Constructing Consistent Comprehensive Searches in Large Engineering Databases—Tips and Recommendations for Literature Reviews

Dr. Sarah Over, Virginia Tech

Dr. Sarah Over is the Engineering Collections and Research Analyst at Virginia Tech, serving as their Engineering Librarian and representative for their new Patent and Trademark Resource Center. She is also part of a team focused on research impact and intelligence to support the College of Engineering and Office of Research and Innovation at Virginia Tech. Dr. Over's background is in aerospace and nuclear engineering, with years of experience teaching engineering research methods and introductory coding.

C. Cozette Comer, Virginia Polytechnic Institute and State University

C. Cozette Comer has been conducting and supporting scoping/mapping reviews, systematic reviews, meta-analyses, and other forms of evidence synthesis since 2018 as both a researcher and information professional. She is currently the Evidence Synthesis Services Coordinator at the University Libraries at Virginia Tech, directing the development of support and educational services for faculty, students, staff, and community members across disciplines and in interdisciplinary contexts.

Constructing Consistent Comprehensive Searches in Large Engineering Databases – Tips and Recommendations for Literature Reviews

Abstract:
Engineering and STEM librarians and researchers regularly use a number of disciplinary databases for finding literature. For evidence synthesis (ES) research projects, knowing database capabilities and how to most effectively search each database is critical. The complex, often comprehensive search strategies necessary for ES reviews can require use of "advanced" search options and controlled vocabulary/index terms. However, the functionality of search interfaces used to access databases varies widely. Thus, searchers must expend more time and effort to translate searches consistently, as each database has different features and limitations.

This article will cover major engineering databases used in comprehensive searches, including Engineering Village and more. Each database's search implementation will be explained using an example search on hearing disabilities in computing education, which was developed for a scoping review by the authors in collaboration with a faculty member and a graduate student from the Computer Science department at Virginia Tech. Aspects of advanced searching such as truncation, proximity searching, exact phrases, and controlled vocabulary/index terms will also be highlighted. Overall, these databases require more research into how to construct searches compared to some interdisciplinary databases, but still have their place in finding quality engineering research efficiently.

Keywords:
Literature Reviews, Comprehensive Searches, Engineering Village, IEEE Xplore, ACM Digital Library

Introduction

Evidence synthesis (ES) methods such as scoping or mapping reviews, systematic reviews, and meta-analyses are a staple of the knowledge translation or research-to-practice cycle common in health and medicine. The underpinning characteristics of these methods require clear documentation of all stages of the review, from search to synthesis, such that the review could be audited or replicated. This clarity increases the reliability and validity of results and reduces risk of bias throughout the process. The Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA), a commonly used and recognized set of reporting guidelines, require a clear description of information sources and the search strategy, ideally presented in full for each database [1, Items 7 & 8]. Further emphasizing the importance of transparency of the search for ES reviews, in 2021 PRISMA introduced the PRISMA-S searching extension [2], providing additional guidance regarding the details that should be reported for systematic review and other comprehensive ES search strategies.

ES is no longer a methodology exclusive to health sciences. The value that ES methods can bring to the knowledge-translation process is highly applicable across disciplines. In recent years, these methods have been adopted by non-health fields like engineering. Even more recently, there have been calls for increased rigor and adherence to established methodological guidelines e.g., [3]. To meet the expanding needs of engineering faculty, staff, and students interested in applying ES methods, engineering librarians and information professionals require database search interfaces that support transparent, systematic, and complex search strategies.

Background

As systematic reviews and other ES methods become popular across disciplines, engineering researchers and information professionals require tools that allow for “transparent, methodical, and reproducible procedures” [4, pp. 46]. The complex, often comprehensive search strategies necessary for ES reviews can require use of “advanced” search options, transparency about search functionality, and relative consistency across databases. However, the functionality of search interfaces used to access databases varies widely.

The characteristics of databases for large, complex search strategies can vary, as well. For example, in Gusenbauer & Haddaway’s [5] evaluation of 28 databases for suitability of use in systematic reviews, they explored variation across databases in terms of: (a) functionality of Boolean operators (AND, OR, and NOT), (b) options for and functionality of search field limiters (e.g., title, abstract, and/or keyword searching), (c) maximum search string length and/or search limits based on number of characters or Boolean operators, (d) ability to truncate search terms, (e) ability to search using proximity operators, and (f) ability to search for exact terms and/or turn off automated “smart” searching. Other characteristics of databases such as the ability to bulk download results and the presence of a search history that allows for line searching are also important to running complex searches systematically.

Variation across databases (e.g., options available and how they are presented) can increase the time spent translating searches. This variation is easily illustrated by field codes. For example, searching for title and abstract is feasible with a single field code in PubMed by adding [tiab] or [Title/Abstract] to the end of each term. In EBSCOhost databases, title and abstract field codes must be applied independently then joined with an OR operator or by using the search history feature to line search. Scopus allows users to search titles, abstracts, and keywords using a single code that is applied to sets of terms within the parentheses that follow – *TITLE-ABS-KEY ()*). Controlled vocabulary (also called indexed terms, thesaurus terms) are used in comprehensive searches whenever available. These standardized sets of vocabulary are unique to each database and useful for capturing a more comprehensive set of literature systematically. For example, using a controlled vocabulary term for "material science" will bring back records that are related to material science (and might use terms like metallurgy, polymers, or composites) including records that may not use the exact phrase "material science" in title, abstract, or keywords.

Although it is a core principle of systematic review and similar ES methods [2], [4], [5], search interface functionality is not always clear. The search algorithms in academic journal databases are considered relatively transparent and unbiased compared to web browsers like Google Scholar e.g., [6], or semantic search and other AI tools [5]. Replicability and reproducibility of the search in systematic reviews is a current topic of interest in the broader field of ES. For example, studies such as the *REProducibility and Replicability in Syntheses of Evidence (REPRISE)* project evaluating the overall replicability/reproducibility of 2020 health topic systematic reviews address "The extent of variation in results when replicating the search, selection, data collection and analysis processes of an original review" [7].

Despite being an ongoing inquiry among ES researchers and information professionals more broadly, there is a need to expound the characteristics and limitations of engineering-specific databases. This paper highlights three commonly used platforms/databases: Engineering Village (EV), IEEE Xplore, and Association of Computing Machinery Digital Library (ACM DL) for engineering and computer science topics. Each database's search implementation will be explained using an example search on hearing disabilities in computing education, which was developed for a scoping review by the authors in collaboration with a faculty member and a graduate student from the Computer Science department at Virginia Tech. Aspects of advanced searching such as truncation, proximity searching, exact phrase searching, and controlled vocabulary/index terms will also be highlighted. More research regarding the transparency and replicability of search strategies across disciplinary and interdisciplinary databases is required, but this paper serves as a starting point for engineering researchers and information professionals seeking support for complex, systematic searches.

Overview of Engineering Databases

In this section, the coverage and capabilities of each database/platform will be explained to provide context for search syntax decisions. Overall EV, IEEE Xplore and ACM DL date back to the late 1800’s or early 1900’s in coverage. EV has a greater number of records (30+ million), however IEEE Xplore (6+ million) and ACM DL (3+ million) provide full text searching and access. All three also have their own versions of basic and advanced searching options, prompting significant differences in how searches are constructed for each.

Engineering Village

EV provides access to several databases depending on institutional subscription. For Virginia Tech, this includes Compendex, Inspec and Knovel. Knovel is also provided separately with advanced ebook and materials property search options in a separate platform. Within EV, there are three separate search mode options: “Quick” (basic), “Expert” (advanced) and “Thesaurus” (controlled vocabulary/index terms). Many filters can be applied after search execution. The exact filter options available will depend on the search mode (e.g., Quick v. Expert) and databases included. These filters include options common across all databases such as document type, year, author, source, country and more. Autostemming can be turned on or off (default is on for Quick and off for Expert), which can be useful if a user needs to only search specific stems or be transparent about the exact terms used (with autostemming off). Turning autostemming on can be helpful for expanding a search but will reduce transparency. For example, if autostemming is off, searching ‘engineer’ would only search that specific word while if autostemming is on, the terms engineering, engineered, etc. would also be included in search results.

EV’s Quick search is appropriate for many searches, from classroom instruction to literature reviews, and supports use of all operators and wildcards. A user can also add additional search boxes (12 total) to combine independent searches with AND, OR, or NOT. The Quick search does limit the fields searched however, so the user can only pick the most common options: all fields, subject/title/abstract, abstract, author, title, and publisher.

In contrast, the Expert search option has a high number of fields available and may switch automatically to this mode if a user filters by another field such as “Controlled Vocabulary.” Fields can be added to Expert searches by using the WN tag (WN is short for within in EV syntax) combined with a field code. For example, WN TI will search only in article titles. A full list of options available to EV subscribers includes TI (title), AB (abstract), AU (author), CV (controlled vocabulary), CF (conference information), PEC (cooperative patent classification, CPC), and many more including ALL (all fields).

EV also provides a searchable Thesaurus for users to find controlled vocabulary (CV) for the databases included in their institutional subscription. In the case of Virginia Tech, both Compendex and Inspec have index terms available for use as CV in EV. Users can find CV by searching for exact matches to a term (exact term), approximate matches to a term (vocabulary), or by browsing for CV of interest. The CV search results also provide links to broader, narrower, and related terms to help users find those relevant to their literature search. CV can also be conveniently exported after running a Quick or Expert search in the filters (as all filters offer in EV).

Although EV has many helpful features that assist in the precise searching required by ES reviews, there are some limitations with respect to proximity operators. Proximity operators such as NEAR have restrictions in that they cannot be used in conjunction with other Boolean logical statements (i.e. (A OR B) NEAR/1 (C OR D) - must be (A NEAR/1 C) OR (B NEAR/1 C) OR ...). This restriction combined with the autostemming feature means that users must think through searches critically to decide when to use autostemming and when to use exact phrases (or turn off autostemming). EV previously also restricted use of wildcards and truncation within exact searches (quotes), but as of early February 2024, this limitation has been removed and can be used freely now similarly to Scopus.

IEEE Xplore

Another large engineering database widely used is IEEE Xplore, and although a subscription is not required to search, depending on institutional subscriptions it can include full-text access to IEEE and other publishers. For the purposes of ES reviews, the breadth of IEEE Xplore is useful for a range of topics including power, electronics, computer science and even education and history related to these areas. IEEE Xplore provides three search options: “Global” (basic), “Advanced" (structured advanced search), and "Command” (unstructured advanced search). Unless a user updates their search preferences in their IEEE Xplore account, the default search field is all metadata and all publishers. After executing a search, users can add another search phrase using the “Search within results” option, functionally joining that search with an AND operator to the existing search. Like EV, IEEE Xplore provides several filters including type of work, year, author, affiliation, publication title, publisher, and more. Some filters may not show depending on the results retrieved by a search. For example, if no standards are returned by a search, the filters related to standards will not display.

The Global or basic search is appropriate for many applications as the search box supports Boolean operators, exact searching, and wildcards. Although this type of search does not allow users to select specific fields or controlled vocabulary, users can apply filters to their search before initial execution, for example by specifying document type (including courses) or citations.

Both the Advanced and Command searches allow for additional field search options and primarily differ in how a user inputs information. The Advanced search allows users to enter search terms into multiple search boxes (up to 10 extra boxes) and specify a field for each box and Boolean operator (AND, OR, NOT) to combine with the next box. The Command search is a single large box that offers a user the most flexibility when entering their searches, designed for advanced searches input in paragraph-type form. Users also have access to a built-in search builder that will add field codes and search operators (AND, OR, NOT, NEAR, ONEAR) to a set of terms.

Overall, IEEE Xplore functions similarly to Scopus in how it handles wildcards, proximity searching and exact searches, however there are some limitations. Currently IEEE Xplore is limited to 9 wildcards in a search but, allows for additional wildcards to be added post-execution using the search within results feature. Also, based on testing longer searches designed for ES reviews, IEEE Xplore does limit the length of search strings, causing some searches to generate an error if executed (when searching all metadata, over 500 words including operators sometimes is an issue, other times over 700). As IEEE includes autostemming, compact searches can be an effective option for avoiding this issue, though at the risk of losing precision regarding exact terms searched. Last, although IEEE Xplore does not have a thesaurus search, IEEE does offer download options for the taxonomy and thesaurus (version 1.03 as of January 2024).

Association of Computing Machinery Digital Library

The last database highlighted in this work is the Association of Computing Machinery's Digital Library (ACM DL). Similarly to IEEE Xplore, ACM DL can be searched without an institutional subscription, however a subscription provides access to full text. ACM DL also has broad coverage of computer science topics relevant to several engineering disciplines and related areas. ACM DL provides two search options, a basic search and an "Advanced" search that can be executed in either the "ACM Full-Text Collection" or the "ACM Guide to Computing Literature." The latter includes more types of publications that are considered grey literature in context for ES reviews – theses, presentations, and more. In contrast to EV and IEEE Xplore, ACM DL's default search all field code searches full text. ACM DL offers many filters that can be selected post search execution. Filters include advisors, reviewers, and sponsors in addition to the more common filters such as year, publisher, and content type filters available in most databases. One other difference in ACM DL compared to other databases is the option to use special characters instead of the standard Boolean operators AND, OR, and NOT in search statements. For example, && can represent AND, || or + can represent OR, and - or ! can represent NOT. For special characters to be included as part of the search terms, users must "escape" them with the backslash (\), similarly to syntax in LaTeX.

The basic search has less features in ACM DL compared to some databases, but still accepts quotes, wildcards, and Boolean operators. ACM DL will automatically add parentheses (uses square brackets) if not specified, automatically following Boolean order of operations such as A AND B OR C AND D being read/executed as [A AND B] OR [C AND D]. The default search also assumes an OR between words if not specified by the user.

The Advanced search is a more structured option with up to (only) two search boxes to search within specific fields including title, abstract, key words, full text and more. Users can also add filters to match all, any or none for metadata such as publisher or funding agency.

ACM DL has more limitations compared to EV and IEEE Xplore but includes broad full text-type searches convenient for specialized topics. For more complex searches, the main limitations are its inability to handle nested parentheses and lack of proximity operators. ACM DL does not support wildcards within exact searches or wildcards at the start of a search term. Last, ACM DL does not offer a searchable thesaurus or the option to include index terms/controlled vocabulary, although controlled vocabulary is functioning in the background and will be listed on the result pages for individual records.

Summary

Overall, Engineering Village, IEEE Xplore and ACM Digital Library each have unique search features and are distinct in functionality from other major databases/platforms. These differences require significant modification of complex searches often employed in ES reviews. **Table 1** summarizes some key features and limitations of each database using Scopus as a baseline. These include characteristics that directly impact the ability to search for literature in a transparent and/or precise manner such as wildcards and proximity search features. Given any combination of databases/platforms, users will need to make decisions about how to navigate these in a consistent, and ideally replicable manner. See the Appendix for links to search documentation from EV, IEEE Xplore and ACM DL.

Table 1: Database features overview relevant to search string development.

Database/ Platform	**Records (millions)**	**Search length high limit?**	**Full-text searching?**	**Wildcards used freely?**	**Proximity Operators?**	**Reads nested parentheses?**
Scopus	90+	√	x	√	√	√
EV	30+	√	x	√	√	√
IEEE Xplore	6.2+	x	√	x	√	√
ACM DL	3.6+	√	√	x	x	x

Constructing Searches for ES Applications

As covered above, the databases/platforms presented here each have features and limitations that can make it challenging to construct equivalent searches in each database. This section will present options for each based on an example search developed for a scoping review collaboratively between a faculty member and a graduate student in Virginia Tech's Computer Science department and the University Libraries' ES Services. Although search strategy development was iterative over a few months as is common in ES projects, the process of designing the search strategy will not be covered. Instead, a portion of this search strategy is used to demonstrate consistent construction of the search strings in the three databases/platforms explored in preceding sections.

The example scoping review aims to address the following research question: What factors contribute to barriers and successes for students who are Deaf or Hard-of-Hearing (HoH) in STEM and computing-related undergraduate programs? This review search strategy was split into four topical areas: (a) higher education, (b) deaf/hard-of-hearing, (c) STEM fields, and (d) persistence/success. The latter two areas were simple search constructions consisting of multiple words joined by OR (such as: (achievement OR success OR ...)). The first two areas required more advanced search options, including proximity operators. Therefore, the examples presented here will focus on higher education and deaf/hard-of-hearing for brevity and to reserve the complete search for the full scoping review publication.

Presented in **Table 2** is a summary of the search terms for higher education and deaf/hard-of-hearing and how they will be combined in the search string. This table has been abbreviated to only include one stem of most search terms. However, a searcher could use autostemming, truncation (e.g., colleg*), or spell out the specific endings of interest (e.g., College OR Colleges OR Collegiate), depending on their needs and capabilities of each database. Note that not all possible terms are included in the examples presented here. Some might have been excluded from the full search as testing revealed some potentially relevant terms ineffective for the research question (such as institution – results introduced noise from medical institutions and academic institutions were already captured).

Table 2: Summary of search terms within search structure for two areas of the example search. Note the split in columns indicates an AND, while terms in the same box are connected by OR unless specified.

<table>
<tr><th colspan="2">Higher Education</th><th colspan="2">Hearing</th></tr>
<tr><td colspan="2">College
University
Undergraduate
“higher education”</td><td colspan="2" rowspan="2">Deaf
Deafness
hypoacusis
Presbycusis
Hard-of-hearing
Hearing-impair
Unhearing
Nonhearing
Non-hearing
“non hearing”
“Cochlear implant”
“Cochlear implants”</td></tr>
<tr><td rowspan="3">Post-secondary
Postsecondary
“post secondary”
Post-graduate
Post-graduates
Postgraduate
Postgraduates
“post graduate”
“post graduates”
Tertiary
First-year
“first year”
...
Fourth-year
“fourth year”
Freshman
Sophomore
Junior
Senior</td><td rowspan="3">Academia
Education
School
Student
Teach
Instruction
Pedagogy
Didactic</td></tr>
<tr><td colspan="2">Proximity, n = 1
Any order</td></tr>
<tr><td>Aid
Hard
Hardly
Impair
Challenge
Constrain
Defect
Difficult
...
Partial
Poor</td><td>Hear
Hears
Hearing
auditory</td></tr>
</table>

From **Table 2** the following search is then constructed for use in multiple databases for autostemming (with quotes added to prevent autostemming or prevent separation of terms joined with a dash where needed):

If using autostemming, the search would be constructed as:

```
((college OR university OR undergraduate OR "higher education")
OR (("post-secondary" OR postsecondary OR ... OR freshman OR ...
OR senior) AND (academia OR ... OR didactic)))

 AND

(("deaf" OR "deafness" OR hypoacusis OR ... OR "cochlear
implant*") OR ((aid OR "hard" OR "hardly" OR ... OR "hearing-
impair*" OR ... OR poor) NEAR/1 ("hear" OR "hears" OR "hearing"
OR auditory)))
```

If not using autostemming then the search would be constructed as:

```
((colleg* OR universit* OR undergrad* OR "higher education") OR
(("post-secondary" OR postsecondary OR ... OR freshman OR
freshmen OR ... OR senior*) AND (academ* OR ... OR didactic*)))

 AND

(("deaf" OR "deafness" OR hypoacusis OR ... OR "cochlear
implant*") OR ((aid* OR hard OR hardly OR ... OR "hearing-
impair*" OR ... OR poor*) NEAR/1 (hear OR hears OR hearing OR
auditory)))
```

Either one of these options will cause issues in EV, IEEE Xplore, and ACM Digital Library (DL). In addition, this does not include any index terms or controlled vocabulary yet, another unique characteristic across databases.

Engineering Village – Search Construction

In EV, directly using either of the above search options would result in errors due to the way the NEAR operator is used. To include all these possible terms with the proximity operator, that portion of the search must be rewritten as follows for each term:

Autostemming on:

```
(aid NEAR/1 "hear") OR (aid NEAR/1 "hears") OR (aid NEAR/1
"hearing") OR (aid* NEAR/1 "auditory")...
```

Autostemming off:

```
(aid* NEAR/1 hear) OR (aid* NEAR/1 hears) OR (aid* NEAR/1
hearing) OR (aid* NEAR/1 auditory)...
```

In the case of navigating the use of NEAR (or ONEAR) operators, it might be preferred to use autostemming or truncation and wildcards to avoid the need to excessively expand the search string. If needed, users should consider using spreadsheet software to automatically concatenate terms. For example, in Excel, this can be accomplished via:

```
=B2&" NEAR/1 "&$C$2
```

Where columns B and C contain terms to be used in the proximity search (aid and hear above).

After filling down columns, the textjoin function can be used to combine all variations into one search string (where columns E and F are the results of the NEAR combining above):

```
=TEXTJOIN(" OR ", TRUE, E2:F83)
```

After accounting for limitations with respect to proximity searching, the example searches presented will execute without errors.

IEEE Xplore – Search Construction

In IEEE Xplore there are two limitations to account for the example presented here: wildcard use and search string length. IEEE Xplore includes autostemming by default and there is no option to turn this off. This search must also be executed via the Global or Command search mode as proximity operators are not supported in the structured Advanced search option. If it is desired to not search the default "All Metadata" each field would need to be searched separately to match the title/abstract/keywords search approach employed in other databases.

To address wildcard limits, remove all uses of * if possible (limit is 9 wildcards, * or ? used):

```
((college OR university OR undergraduate OR "higher education")
OR (("post-secondary" OR postsecondary OR ... OR freshman OR ...
OR senior) AND (academia OR ... OR didactic)))

AND

(("deaf" OR "deafness" OR hypoacusis OR ... OR "cochlear implant"
OR "cochlear implants") OR ((aid OR "hard" OR "hardly" OR ... OR
"hearing-impair" OR "hearing-impaired" OR "hearing-impairment" OR
... OR poor) NEAR/1 ("hear" OR "hears" OR "hearing" OR
auditory)))
```

However, if expanding the search by including exact phrases in place of truncation, the search string may become too long to be executed in IEEE Xplore (may attempt to complete search for a few minutes or less and then give an error message). The user must decide whether to use ONEAR/0 instead and rely on autostemming. This will result in a higher results yield which may be undesirable, depending on the search purpose.

If searching all metadata fields is not desired, the user can either run multiple searches, download all results, and deduplicate in their selected review software, or combine the metadata searches with OR to capture all results in one search. For example, the higher education area could be searched as follows (and then combined with a similarly formatted string for the other topical areas):

```
(("Document Title":college OR "Document Title":university ...

 OR

(("Abstract":college OR "Abstract":university ...

 OR

(("Author Keywords":college OR "Author Keywords":university ...
```

Again, to build these searches, spreadsheet software is recommended to combine terms with field codes efficiently. Lastly, users should be aware that stylized quotes (e.g., those formatted in a word processor) can produce errors in IEEE Xplore and will need to remove those before executing searches (e.g. " " vs “ ”).

ACM Digital Library – Search Construction

ACM DL requires the most modification of search strings compared to the other databases due to limitations for nested parentheses and lack of proximity operators. In addition, full text and autostemming are executed by default for searches in ACM DL, which may not be desired depending on the search.

First, a user will need to decide how to approximate a proximity search either by using exact phrase searching or the AND operator. In this case, an exact phrase search might be more effective as it is expected that the terms “hard” and “hearing” will be commonly next to one another or within one word (NEAR/1). In contrast, if NEAR/3 or 5 were to be used, AND might be a more logical approximation. For the search presented here, the proximity phrase can be then modified as follows (using a spreadsheet for efficiency):

```
(‶aid hear″ OR "hear aid″ OR ‶aid hearing″ OR "hearing aid″...
```

Next, the issue of nested parentheses in the education terms can be addressed similarly as ACM DL will add parentheses for the user:

```
(college OR university OR undergraduate OR ‶higher education″ OR
‶post-secondary″ AND academia OR ‶post-secondary″ AND
education...
```

In ACM DL will be read as:

```
[All: college] OR [All: university] OR [All: undergraduate] OR
[All: "higher education"] OR [[All: "post-secondary"] AND [All:
academia]] OR [[All: "post-secondary"] AND [All: education]]...
```

Combining these will result in a much longer search, but will be contained in (A OR B...) AND (C OR D...) AND... type searches that ACM DL can execute. Finally, to limit the "all" field code searches demonstrated in the result above, a user can pick specific metadata in the Advanced search or add field codes similarly to IEEE Xplore.

Adding Controlled Vocabulary and Index Terms

Once searches have been modified for each database, EV and IEEE Xplore also offer the option to search controlled vocabulary/index terms. In EV, both Compendex and Inspec have their own vocabularies, so a user will need to search both to find all options. While in IEEE Xplore, a user will need to explore the downloaded thesaurus and/or taxonomy. For the example search presented here, the discovered terms are presented below in **Table 3**.

Table 3: IEEE Xplore index terms and EV controlled vocabulary selected for the example search. Note that there were additional controlled vocabulary terms for higher education, but those were inclusive of museums and/or K-12 education and deemed irrelevant to the search.

	Higher Education	**Hearing**
EV	further education	Cochlear implants Hearing aids
IEEE Xplore	(none)	deafness cochlear implants auditory implants hearing aids

Whether in IEEE Xplore or EV, these terms are added via OR with the appropriate field code as part of the search:

Engineering Village (use Expert search):

```
(((colleg* OR universit* OR undergrad* ...) WN KY) OR ({further
education} WN CV))

AND

((("deaf" OR ... (aid* NEAR/1 "hear") OR (aid* NEAR/1 "hears") OR
(aid* NEAR/1 "hearing") OR (aid* NEAR/1 "auditory")... ) WN KY)
OR (({cochlear implants} OR {hearing aids}) WN CV))
```

IEEE Xplore (use Command search, not searching specific fields, all metadata):

```
((college OR university OR undergraduate OR “higher education”)
OR ((“post-secondary” OR postsecondary OR ... OR freshman OR ...
OR senior) AND (academia OR ... OR didactic)))

 AND

((“deaf” OR “deafness” OR hypoacusis OR ... OR “cochlear implant”
OR “cochlear implants”) OR ((aid OR “hard” OR “hardly” OR ... OR
“hearing-impair” OR “hearing-impaired” OR “hearing-impairment” OR
... OR poor) NEAR/1 (“hear” OR “hears” OR “hearing” OR auditory))
OR (“IEEE Terms”:deafness OR...))
```

For ACM DL, there are index terms that can be viewed for individual articles, however they are not searchable due to there not being an explicit field code. The only keyword or term field code is for author supplied terms, not index terms.

Overall, each of these platforms/databases will require modification of search strings to execute searches consistently. It is vital that information professionals are aware and understand how to navigate the unique characteristics of core engineering databases when designing and executing systematic, comprehensive searches such as those required for evidence synthesis reviews. In Engineering Village, a user will need to account for limitations with proximity operators that will require expansion of search strings or decide to use alternative search syntax to match other engineering databases such as ACM DL. EV has the advantage compared to IEEE and ACM DL though in that one can easily search several fields including title/abstract/keywords simultaneously using KY. IEEE Xplore in comparison can support proximity operations without the need for expansion and does support full-text searching. However, the wildcards and search length may need adjusted due to built-in limitations, and searching fields besides all metadata must be done individually. Last, ACM DL is generally the most limited of these databases with regards to search syntax due to lack of proximity operators and issues with nested parentheses that will require adjustment of search strings. ACM DL, like IEEE Xplore provides searching full text, although it is by default instead of an option, although unlike IEEE Xplore and EV controlled vocabulary cannot be searched. Finally, ACM DL is probably the most unique in syntax out of these three databases as well due to the options of using special characters such as && or || to mean AND or OR respectively, so users must be careful of entry to ensure search strings are read as intended.

Although it will require further effort for the user, modifications described in this paper will help promote consistent searches across these databases and with other databases such as Scopus or those offered in EBSCOHost. In the Appendix, further information is provided, including a link to a spreadsheet for users to try these modifications themselves. Anecdotally, while conducting this analysis, minor updates were launched in the Engineering Village database which allowed for more sophisticated searching. Therefore, in addition to the findings and resources presented in this paper, it is also imperative that information professionals who must conduct transparent, systematic searches stay abreast of changes in database functionality over time.

Conclusions

The need for systematic, replicable, and consistent searches in academic databases is a growing area of interest across disciplines, particularly in engineering. As ES methods become common in fields like engineering, researchers and information professionals must consider the functional nuances of their databases to develop complex, transparent search strategies. This summary and demonstration of three engineering databases (EV, IEEE Xplore, and ACM DL) adds to the growing literature aimed at increasing user understanding and, ideally, vendor and database administrator's understanding of their users' emerging needs. This need is only exacerbated by the impending AI revolution of information seeking and literature reviews. While generative AI offers a rapid alternative to traditional computer-aided search methods, it further obscures the users' understanding of how the information is selected, the characteristics of the total corpus from which that information was located, etc. This tradeoff may be suitable for some cases, but is unacceptable in others, such as ES reviews. Academic databases continue to offer significant value to advanced users who require sophisticated and transparent searching options.

References

[1] M. J. Page et al., "PRISMA 2020 explanation and elaboration: updated guidance and exemplars for reporting systematic reviews," BMJ, p. n160, Mar. 2021, doi: 10.1136/bmj.n160. Available: https://www.bmj.com/lookup/doi/10.1136/bmj.n160

[2] M. L. Rethlefsen et al., "PRISMA-S: an extension to the PRISMA Statement for Reporting Literature Searches in Systematic Reviews," Syst Rev, vol. 10, no. 1, p. 39, Jan. 2021, doi: 10.1186/s13643-020-01542-z. Available: https://systematicreviewsjournal.biomedcentral.com/articles/10.1186/s13643-020-01542-z

[3] M. Phillips, J. B. Reed, D. Zwicky, and A. S. Van Epps, "A scoping review of engineering education systematic reviews," J of Engineering Edu, p. jee.20549, Jul. 2023, doi: 10.1002/jee.20549. Available: https://onlinelibrary.wiley.com/doi/10.1002/jee.20549

[4] M. Borrego, M. J. Foster, and J. E. Froyd, "Systematic Literature Reviews in Engineering Education and Other Developing Interdisciplinary Fields," J of Engineering Edu, vol. 103, no. 1, pp. 45–76, 2014, doi: 10.1002/jee.20038. Available: https://onlinelibrary.wiley.com/doi/10.1002/jee.20038.

[5] M. Gusenbauer and N. R. Haddaway, "Which academic search systems are suitable for systematic reviews or meta-analyses? Evaluating retrieval qualities of Google Scholar,

PubMed, and 26 other resources," Research Synthesis Methods, vol. 11, no. 2, pp. 181–217, 2020, doi: 10.1002/jrsm.1378. Available: https://onlinelibrary.wiley.com/doi/10.1002/jrsm.1378

[6] D. Giustini and M. N. Kamel Boulos, "Google Scholar is not enough to be used alone for systematic reviews," OJPHI, vol. 5, no. 2, 2013, doi: 10.5210/ojphi.v5i2.4623. Available: http://journals.uic.edu/ojs/index.php/ojphi/article/view/4623

[7] M. J. Page et al., "The REPRISE project: protocol for an evaluation of REProducibility and Replicability In Syntheses of Evidence," Syst Rev, vol. 10, no. 1, p. 112, 2021, doi: 10.1186/s13643-021-01670-0. Available: https://systematicreviewsjournal.biomedcentral.com/articles/10.1186/s13643-021-01670-0

Appendix

Documentation for EV, IEEE, and ACM DL

Although some of the documentation for these databases can be behind subscription paywalls, included below are links to each database's main documentation for different types of searches and/or search features.

Engineering Village:

- "Quick" Search Information: https://service.elsevier.com/app/answers/detail/a_id/25632/supporthub/engineering-village/
- "Expert" Search Information: https://service.elsevier.com/app/answers/detail/a_id/25633/supporthub/engineering-village/
- Search field codes: https://service.elsevier.com/app/answers/detail/a_id/25673/supporthub/engineering-village/

IEEE Xplore:

- "Global" Search Information: https://ieeexplore.ieee.org/Xplorehelp/searching-ieee-xplore/global-search
- "Advanced" Search Information: https://ieeexplore.ieee.org/Xplorehelp/searching-ieee-xplore/advanced-search
- "Command" Search Information: https://ieeexplore.ieee.org/Xplorehelp/searching-ieee-xplore/command-search
- Taxonomy and Thesaurus: https://www.ieee.org/publications/services/thesaurus.html, version 1.03 as of January 2024

ACM DL:

- "Advanced" Search Information: https://dl.acm.org/search/advanced
- Training and Resources: https://libraries.acm.org/training-resources

Spreadsheet for Modifications to Produce More Consistent Searches

Screenshot of the EV tab of the spreadsheet for a preview - users can follow the below link to view and download a copy for themselves.

	A	B	C	D	E	F	G	H	I	J	K
1	Engineering Villiage: Adding proximity searching - set up as if you are searching Terms 1 NEAR Terms 2 with each column of terms being related and separated by OR										
2	Note: currently set up for many terms in column C, but add more columns after column I if more terms in column D (make sure to pay attention to use of $'s in equations)										
3											
4			**Terms 1**	**Terms 2**		**Terms 1 Prox Term 2a**	**Terms 1 Prox Term 2b**	**Terms 1 Prox Term 2c**	**Terms 1 Prox Term 2d**		
5	**Proximity Operator with N if desired**		Engineering	Libraries		Engineering NEAR/1 Libr	Engineering NEAR/1 Libr	Engineering NEAR/1 Libr	Engineering NEAR/1 Librarians		
6	NEAR/1		Sciences	Library		Sciences NEAR/1 Librarie	Sciences NEAR/1 Library	Sciences NEAR/1 Libraria	Sciences NEAR/1 Librarians		
7			Math	Librarian		Math NEAR/1 Libraries	Math NEAR/1 Library	Math NEAR/1 Librarian	Math NEAR/1 Librarians		
8			Technology	Librarians		Technology NEAR/1 Libra	Technology NEAR/1 Libra	Technology NEAR/1 Libra	Technology NEAR/1 Librarians		
9											
10											
11											
12	**Final String:**										
13	Engineering NEAR/1 Libraries OR Engineering NEAR/1										
14	Library OR Engineering NEAR/1 Librarian OR Engineering										
15	NEAR/1 Librarians OR Sciences NEAR/1 Libraries OR										
16	Sciences NEAR/1 Library OR Sciences NEAR/1 Librarian OR										
17	Sciences NEAR/1 Librarians OR Math NEAR/1 Libraries OR										
18	Math NEAR/1 Library OR Math NEAR/1 Librarian OR Math										
19	NEAR/1 Librarians OR Technology NEAR/1 Libraries OR										
20	Technology NEAR/1 Library OR Technology NEAR/1										
21	Librarian OR Technology NEAR/1 Librarians										
22											
23											

https://bit.ly/EngDBmodsheet