchers create systematic reviews.

searchrefiner comprises three core components: (i) a *query interface* i.e. search engine results page, similar to existing tools (Figures 1, 2, and 3, Section 2.1), (ii) the *query visualiser* which allows users to explore how many citations are retrieved for each clause and the overlap between different clauses (Figures 4 and 5, Section 2.2) and to edit queries using a drag-and-drop structured editor (Figure 5), and (iii) a *query transformation* tool which allows users to directly edit the abstract syntax tree of a Boolean query (Figure 6, Section 2.3). Each component is described in detail in the following sections.

Users may issue, visualise, and edit queries in both the PubMed and Ovid MEDLINE query syntaxes. These are two very popular query syntaxes, and both are used extensively in systematic reviews. Currently no tools exist in the systematic review domain for visualising and understanding queries. For the first time, researchers creating and updating systematic reviews can gain a richer understanding of exactly what aspects of a query to refine to better improve the effectiveness of queries.

For both target users — i.e. experts with experience formulating queries and novice users learning how to formulate queries, searchrefiner can assist in novel ways. The most substantial contribution of which is the tree visualisation, which is to the best of the authors knowledge an entirely new, novel idea.

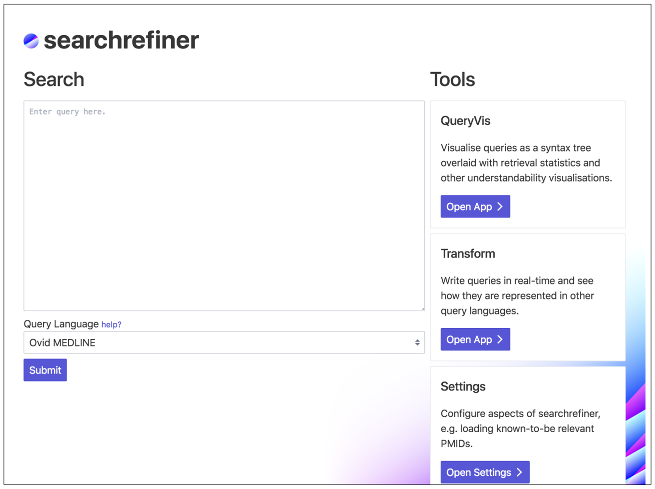
The following sections describe in detail the use cases of each aspect of searchrefiner with screen-shots demonstrating example usage.

2.1 Query Interface

The query interface (Figures 1 and 2) allows users of searchrefiner to issue queries in both Ovid MEDLINE and PubMed syntaxes directly to PubMed. The resulting page displays the total number of retrieved citations by the query, statistics about the query, and the abstract syntax tree of the query for debugging purposes.

2.2 Query Visualisation

The query visualisation component has three main aspects. The first is a two-way binding of textual query to a structured editor (Figure 5, the leftmost two tabs in this interface — the first set of two tabs is independent of the two right hand side tabs). This allows users to paste in existing queries (e.g., from other interfaces or existing systematic reviews) and edit them conveniently in a structured way by dragging and dropping clauses of the Boolean query. The logical operators, fields, queries, and MeSH explosion can be edited in this structured editor and all updates are reflected in the textual editor in real time. The structured editor can be used to supplement search interfaces in other tools because of this two way binding. For example, a query can be formulated in PubMed, but to make refinements to it without introducing syntactic errors one may paste it into the textual editor, edit in the structured editor, and copy it back to PubMed via the textual editor. Figure 7 presents what a typical query looks like in the Ovid MEDLINE syntax, and describes each of the details of the query. The structured editor removes the requirement for understanding the details of such a query. It allows, in fact, novice users to write and edit queries

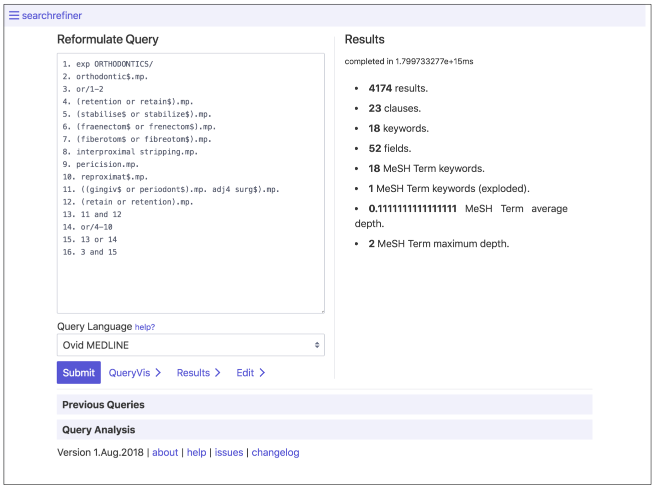


Visualise queries as a syntax tree overlaid with retrieval statistics and other understandability visualisations.

Write queries in real-time and see how they are represented in other query languages.

Configure aspects of searchrefiner, e.g. loading known-to-be relevant PMIDs.

Figure 1: Start page of searchrefiner. A search query in the Ovid MEDLINE or PubMed query syntax may be issued or a tool on the right hand side may be accessed. Previous queries can be seen below the search box and can be reissued.

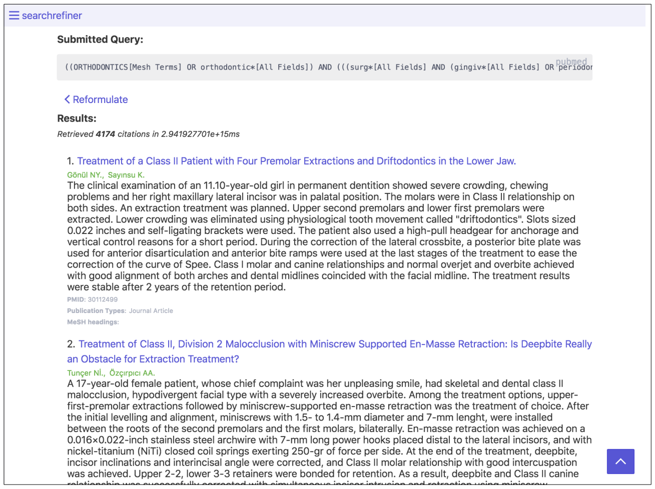


```
1. exp ORTHODONTICS/
2. orthodontic$.mp.
3. or/1-2
4. (retention or retain$).mp.
5. (stabilise$ or stabilize$).mp.
6. (fraenectom$ or frenectom$).mp.
7. (fiberotom$ or fibreotom$).mp.
8. interproximal stripping.mp.
9. pericision.mp.
10. reproximat$.mp.
11. ((gingiv$ or periodont$).mp. adj4 surg$).mp.
12. (retain or retention).mp.
13. 11 and 12
14. or/4-10
15. 13 or 14
16. 3 and 15
```

completed in 1.799733277e+15ms

- **4174** results.
- **23** clauses.
- **18** keywords.
- **52** fields.
- **18** MeSH Term keywords.
- **1** MeSH Term keywords (exploded).
- **0.1111111111111111** MeSH Term average depth.
- **2** MeSH Term maximum depth.

Figure 2: searchrefiner query page. This page can be used to view how many results a query retrieved, examine statistics about a query, and view how the query was transformed into an abstract syntax tree for debugging purposes.



Submitted Query:

```
((ORTHODONTICS[Mesh Terms] OR orthodontic*[All Fields]) AND (((surg*[All Fields] AND (gingiv*[All Fields] OR periodor
```

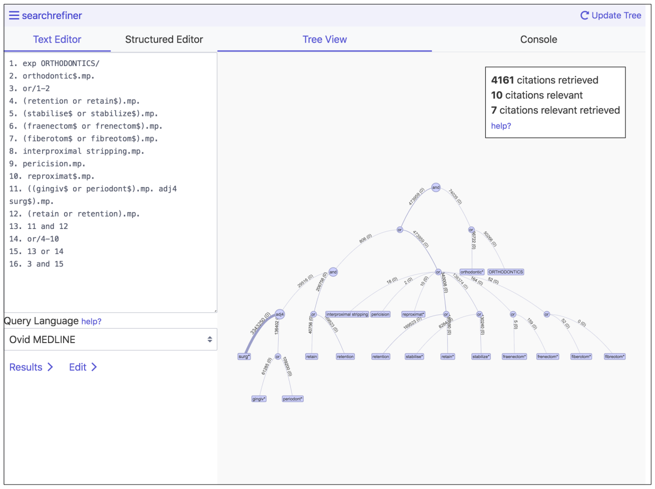
Results:

Retrieved ***4174*** *citations in 2.941927701e+15ms*

1. Treatment of a Class II Patient with Four Premolar Extractions and Driftodontics in the Lower Jaw.
Gönül NY., Sayınsu K.
The clinical examination of an 11.10-year-old girl in permanent dentition showed severe crowding, chewing problems and her right maxillary lateral incisor was in palatal position. The molars were in Class II relationship on both sides. An extraction treatment was planned. Upper second premolars and lower first premolars were extracted. Lower crowding was eliminated using physiological tooth movement called "driftodontics". Slots sized 0.022 inches and self-ligating brackets were used. The patient also used a high-pull headgear for anchorage and vertical control reasons for a short period. During the correction of the lateral crossbite, a posterior bite plate was used for anterior disarticulation and anterior bite ramps were used at the last stages of the treatment to ease the correction of the curve of Spee. Class I molar and canine relationships and normal overjet and overbite achieved with good alignment of both arches and dental midlines coincided with the facial midline. The treatment results were stable after 2 years of the retention period.
PMID: 30112499
Publication Types: Journal Article
MeSH headings:

2. Treatment of Class II, Division 2 Malocclusion with Miniscrew Supported En-Masse Retraction: Is Deepbite Really an Obstacle for Extraction Treatment?
Tunçer Nİ., Özçırpıcı AA.
A 17-year-old female patient, whose chief complaint was her unpleasing smile, had skeletal and dental class II malocclusion, hypodivergent facial type with a severely increased overbite. Among the treatment options, upper-first-premolar extractions followed by miniscrew-supported en-masse retraction was the treatment of choice. After the initial levelling and alignment, miniscrews with 1.5- to 1.4-mm diameter and 7-mm lenght, were installed between the roots of the second premolars and the first molars, bilaterally. En-masse retraction was achieved on a 0.016×0.022-inch stainless steel archwire with 7-mm long power hooks placed distal to the lateral incisors, and with nickel-titanium (NiTi) closed coil springs exerting 250-gr of force per side. At the end of the treatment, deepbite, incisor inclinations and interincisal angle were corrected, and Class II molar relationship with good intercuspation was achieved. Upper 2-2, lower 3-3 retainers were bonded for retention. As a result, deepbite and Class II canine relationship was successfully corrected with simultaneous incisor intrusion and retraction using miniscrew

Figure 3: Exploring the citations that are retrieved by the a query. The title and abstract are shown, as well as authors, the PubMed ID, the publication type, and any MeSH headings.



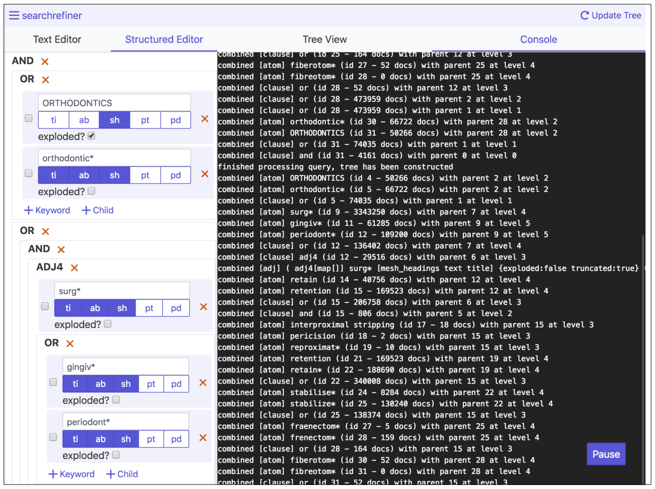
```
1. exp ORTHODONTICS/
2. orthodontic$.mp.
3. or/1-2
4. (retention or retain$).mp.
5. (stabilise$ or stabilize$).mp.
6. (fraenectom$ or frenectom$).mp.
7. (fiberotom$ or fibreotom$).mp.
8. interproximal stripping.mp.
9. pericision.mp.
10. reproximat$.mp.
11. ((gingiv$ or periodont$).mp. adj4
surg$).mp.
12. (retain or retention).mp.
13. 11 and 12
14. or/4-10
15. 13 or 14
16. 3 and 15
```

4161 citations retrieved
10 citations relevant
7 citations relevant retrieved

Figure 4: Visualising a query. The query on the left in the textual representation is visualised as a tree on the right. The nodes represent keywords (rounded-rectangles) and Boolean operators (circles); the edges show the number of citations retrieved by the node underneath.

without learning the different query syntaxes for each database that is used to retrieve literature for systematic reviews (all queries syntaxes look the same in the structured editor and the structured editor can output queries in different syntaxes).

The second innovative aspect of the query visualisation tool is the tree view (Figure 4). By using the tree view, one may visualise the number of citations that are retrieved by each clause of the query, and how the number of retrieved citations is affected by logical operators (i.e. AND, OR, NOT, etc.). This visualisation can be augmented with relevance information (either seed citations or



ORTHODONTICS
□ ti ab sh pt pd ✕
exploded? ☑

orthodontic*
□ ti ab sh pt pd ✕
exploded? □

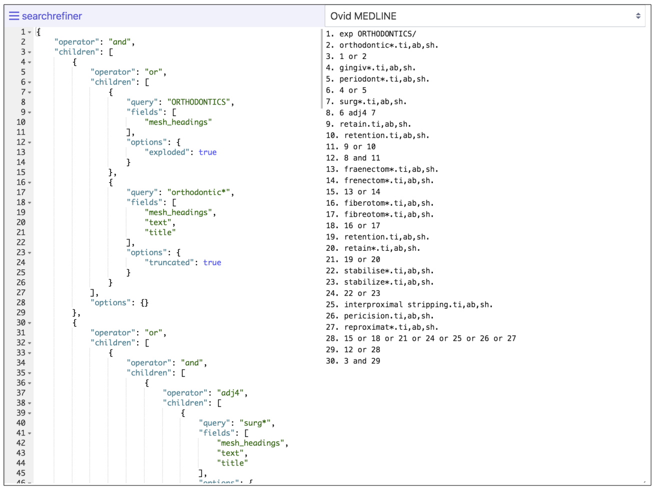
surg*
□ ti ab sh pt pd ✕
exploded? □

gingiv*
□ ti ab sh pt pd ✕
exploded? □

periodont*
□ ti ab sh pt pd ✕
exploded? □

```
combined [clause] or (id 25 – 164 docs) with parent 12 at level 3
combined [atom] fiberotom* (id 27 – 52 docs) with parent 25 at level 4
combined [atom] fibreotom* (id 28 – 0 docs) with parent 25 at level 4
combined [clause] or (id 28 – 52 docs) with parent 12 at level 3
combined [clause] or (id 28 – 473959 docs) with parent 2 at level 2
combined [clause] or (id 28 – 473959 docs) with parent 1 at level 1
combined [atom] orthodontic* (id 30 – 66722 docs) with parent 28 at level 2
combined [atom] ORTHODONTICS (id 31 – 50266 docs) with parent 28 at level 2
combined [clause] or (id 31 – 74035 docs) with parent 1 at level 1
combined [clause] and (id 31 – 4161 docs) with parent 0 at level 0
finished processing query, tree has been constructed
combined [atom] ORTHODONTICS (id 4 – 50266 docs) with parent 2 at level 2
combined [atom] orthodontic* (id 5 – 66722 docs) with parent 2 at level 2
combined [clause] or (id 5 – 74035 docs) with parent 1 at level 1
combined [atom] surg* (id 9 – 3343250 docs) with parent 7 at level 4
combined [atom] gingiv* (id 11 – 61285 docs) with parent 9 at level 5
combined [atom] periodont* (id 12 – 109200 docs) with parent 9 at level 5
combined [clause] or (id 12 – 136402 docs) with parent 7 at level 4
combined [clause] adj4 (id 12 – 29516 docs) with parent 6 at level 3
combined [adj] ( adj4[map[]] surg* [mesh_headings text title] {exploded:false truncated:true}
combined [atom] retain (id 14 – 40756 docs) with parent 12 at level 4
combined [atom] retention (id 15 – 169523 docs) with parent 12 at level 4
combined [clause] or (id 15 – 206758 docs) with parent 6 at level 3
combined [clause] and (id 15 – 806 docs) with parent 5 at level 2
combined [atom] interproximal stripping (id 17 – 18 docs) with parent 15 at level 3
combined [atom] pericision (id 18 – 2 docs) with parent 15 at level 3
combined [atom] reproximat* (id 19 – 10 docs) with parent 15 at level 3
combined [atom] retention (id 21 – 169523 docs) with parent 19 at level 4
combined [atom] retain* (id 22 – 188690 docs) with parent 19 at level 4
combined [clause] or (id 22 – 340008 docs) with parent 15 at level 3
combined [atom] stabilise* (id 24 – 8284 docs) with parent 22 at level 4
combined [atom] stabilize* (id 25 – 130240 docs) with parent 22 at level 4
combined [clause] or (id 25 – 138374 docs) with parent 15 at level 3
combined [atom] fraenectom* (id 27 – 5 docs) with parent 25 at level 4
combined [atom] frenectom* (id 28 – 159 docs) with parent 25 at level 4
combined [clause] or (id 28 – 164 docs) with parent 15 at level 3
combined [atom] fiberotom* (id 30 – 52 docs) with parent 28 at level 4
combined [atom] fibreotom* (id 31 – 0 docs) with parent 28 at level 4
combined [clause] or (id 31 – 52 docs) with parent 15 at level 3
```

Figure 5: Left: editing a query using the structured editor. Clauses of a query can be rearranged by dragging, and fields can be toggled using the buttons underneath each query. MeSH explosion (where applicable) may be selected here as well. Right: Output of the retrieval and clause combination process. This console is a read-only view of the current operation of the server.



```
{
    "operator": "and",
    "children": [
        {
            "operator": "or",
            "children": [
                {
                    "query": "ORTHODONTICS",
                    "fields": [
                        "mesh_headings"
                    ],
                    "options": {
                        "exploded": true
                    }
                },
                {
                    "query": "orthodontic*",
                    "fields": [
                        "mesh_headings",
                        "text",
                        "title"
                    ],
                    "options": {
                        "truncated": true
                    }
                }
            ],
            "options": {}
        },
        {
            "operator": "or",
            "children": [
                {
                    "operator": "and",
                    "children": [
                        {
                            "operator": "adj4",
                            "children": [
                                {
                                    "query": "surg*",
                                    "fields": [
                                        "mesh_headings",
                                        "text",
                                        "title"
                                    ],
                                    "options": {
```

```
1. exp ORTHODONTICS/
2. orthodontic*.ti,ab,sh.
3. 1 or 2
4. gingiv*.ti,ab,sh.
5. periodont*.ti,ab,sh.
6. 4 or 5
7. surg*.ti,ab,sh.
8. 6 adj4 7
9. retain.ti,ab,sh.
10. retention.ti,ab,sh.
11. 9 or 10
12. 8 and 11
13. fraenectom*.ti,ab,sh.
14. frenectom*.ti,ab,sh.
15. 13 or 14
16. fiberotom*.ti,ab,sh.
17. fibreotom*.ti,ab,sh.
18. 16 or 17
19. retention.ti,ab,sh.
20. retain*.ti,ab,sh.
21. 19 or 20
22. stabilise*.ti,ab,sh.
23. stabilize*.ti,ab,sh.
24. 22 or 23
25. interproximal stripping.ti,ab,sh.
26. pericision.ti,ab,sh.
27. reproximat*.ti,ab,sh.
28. 15 or 18 or 21 or 24 or 25 or 26 or 27
29. 12 or 28
30. 3 and 29
```

Figure 6: Editing a query using the abstract syntax tree. Edits to the abstract syntax tree are updated in real time on the right hand side.

relevance assessments) in order to monitor the number of known relevant citations retrieved, or for training purposes.

2.3 Transformation

The final component of searchrefiner allows users to directly edit the abstract syntax tree representation of queries, and visualise the resulting query in real time (Figure 6). This component receives a query from the other two components to allow for quick editing or debugging of a query. A one-way data binding between the editor on the left hand side and the query on the right updates in real time

```
1. exp ORTHODONTICS/
2. orthodontic$.mp.
3. or/1-2
4. (retention or retain$).mp.
5. (stabilise$ or stabilize$).mp.
6. (fraenectom$ or frenectom$).mp.
7. (fiberotom$ or fibreotom$).mp.
8. "interproximal stripping".mp.
9. pericision.mp.
10. reproximat$.mp.
11. ((gingiv$ or periodont$).mp. adj4 surg$).mp.
12. (retain or retention).mp.
13. 11 and 12
14. or/4-10
15. 13 or 14
16. 3 and 15
```

Figure 7: A typical query in the Ovid MEDLINE syntax. Keywords of the query can be expressed individually on lines (e.g. line 1 or 2), or grouped together (e.g. line 4). These keyword clauses can be combined with infix logical operators if grouped on a single line or by stating which lines to combine (e.g. line 15 combines the clauses specified on line 13 and 14 with the `OR` operator). Keywords are restricted to fields by the two letter identifier at the end (e.g. `.mp.`, which restricts the keyword to the title, abstract, and MeSH headings). MeSH only keywords are specified with `/` and can be exploded with `exp` (e.g. line 1). Keywords can also be explicitly stemmed or expanded with the `$` and `*` modifiers.

as the abstract syntax tree is edited. The query on the right may be represented either in Ovid MEDLINE or a PubMed syntax (the example in Figure 6 is outputting the query in MEDLINE syntax).

3 DEMO AUDIENCE EXPERIENCE

Conference attendees wishing to experience what it is like to use searchrefiner will be presented with the role of information specialist, tasked with refining a query to retrieve fewer non-relevant citations. This task is the primary use case of the tool and demonstrates the core visualisation and query editing aspects. Attendees will be assigned a query such as the one in Figure 7 and asked to use the tool to refine the search. The demonstration will be pre-loaded with known-relevant citations (seeds) that must be retrieved. An explanation of the query syntax and brief introduction to the tool will be provided prior to use of searchrefiner.

4 IMPACT

The impact of searchrefiner is two fold: it enables expert users to gain a deeper understanding of exactly how queries retrieve citations through visualisation, and how to refine these queries to produce more effective queries. Additionally, it provides a framework for novice searchers to learn how to search for literature by gaining an intuition of how queries retrieve citations visually. These two core aspects of searchrefiner have the potential to drastically improve the queries that are used to retrieve literature for systematic reviews.

We plan to continue development of this tool with the direction and assistance from experts by working closely with information specialists and researchers conducting systematic reviews. We also plan to improve the user experience and measure the effects this tool has on reducing the workload of systematic review creation by undertaking user studies. These user studies involve performing the same systematic review three times: one using existing query formulation interfaces, one using existing query formulation interfaces augmented with searchrefiner, and finally one using searchrefiner only. These studies will determine if searchrefiner has a measurable effect on the workload associated with systematic review creation.

5 USAGE

The primary and intended use of searchrefiner is through the web interface previously described. However, in addition to the web interface, there is a REST API available which can be used to perform the actions described in this paper. This allows searchrefiner to be both integrated into existing tools and to augment them.

searchrefiner is an open source project that is freely available to download from https://ielab.io/searchrefiner. Links to a hosted version may also be found via the aforementioned link, as well as information on how to compile and steps to set up searchrefiner locally.

Acknowledgements. The authors would like to thank Daniel Locke, Jimmy, Bevan Koopman, and Justin Clark for their insightful comments on early drafts of this work. Harrisen Scells is the recipient of a CSIRO PhD Top Up Scholarship. Dr Guido Zuccon is the recipient of and Australian Research Council DECRA Research Fellowship (DE180101579) and a Google Faculty Award.

REFERENCES

[1] Su Golder, Yoon Loke, and Heather M McIntosh. 2008. Poor reporting and inadequate searches were apparent in systematic reviews of adverse effects. *Journal of Clinical Epidemiology* 61, 5 (2008), 440–448.

[2] Orland Hoeber, Xue-Dong Yang, and Yiyu Yao. 2005. Visualization support for interactive query refinement. In *Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence.*

[3] Iain J Marshall, Joël Kuiper, and Byron C Wallace. 2015. RobotReviewer: evaluation of a system for automatically assessing bias in clinical trials. *Journal of the American Medical Informatics Association* (2015).

[4] Kathleen A McGraw, Margaret J Anderson, et al. 2009. Analysis of the reporting of search strategies in Cochrane systematic reviews. *Journal of the Medical Library Association* 97, 1 (2009), 21.

[5] Florina Piroi, Aldo Lipani, Mihai Lupu, and Allan Hanbury. 2015. DASyR (IR)-document analysis system for systematic reviews (in Information Retrieval). In *Proceedings of the 13th International Conference on Document Analysis and Recognition.* IEEE, 591–595.

[6] Melissa L Rethlefsen, Ann M Farrell, Leah C Osterhaus Trzasko, and Tara J Brigham. 2015. Librarian co-authors correlated with higher quality reported search strategies in general internal medicine systematic reviews. *Journal of Clinical Epidemiology* 68, 6 (2015), 617–626.

[7] Harrisen Scells, Daniel Locke, and Guido Zuccon. 2018. An Information Retrieval Experiment Framework for Domain Specific Applications. In *The 41st International ACM SIGIR Conference on Research And Development in Information Retrieval.*

[8] Christin Seifert, Johannes Jurgovsky, and Michael Granitzer. 2014. FacetScape: A visualization for exploring the search space. In *Information Visualisation (IV), 2014 18th International Conference on.* IEEE, 94–101.

[9] Anselm Spoerri. 1993. InfoCrystal: A visual tool for information retrieval & management. In *Proceedings of the second international conference on Information and knowledge management.*

[10] Mercedes Torres Torres and Clive E Adams. 2017. RevManHAL: towards automatic text generation in systematic reviews. *Systematic Reviews* 6, 1 (2017), 27.