Evaluation synthesis analysis can be accelerated through text mining, searching, and highlighting: A case-study on data extraction from 631 UNICEF evaluation reports

Authors:
Lena Schmidt[1, 2], Pauline Addis[1, 2], Erica Mattellone[3], Hannah O'Keefe[1, 2], Kamilla Nabiyeva[3], Uyen Kim Huynh[3], Nabamallika Dehingia[3], Dawn Craig[1, 2], Fiona Campbell[1, 2]

1: National Institute for Health and Care Research Innovation Observatory, Newcastle University, Newcastle, UK

2: Population Health Sciences Institute, Newcastle University, Newcastle, UK

3: UNICEF, 3 United Nations Plaza, New York, USA

Corresponding author:

Lena Schmidt, email: lena.schmidt@io.nihr.ac.uk

3 Helix, Science Square, Newcastle upon Tyne NE4 5TG, United Kingdom

+44 (0) 191 208 2259

Keywords:

Evaluation Synthesis, Automated Data Extraction, Text Mining, Impact Evaluation

Abstract

Background: The United Nations Children's Fund (UNICEF) is the United Nations agency dedicated to promoting and advocating for the protection of children's rights, meeting their basic needs, and expanding their opportunities to reach their full potential. They achieve this by working with governments, communities, and other partners via programmes that safeguard children from violence, provide access to quality education, ensure that children survive and thrive, provide access to water, sanitation and hygiene, and provide life-saving support in emergency contexts. Programmes are evaluated as part of UNICEF Evaluation Policy[1], and the publicly available reports[2] include a wealth of information on results, recommendations, and lessons learned.

Objective: To critically explore UNICEF's impact, a systematic synthesis of evaluations was conducted to provide a summary of UNICEF main achievements and areas where they could improve, as a reflection of key recommendations, lessons learned, enablers, and barriers to achieving their goals and to steer its future direction and strategy. Since the evaluations are extensive, manual analysis was not feasible, so a semi-automated approach was taken.

Methods: This paper examines the automation techniques used to try and increase the feasibility of undertaking broad evaluation syntheses analyses. Our semi-automated human-in-the-loop methods supported data extraction of data for 64 outcomes across 631 evaluation reports;[3] each of which comprised hundreds of pages of text. The outcomes are derived from the five goal areas within

[1] E/ICEF/2023/27 (undocs.org) (last accessed 06/08/2024)
[2] Evaluation reports | UNICEF Evaluation (last accessed 06/08/2024)
[3] https://www.unicef.org/evaluation/reports (last accessed 06/08/2024)

UNICEF 2022-2025 Strategic Plan. For text pre-processing we implemented PDF-to-text extraction, section parsing, and sentence mining via a neural network. Data extraction was supported by a freely available text-mining workbench, SWIFT-Review. Here, we describe using comprehensive adjacency-search-based queries to rapidly filter reports by outcomes and to highlight relevant sections of text to expedite data extraction.

Results: While the methods used were not expected to produce 100% complete results for each outcome, they present useful automation methods for researchers facing otherwise non-feasible evaluation syntheses tasks. We reduced the text volume down to 8% using deep learning (recall 0.93) and rapidly identified relevant evaluations across outcomes with a median precision of 0.6. All code is available and open-source.

Conclusions: When the classic approach of systematically extracting information from all outcomes across all texts exceeds available resources, the proposed automation methods can be employed to speed up the process while retaining scientific rigour and reproducibility.

Strengths and limitations of this study

- Systematic impact evaluation syntheses are a vital tool to critically evaluate and plan future work of organisations such as UNICEF; but they are often not feasible due to the size, structure, and amount of evaluation report documents.
- To increase feasibility of analysis we describe a semi-automated human-in-the-loop system which was applied in a synthesis of 631 evaluations across 64 outcomes.
- The proposed open-source code and methods made an evaluation synthesis feasible by reducing text and streamlining the identification of relevant reports for each outcome.
- By making code open-source and adaptable we aim to encourage accelerated, yet transparent and reproducible results.
- While the methods cannot produce 100% complete or correct results for each outcome, they present useful automation methods for researchers facing otherwise non-feasible evaluation syntheses tasks.

Introduction

Background

The United Nations Children's Fund (UNICEF) is the United Nations agency dedicated to promoting and advocating for the protection of children's rights, meeting their basic needs, and expanding their opportunities to reach their full potential. They achieve this working with governments, communities, and other partners via programmes that safeguard children from violence, provide access to quality education, ensure that children survive and thrive, provide access to water, sanitation and hygiene, and provide life-saving support in emergency contexts.

For this broad portfolio of programmes, UNICEF publish comprehensive evaluation reports to disseminate outcomes of their interventions and to accelerate results for children. These evaluation reports include a wealth of information on what has been achieved, enabling and hindering factors, recommendations, and lessons learned in the implementation of specific UNICEF projects. As such, the evaluations often are stand-alone, project- or programme-focussed reports, while some also represent efforts to synthesise existing evaluations. Between 2018 and 2023 UNICEF have published 875 such evaluation reports[4].

Evaluation reports are commonly published by aid and non-profit organisations to communicate the methods and outcomes of projects. The International Initiative for Impact Evaluation (3ie), for example, maintains an online portal of more than 13,000 impact evaluations spanning sectors such as health, education, energy, or social protection[5]. The United Nations Development Programme (UNDP) maintains a database with more than 6000 evaluation reports and an in-built analysis platform[6]. Oxfam, a British group of non-governmental organizations, maintains a database of 88 impact evaluations[7]. UNICEF's evaluation function covers thematic, humanitarian, real-time, country level and syntheses aimed at assessing impact, efficiency, and effectiveness of its programs and maintains a digital database of a broader range of evaluative reporting[8] (see Figure 1). UNICEF

[4] https://www.unicef.org/evaluation/reports#/?&gerosRating=(blank),Not%20Rated,Missing,Unsatisfactory,Fair,Satisfactory,Highly%20Satisfactory&yearofCompletion=2023,2022,2021,2020,2019,2018 (last accessed 06/08/2024)
[5] https://developmentevidence.3ieimpact.org/ (last accessed 06/08/2024)
[6] http://web.undp.org/evaluation/media-centre/blogs/aida-2.shtml (last accessed 06/08/2024)
[7] https://policy-practice.oxfam.org/keyword/impact-evaluation/ (last accessed 06/08/2024)
[8] https://www.unicef.org/evaluation/reports#/ (Last accessed 01/07/2024)

evaluation reports are published in English, French, Spanish, or Portuguese and commonly include an executive summary, background, methodology, conclusions, lessons learned, recommendation for future programmes, references and multiple appendices with additional text, data, and tables in order to be transparent about the work that was carried out in the scope of each project. Therefore, they often exceed 150 pages of plain text, as commonly seen with reports from other governing bodies or leading organisations that conduct this scale of work. This complicates manual secondary analysis, due to the large amounts of unstructured, and potentially irrelevant, data. To increase the feasibility of analysing such a large, dataset in a timely, yet also reproducible manner, we developed a methodology that includes tagging and text-mining approaches to rapidly identify relevant data within the reports and expedite analysis.

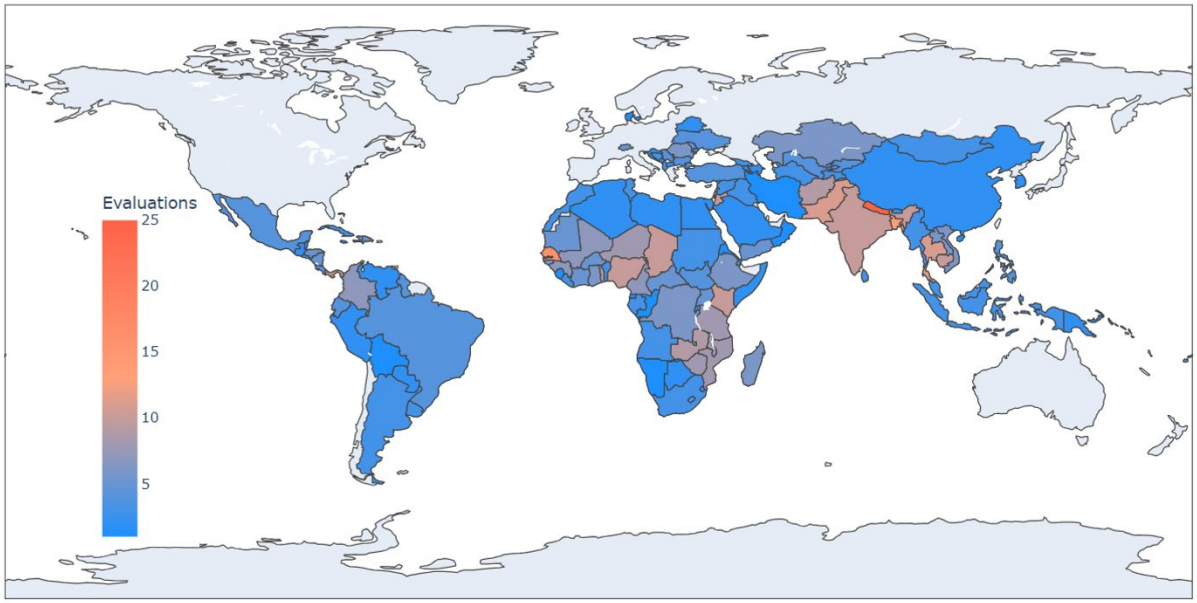

Figure 1: Number of UNICEF evaluation reports published by each country office, visualising the global spread of programmes

Aim

The aim of this paper is to demonstrate how text mining was used in a human-in-the-loop system to make an evaluation synthesis feasible and sustainable, while keeping its scope as broad as possible.

We describe the process of synthesising information from UNICEF reports, with respect to 64 outcomes related to five goal areas formulated as part of the Strategic Plan for 2022-2025: 'Every child survives and thrives', 'Every child learns', 'Every child is protected from violence and exploitation', 'Every child lives in a safe and clean environment', and 'Every child has access to inclusive social protection and lives free from poverty' (UNICEF, 2022). Outcomes focus on specific problems within each goal area, for example school attendance rates, access to safe drinking water, or domestic violence. The present paper focuses on semi-automation methods that made this analysis feasible. All code and trained models are available here: https://github.com/NIHRIO/EvaluationSynthesisMethods.

Sustainability and applicability

The automation and text-mining methods described in this paper were developed and tested as a case-study within UNICEF evaluation reports. These reports were created across more than 120 different country offices, they vary in length, and describe project across a diverse range of topics, such as sanitary infrastructure, cash transfers, vaccinations, or IT infrastructure. Reports from other

organisations such as 3ie or Oxfam similarly are likely to differ in length, structure, and content, hence we have aimed to create generalisable methods that are easily adaptable. Additionally, by making the code and text-mining methods open-source, and by using third-party software that is free to use and available to the general public, we have tried to ensure that our methods are available to other researchers who wish to apply them to synthesise similar report datasets.

Methodology

The following section introduces the dataset and automation pipeline in our case-study. We divided the process into several steps. For each step we provide open-source Python code on GitHub, developed as part of the case-study, to encourage adaptations and use in different research contexts. Figure 2 displays the methodology and number of reports included in each stage, in the form of a flowchart.

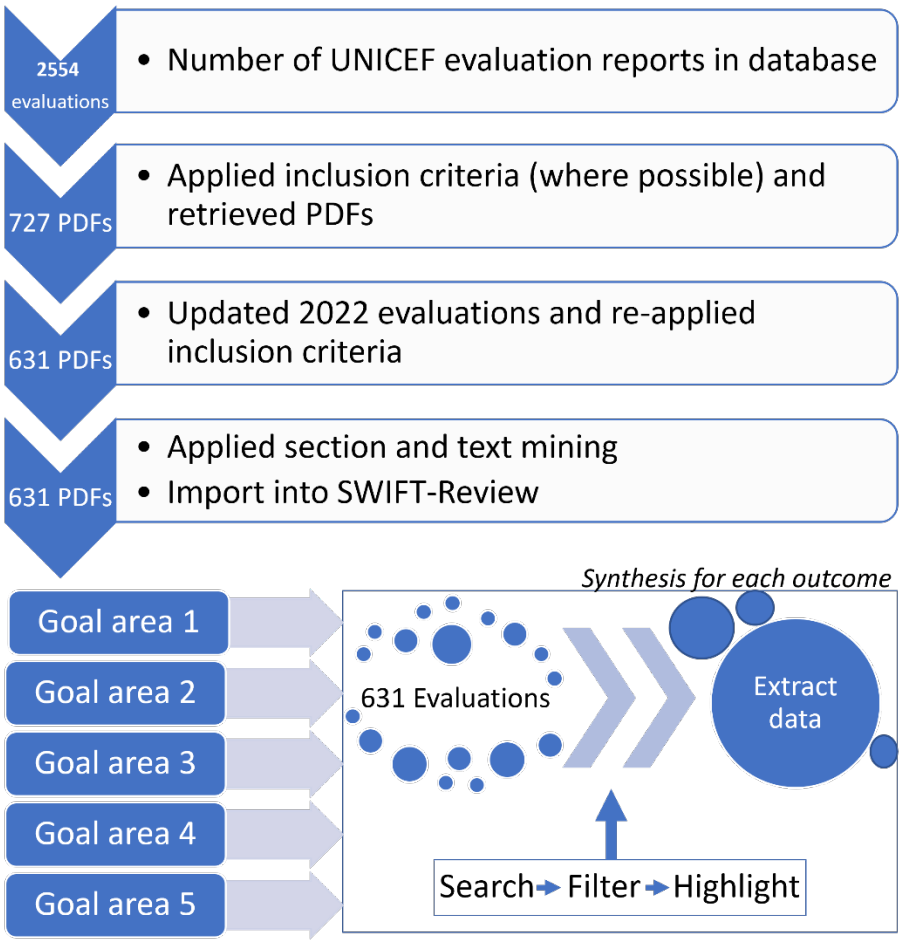

Figure 2: Methodology flowchart, displaying the number of evaluation reports included in each stage of the analysis

Dataset

PDFs of 727 reports were obtained with the help of UNICEF, from their public evaluation reports database[9]. We included all reports that fulfilled the following inclusion criteria:

1. Published between 2018-2022
2. Reports previously rated as 'Satisfactory' or better, using UNICEF's *evaluation quality-assessment system*; GEROS [10]

After updating the data with the latest 2022 evaluations and re-applying the inclusion criteria, 631 evaluation reports remained within the final dataset.

[9] https://www.unicef.org/evaluation/reports (last accessed 06/08/2024)
[10] https://www.unicef.org/evaluation/documents/global-evaluation-reports-oversight-system-geros-handbook-and-summary-2017 (last accessed 06/08/2024)

Automation methodology

The automation steps are as follows:

1. Bulk PDF-to-text conversion
2. Identification of relevant sections of text using rule-based methods and mining sentences using a neural network
3. Use of SWIFT-Review (Howard et al., 2016) to filter reports and highlight text for data extraction. Additionally, we provide a script to support the development of comprehensive adjacency-based search strategies
4. Automatic translation of non-English text

1. Bulk PDF-to-text conversion

We used the Python package *pypdf*[11] to convert report PDFs into text. First, the script accesses a folder where PDFs are saved. It then iterates through all files and attempts to read them as PDFs. Next, it accesses the PDFs page-by-page, extracts text and saves it as plain text file in a different folder. In rare cases there may be PDF processing errors, to mitigate this limitation our script records the names of affected files.

2. Identification of relevant sections of text using rule-based and neural methods

Rule-based methods:

We tested automatically extracting executive summary, lessons, and recommendation sections as these were deemed to be most likely to include relevant information. Initially, a rule-based approach was tested: matching text and extract sections by identifying words that would appear in a section heading, including synonyms and translations into French, Spanish, and Portuguese. For example, a case-insensitive regular expression search for '(lessons)|(lições)|(leçon)|(lecciones)|(good practice)' in the vicinity of a numbered item and carriage returns might indicate a section header for 'Lessons Learned'. However, it is very challenging to identify the end of a section, as some but not all reports include subsections or multiple paragraphs. Due to this problem, and additional variability in section header names, the resulting text for lessons and recommendation sections was not of sufficient quality.

Executive summaries, however, were identified automatically using the case-insensitive regular expression '(executive summary)|(RESUMEN EJECUTIVO)|(RÉSUMÉ EXECUTIF)|(Résumé exécutif)|(SOMMAIRE EXECUTIF)|(Resume executif)'. Due to this section's reliable placing at the beginning of a report and a limitation to 12 pages of text, we were more confident to use automatically extracted text from this section for the analysis. We were unable to identify 22% of the executive summaries automatically due to variations in wording or structure and quality of the PDF documents. These executive summary sections were extracted and added to the dataset manually.

More advanced text-mining from full texts to raise data quality:

One researcher spent around three hours processing ten randomly-chosen evaluations and identifying relevant sentences describing 'Enablers', 'Barriers', 'Lessons Learned', 'Recommendations', and 'Background' from their respective sections within the report's full text. The first four categories include the target information that is useful for this project, while the 'Background' class includes a mixture of undesirable text, such as table of contents, abbreviations, introduction, or methods sentences.

[11] https://pypdf.readthedocs.io/en/stable/

Using the sentences for each of these categories, a neural network based on the transformer architecture (Devlin et al., 2018) and a previously published model called 'SPECTER' (Arman et al., 2020) was trained to identify relevant sentences of each category. A random split of 60% of the data were used for training, and the rest for evaluation. This dataset had limitations: being very small, labelled rapidly, and containing classes with senses that may be ambiguous or overlap. However, this makes it an excellent test case for applying text mining methods on future projects analysing large amounts of unstructured grey literature, to maximise research outputs with respect to very tight timeframes and low resources available for analysis.

As this model was trained exclusively on English data, 162 non-English evaluations that were previously identified by UNICEF to be either French (n=96), Spanish (n=63) or Portuguese (n=3) were translated using the freely available Google Translate API and the *googletrans* python package[12]. All documents were then split into sentences. Pre-processing methods such as removing encoding errors, page breaks, and sentences shorter than 20 characters were applied to reduce noise in the dataset. Every sentence passed through the classifier and sigmoid layer and received a prediction of the likelihood of belonging to each class, thus creating a multi-class multi-label prediction scenario (see Figure 3). When we applied the model to the full dataset, for each evaluation document the 30 sentences with the highest probability scores were chosen.
To increase sensitivity, for each category some additional sentences were chosen, for example for 'Lessons learned', the filters 'need to', 'may', 'lesson' were applied to all sentences and the 30 most likely sentences containing each term were also added to the final dataset if they were not already contained within the model predictions. The ordering was based on the model's predicted probabilities for this class, highest first.

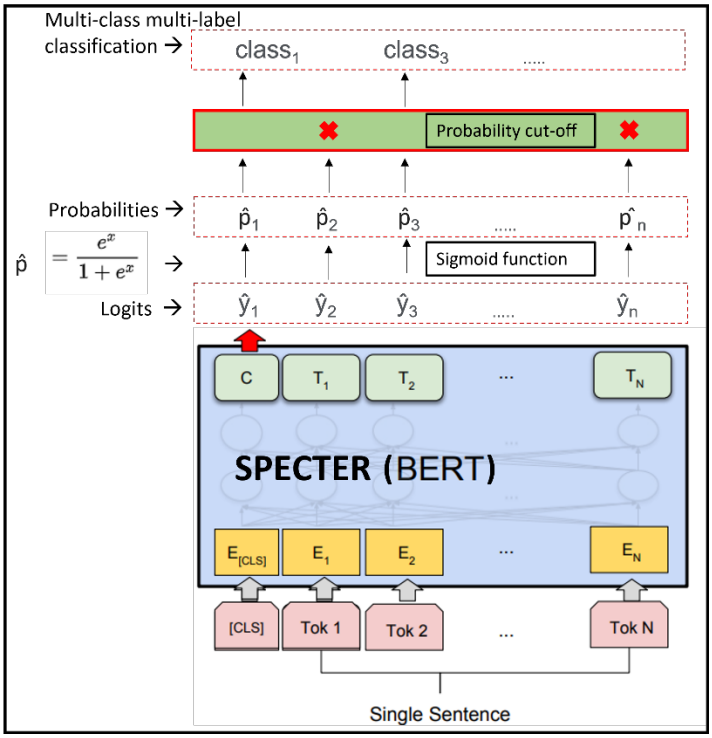

Figure 3: Neural network architecture. The input is embedded using the pretrained SPECTER network. We use it to obtain probabilities and use a cut-off value, thus creating a multi-class multi-label prediction scenario. Figure adapted from Devlin et al. (2018).

[12] https://pypi.org/project/googletrans/ (last accessed 16/04/2024)

3. Semi-automating data extraction with SWIFT-Review

After the text-mining step, each original evaluation report was reduced to a plain text file including an abstract-length short description provided by UNICEF, the report's 12-page executive summary, and mined sentences. We created a RIS file that contained the content of these text files within the abstract field, as well as metadata for each report in the form of keywords. This allowed us to import the data into third-party software for further analysis.

To facilitate the full analysis of the final 631 evaluation reports for each of the outcomes, we imported the data into a SWIFT-Review project. SWIFT-Review is a freely available text-mining workbench that facilitates searching, keyword highlighting, and topic modelling (Howard et al., 2016).

We created comprehensive tagging strategies for each outcome, to rapidly identify relevant information within each report. SWIFT-Review includes advanced search functionalities to tag or highlight text in fields such as title, abstract (which in our case included all text data described earlier), keywords, and pre-processed versions of the text such as stemmed versions. It also allows large Boolean searches, combining clusters of terms with 'AND', 'OR', and 'NOT' operators, wildcards, and adjacency searches to find documents containing target terms within a distance of N words.

Due to the complexity of our outcomes of interest, and the length of the documents, we opted against using simple Boolean searches to tag documents. For example for outcome 1.8 "*Percentage of surviving infants who received (a) first dose and (b) three doses of diphtheria, tetanus and pertussis (DTP) vaccine (WHO)*" using search terms *('diphtheria' OR 'tetanus' OR 'lockjaw' OR 'pertussis' OR 'whooping cough') AND ('vaccine' OR 'vaccinated' OR 'immunisation' OR 'immunization' OR 'jab' OR 'injection')* would have meant that every document using these terms anywhere would have been selected; even if they were not in direct context. To increase precision of our results we instead set the search up to use adjacency searching, thus combining every word in the first Boolean arm with each word in the second arm and allowing a default of up to 5 words between target terms.

For each outcome, between one to three search arms were devised to search, filter, and highlight data for extraction. A data scientist and an information specialist created the initial versions, and a senior reviewer responsible for the final data extraction reviewed and extended them to maximise sensitivity of the results. To save time, a python script was used to combine the terms from each arm of the search into an adjacency search query on SWIFT-Review's "tiab_stemmed" field. This field includes a pre-processed version of the text where only word stems are used to match text, thus reducing the need for wildcards. After generating and running the searches on our set of 631 reports we adjusted them as needed, either by decreasing the default 5 words for the adjacency to decrease the number of hits, or by skimming results from outcomes with a high number of hits to remove terms that appear to retrieve irrelevant information.

This led to search strategies such as *'tiab_stemmed:"adolescents school dropped out"~5'* to filter evaluations. That search, in practice, filters all evaluation where the words "adolescents school dropped out" and their grammatical variations appear within the proximity of 5 words of each other, eg. in "The scholarship programs and monitoring of children and **adolescents** who have **dropped out** of **school**".

Figure 4 shows how the search for outcome 1.8 was applied to the 631 documents in SWIFT-Review. The search itself was pasted into the query field on the top left corner of the screen. After executing the search, SWIFT-Review matched 12 evaluation documents that are selectable on the bottom half of the screen. After selecting a document, SWIFT displays the document text in the top right corner,

with yellow highlights applied on the words that matched the search. Due to overlaps and similar content between outcomes, we grouped some together for screening and data extraction, reducing the total number of separate outcomes to 34 (data shown in Appendix 1). For example, outcomes 1.8 and 1.9 were combined (1.8: 'Percentage of surviving infants who received (a) first dose and (b) three doses of diphtheria, tetanus and pertussis (DTP) vaccine (WHO)', and 1.9: 'Percentage of surviving infants who received first dose of the measles-containing vaccine'). This allowed us to rapidly filter and skim relevant evaluations for each of the outcomes while avoiding duplication of effort.

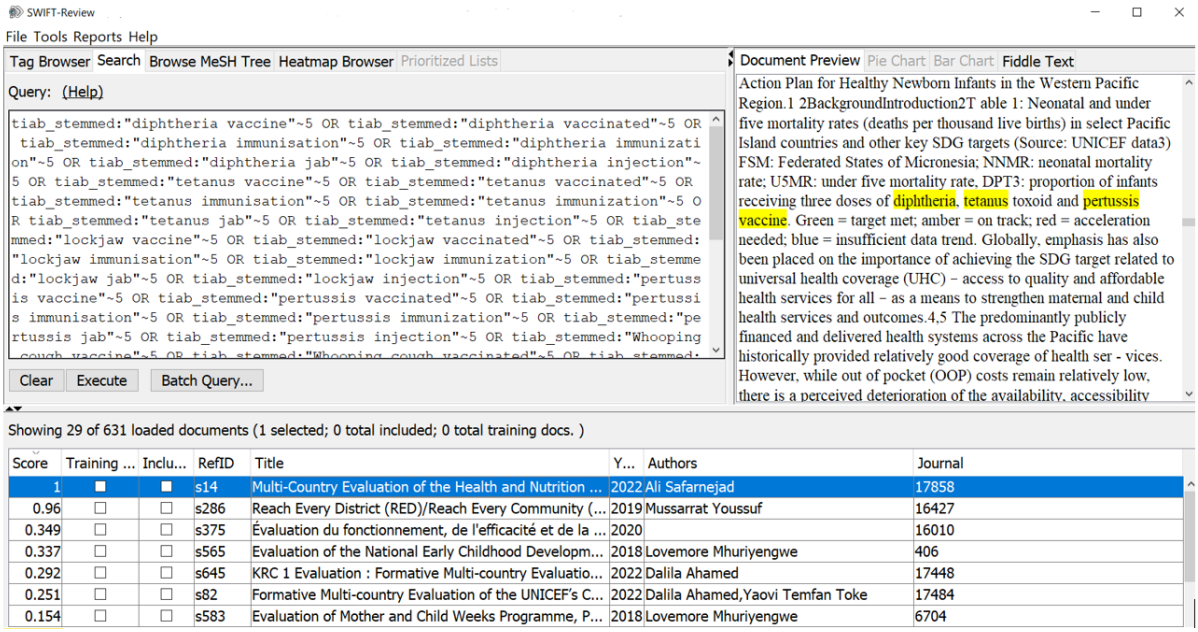

Figure 4: Example output filtering and highlighting 29/631 evaluations for outcome 1.8 and 1.9

4. Dealing with multiple languages and automating translation

UNICEF's metadata for each report indicated the presence of French (n=96), Spanish (n=63) or Portuguese (n=3) evaluations. The neural network supporting the identification of 'Lessons' and 'Recommendation' sentences was trained using English text. Therefore, we used the GoogleTranslate API via the python package googletrans[13] to translate these documents into English before predicting sentences.

In the SWIFT-Review-supported part of the project we trialled two approaches to handle non-English data.

Dataset 1: We used googletrans to translate any non-English text wherever this was possible. This included the abstract and executive summary sections, as well as the previously translated mined lessons and recommendation sentences from the full text.
Dataset 2: We used the original language abstracts and executive summaries. For the additional mining of lessons/recommendation sentences we had to use translated English text because the neural network was only trained on English data.

UNICEF then provided translations for all our terms in the goal area (GA) 2 outcomes searches. We re-created the search strategies for these terms in French, Spanish, and Portuguese to compare results for filtering.

[13] https://pypi.org/project/googletrans/

Steps:

1. Run GA 2 searches in English on Dataset 1 (English translations)
2. Run GA 2 searches in French/Spanish/Portuguese on Dataset 2 (original French/Spanish/Portuguese)
3. Check if the translated searches from step 2 bring up any unique new hits that would be missed had we only used Dataset 1
    a. If yes - consider translating terms for all GAs moving forward
    b. If no - consider working with English/translated data only

We manually reviewed all evaluations that were filtered by the original language searches, but 'missed' by the translated search. Data is shown in Appendix 2. Errors mostly occurred due to American/British English variations and ambiguous words such as the french ‘cours’ which can mean ‘lesson’ or ‘during’ and thus identify false-positive documents. Given the reasonable results of the error analysis we decided to use only automatically translated English text going forward. This led to time savings by avoiding the re-creation of all searches in three more languages and meant that no additional researchers were needed to perform data extraction from the majority of non-English language texts. To avoid missing data, we adapted the existing searches to include American and British spelling variations.

Results

The evaluation results for our sentence classifier to extract lessons and recommendations are shown in Table 1.

Table 11: Evaluation of the text-mining model, assigning positive labels at a probability threshold of 0.2. High recall (i.e. sensitivity) shows that the model is able to identify 93% of the relevant sentences for lessons and recommendations on the independent test set. We used the classifier only to predict these two classes downstream. The first three columns show quantitative evaluation results. The last column, 'Support', indicates the number of labelled evaluation samples from the held-out dataset that was used to calculate results.

Class	Precision	Recall	F1-score	Support
Enablers	0.00	0.00	0.00	10
Barriers	0.00	0.00	0.00	17
Lessons learned	0.29	0.93	0.44	40
Recommendations	0.53	0.93	0.68	45
Background	0.44	1.00	0.61	67

We then applied the trained neural network to the texts of all 631 evaluations. For each sentence, we transformed the neural network's output into a vector that contained the probabilities of this sentence belonging to each of the five classes. We selected the top 30 sentences for the recommendation and lessons classes each. Training for 'Enablers' and 'Barriers' was unsuccessful due to the low amount of positive training examples and ambiguity within them. Additionally, we filtered all sentences from each report using a keyword list for the recommendation and lessons class. This would lead to a theoretical maximum of 120 sentences for each report when considering both lessons and recommendation classes. However, the final number was usually lower than 120, for example obtaining 69 out of 2602 sentences for report 405. In the following, we report mean and standard deviation (SD) number of sentences that were retained by our algorithm. The reports included an average of 1662 sentences (SD 875), an average of 69 sentences (SD 21) were identified as 'Lesson Learned'. For 'Recommendations', the average was 79 sentences (SD 22).

A merged version of lessons and recommendations was created, leading to an average of 143 unique sentences per report (SD 39). This text-mining step therefore reduced the total text volume down to 8% of its original size, when taking the average of 1662 sentences in a full report as baseline. We appended these mined sentences to the executive summaries for each report.

Using vocabulary provided by UNICEF and selected outcomes from each goal area and research output of interest, we created comprehensive search strategies to filter and highlight documents within SWIFT-Review. This included proximity searches to expand the result set, as well as structured data imported from a UNICEF database to tag reports by year or quality ratings, among other variables. We first applied this methodology to goal area 2 outcomes in a pilot. Given the positive results, we decided to move forward with this approach in the full analysis of all outcomes. We recorded which evaluation reports were screened and included in the analysis of each outcome and visualised the country of their respective UNICEF evaluation office in Figure 5. This indicates that our synthesis effort managed to include an even spread of reports on a global scale, despite using text-mining and filtering to accelerate data extraction.

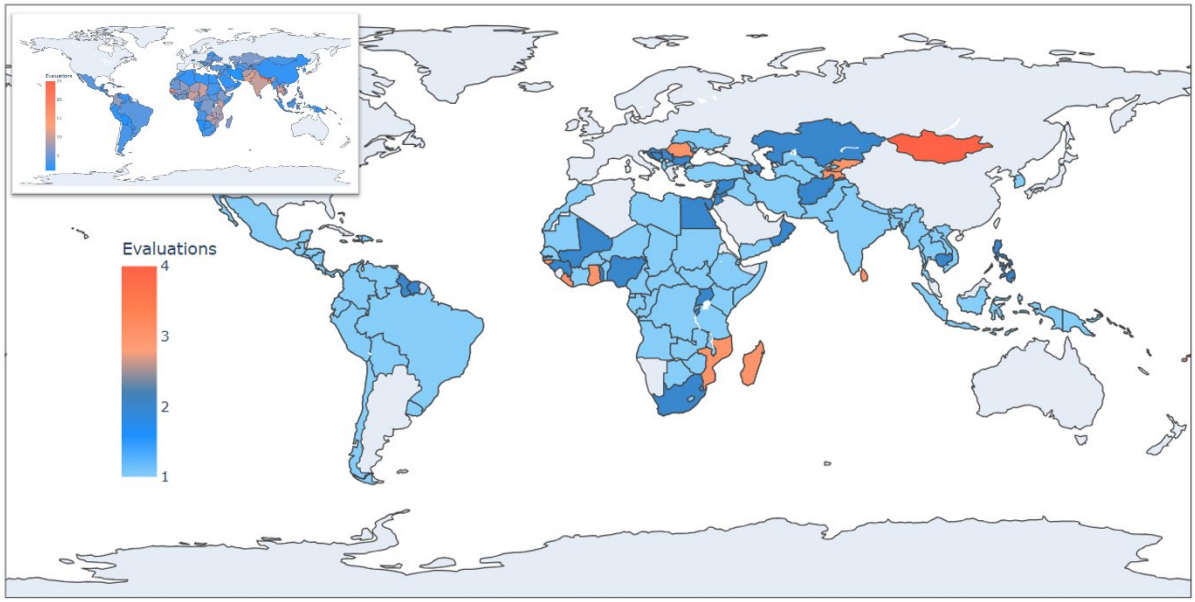

Figure 5: Global spread of evaluations filtered via SWIFT-Review and included in the synthesis of all 64 outcomes in five goal areas. For reference, a thumbnail of Figure 1 is shown in the top-left corner, visualising the actual global spread of all 631 included

Automatic searching and highlighting text from report summary sections for each goal area helped to reduce the overall workload of the analysis. This approach enabled researchers to quickly skim highlighted text of the most likely relevant evaluations, and to discard irrelevant evaluations. For evaluations with relevant highlights this approach allowed us to screen information more quickly, taking approximately 10-20 minutes per evaluation report as measured during a pilot phase for goal area 2 outcomes, although this could be up to 30 minutes when the original PDF had to be consulted. Among all outcomes, the precision of the filtering and tagging in SWIFT-Review was 0.52, having screened a total of 730 evaluations with 386 true positive evaluations that were included in the analysis. On a per-outcome basis, precision values ranged between 0.08-1, with a median of 0.6 (data shown in Appendix 1).

Discussion

Related literature and analysis tools

There are limited examples of tools or semi-automated methods to evaluate bodies of evidence such as these. We identified one tool developed by the United Nations Development Programme (UNDP), which is one of the six United Nations programmes.[14] The UNDP maintains a database with more than 6000 evaluation reports. This database contains an in-built analysis platform that shares key traits with our proposed methods. Released in 2022, Artificial Intelligence for Development Analytics (AIDA)[15] makes the plain text of UNDP evaluation reports accessible down to paragraph-level and provides features such as automatic summarisation. It contains evaluations tagged with Sustainable Development Goals and other thematic keywords. AIDA lets the user filter results by country or publication year, among other options. As such, AIDA serves the same purpose as our proposed methodology, by supporting researchers to filter, summarise, and visualise information. However, key differences are that AIDA is integrated into the UNDP infrastructure, not open-source, and thus not available to analyse other datasets. While it supports keyword searching, it does not support complex search queries or highlighting of results. We are currently unaware of any other tools in this space, but equally are aware that internally facing tools are often difficult to find.

Strengths and limitations

The main strength of the rapid evaluation synthesis methods presented in this paper is that they present a resource-efficient way of analysing an extremely broad evidence base. Due to variations in natural language and complex report structures, we are not claiming that the results of data extraction are 100% complete. However, by using automatic translations, a human-in-the-loop system and systematic Boolean searches to filter the data, we add a degree of methodological rigour; promoting transparency and reproducibility within the process. The search strategies for each outcome are shared in Appendix 1, and the SWIFT-Review project, which can be opened with the free SWIFT-Review desktop application[16], is in Appendix 3. The programming code and automations we developed ourselves are available in a GitHub repository[17].

The main weakness on the methodological side, as mentioned above, is the chance of missing information during the data extraction process. This can happen at two distinct steps in the workflow. First, during the initial literature curation when only the most likely relevant information is retained. Secondly, due to natural language variations, information can also be missed when applying the searches and filtering in SWIFT-Review. Here, the project team needs to balance a trade-off between creating broad and sensitive searches (high sensitivity but low precision -> high workload) and high precision and restrictive searches (high precision but limited sensitivity -> low workload and more rapid synthesis). It is difficult to apply this balance in a consistent manner for all

[14] https://www.un.org/en/about-us/un-system (last accessed 06/08/2024)
[15] http://web.undp.org/evaluation/media-centre/blogs/aida-2.shtml and https://aida.undp.org/landing (last accessed 06/08/2024)
[16] https://www.sciome.com/swift-review/ (Last Accessed 25/07/2024)
[17] https://github.com/NIHRIO/EvaluationSynthesisMethods (last accessed 22/08/2024)

outcomes because the trade-off between sensitivity and precision will be different for each outcome and dependent on the overall amount of evidence for each research question.

On the practical side, the main weakness is that only part of our workflow uses freely available software with a user-interface. The rest of our method, although available as a python package, requires data science or programming experience. However, this encourages the formation of an interdisciplinary team of researchers and drives a team science mentality which is important when tackling global health challenges. The need for human oversight may be seen as another limitation. Some of the methods, for example identifying sections via a rule-based approach, require further tailoring to individual research projects and cannot be used out-of-the-box. For the automated sentence classification, some human labelling of relevant sentences is needed. While human involvement does require resources, it also helps to reduce automation errors which leads to more streamlined processing and resource reductions downstream.

Conclusion

While text-mining and filtering methods are not expected to provide 100% complete results, they can be used to expedite the analysis of complex documents, such as evaluation reports. The methodology presented in this paper is most useful when rapidly analysing a large body of documents, focussing on breadth and accuracy rather than depth and sensitivity of results. By selecting relevant text via an existing summary section (expert-led) and then supplementing it with text-mining (AI-supported) we cut down the amount of irrelevant text presented to human data extractors. Then, by employing comprehensive and systematic search strategies to filter documents for each outcome in a human-in-the-loop system we aimed to boost transparency and reproducibility in the overall process. We provide our code for PDF-to-text, section processing, text-mining, and automatic creation of comprehensive adjacency-based tagging strategies within a python package. We hope this will encourage uptake of automation methods to support researchers interested in synthesising impact evaluations, reports, or grey literature in general.

Integrating natural language processing (NLP) to synthesize UNICEF evaluation reports (or similar) will necessitate significant digital infrastructure advancements. Key among these is the adoption of more structured evaluation reports and standardized templates that will facilitate enhanced machine readability. These changes will require the implementation of unified formatting and consistent terminologies to ensure that NLP algorithms can accurately interpret and process the content. Additionally, transitioning to digital-first documentation practices will support automated data extraction, analysis, and synthesis, enabling more efficient generation of insights from the vast corpus of evaluations. This evolution will enhance the ability to rapidly distil critical findings, trends, and lessons learned, fostering more effective decision-making and resource allocation within UNICEF.

Data availability statement

All programming code for the automations described in this paper is available on GitHub: https://github.com/NIHRIO/EvaluationSynthesisMethods

The weights for the trained SPECTER model for UNICEF data are available here: https://drive.google.com/drive/folders/1-0VXJcY_GKBNq6-5GprPvwdnTc4Raud3?usp=sharing

The SWIFT-Review project is available as Appendix 3, it can be loaded and used using the free desktop application available here: https://www.sciome.com/swift-review/

Conflict of Interest

The authors declare no conflict of interest.

Funding statement

This project was funded by the UNICEF Evaluation Office. The UNICEF Evaluation Office operates independently within the organization, with a mandate to produce impartial and rigorous evidence that informs UNICEF's policies, advocacy efforts, and programmes. For further details, please refer to the revised evaluation policy, available at https://www.unicef.org/executiveboard/revised-evaluation-policy-unicef-srs-2023.

LS, PA, HO, DC, and FC were in part supported by the NIHR Innovation Observatory (National Institute for Health and Care Research (NIHR) [HSRIC-2016-10009/Innovation Observatory]). The views expressed are those of the author(s) and not necessarily those of the NIHR or the Department of Health and Social Care.

Author's contributions:

LS: Methodology, Data Curation, Formal Analysis, Software, Visualization, Writing – Original Draft Preparation

PA: Methodology, Data Curation, Investigation, Validation, Writing – Review & Editing

HO: Methodology, Data Curation, Validation, Writing – Review & Editing

EM: Conceptualization, Methodology, Writing – Review & Editing

KN: Methodology, Data Curation, Writing – Review & Editing

UKH: Methodology, Writing – Review & Editing

ND: Methodology, Data Curation, Writing – Review & Editing

DC: Methodology, Funding Acquisition, Writing – Review & Editing

FC: Conceptualization, Methodology, Data Curation, Investigation, Funding Acquisition, Writing – Review & Editing

References

Arman, C., Sergey, F., Iz, B., Doug, D., & Daniel, S. W. (2020). *SPECTER: Document-level Representation Learning using Citation-informed Transformers*

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*.

Howard, B. E., Phillips, J., Miller, K., Tandon, A., Mav, D., Shah, M. R., Holmgren, S., Pelch, K. E., Walker, V., Rooney, A. A., Macleod, M., Shah, R. R., & Thayer, K. (2016). SWIFT-Review: a text-mining workbench for systematic review. *Systematic Reviews*, *5*(1), 87. https://doi.org/10.1186/s13643-016-0263-z

UNICEF. (2022). *UNICEF Strategic Plan 2022–2025: Renewed ambition towards 2030*. https://www.unicef.org/reports/unicef-strategic-plan-2022-2025