

MeSH Suggester: A Library and System for MeSH Term Suggestion for Systematic Review Boolean Query Construction

Shuai Wang
University of Queensland
Brisbane, Australia
shuai.wang2@uq.edu.au

Hang Li
University of Queensland
Brisbane, Australia
hang.li@uq.edu.au

Guido Zucon
University of Queensland
Brisbane, Australia
g.zucon@uq.edu.au

ABSTRACT

Boolean query construction is often critical for medical systematic review literature search. To create an effective Boolean query, systematic review researchers typically spend weeks coming up with effective query terms and combinations. One challenge to creating an effective systematic review Boolean query is the selection of effective MeSH Terms to include in the query. In our previous work, we created neural MeSH term suggestion methods and compared them to state-of-the-art MeSH term suggestion methods. We found neural MeSH term suggestion methods to be highly effective.

In this demonstration, we build upon our previous work by creating (1) a Web-based MeSH term suggestion prototype system that allows users to obtain suggestions from a number of underlying methods and (2) a Python library that implements ours and others' MeSH term suggestion methods and that is aimed at researchers who want to further investigate, create or deploy such type of methods. We describe the architecture of the web-based system and how to use it for the MeSH term suggestion task. For the Python library, we describe how the library can be used for advancing further research and experimentation, and we validate the results of the methods contained in the library on standard datasets. Our web-based prototype system is available at <http://ielab-mesh-suggest.uqcloud.net>, while our Python library is at <https://github.com/ielab/meshsuggestlib>.

ACM Reference Format:

Shuai Wang, Hang Li, and Guido Zucon. 2021. MeSH Suggester: A Library and System for MeSH Term Suggestion for Systematic Review Boolean Query Construction. In *ACM International WSDM Conference (WSDM '23)*, February 27, 2023, Singapore. ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/3503516.3503530>

1 INTRODUCTION

Medical systematic reviews are high-quality and comprehensive literature reviews with respect to specific medical research questions. To achieve high effectiveness and efficiency in medical systematic reviews, a high-quality search on medical literature repositories such as PubMed and Cochrane is the first and most crucial step to gathering enough evidence to support or refute the hypothesis of

the review. However, these searches depend strongly on the quality of the search queries [5, 16]. A high-quality search query may help researchers to gather enough evidence at the minimum cost, as less irrelevant literature will be retrieved. This task is receiving increasing attention from the community [5, 8, 13–17]. The queries used in medical systematic reviews are typically Boolean queries (search terms are combined using 'AND', 'OR' and 'NOT') and often include terms from the Medical Subject Headings (MeSH) [1]. MeSH is a controlled vocabulary thesaurus arranged in a hierarchical tree structure (specificity increases with depth in a parent→child relationship, e.g., Anatomy→Body Regions→Head→Eye... etc.).

However, due to MeSH's large vocabulary size and the systematic review researchers being often unfamiliar with MeSH definitions, selecting suitable MeSH terms to use for a query is challenging. Methods for the automatic suggestion of MeSH terms given a query have been devised, with the Automatic Term Mapping (ATM) being currently deployed within PubMed [4]. These methods examine a keyword-based query as input (often containing also Boolean operators) and output one or more MeSH terms (sometimes directly in the context of the structured Boolean query). For example for the Boolean query TB[tiab] OR tuberculosis[tiab] OR MDR-TB[tiab] OR XDR-TB[tiab], ATM suggests the MeSH term extensively drug-resistant tuberculosis[MeSH].

In this demonstration paper we build upon our previous work on effective methods for MeSH term suggestion [15, 17] and release a library with associated prototype web system (service and front-end) that implements a number of MeSH term suggestion methods, including ATM and neural methods. The library and web service can be integrated into search services that seek to help users creating Boolean queries for medical systematic reviews, e.g. searchRefiner [13] or PubMed itself. The library can also be used by others wanting to develop new MeSH term suggestion methods as the library is fully extensible and already includes standard evaluation resources (datasets, measures, baselines). The web front-end can be used by researchers wanting to demonstrate their MeSH term suggestion methods, or by users that want to identify the most effective MeSH terms for a query.

2 MESH TERM SUGGESTION METHODS

Our library currently implements six MeSH Term suggestion methods from two broad families of methods: Lexical (the first three below) and Neural (the remaining):

- (1) **ATM** refers to the method currently deployed as part of PubMed for mapping free text into MeSH Terms, journal names or author names. Mapping occurs through the use of rules and mapping tables. We use the ATM implementation available through the PubMed Entrez API [10].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

WSDM '23, February 27, 2023, Singapore

© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-9599-1/21/12...\$15.00
<https://doi.org/10.1145/3503516.3503530>

- (2) **MetaMap** refers to using the MetaMap tool [2] to identify medical concepts in queries; the concepts that include entries from the MeSH hierarchy are then used as suggestions.
- (3) **UMLS** refers to searching through a purposely built search service we setup based on Elasticsearch v7.6. The index consists of UMLS concepts [3]: these include MeSH Terms. The search is performed by issuing the free-text query to the service; the retrieved items are filtered by only including items of the type "MeSH Terms"; a cut-off may be applied to the resulting ranking.
- (4) **Atomic-BERT** refers to ranking MeSH Terms using the underlying dense retriever to rank MeSH terms with respect to each keyword in the query; we then return the top-ranked MeSH Term.
- (5) **Fragment-BERT** refers to performing Atomic-BERT, but before selecting the MeSH Term to suggest, the rankings of the individual query keywords are interpolated using normalised CombSUM rank fusion. The top-ranked MeSH Term is then returned.
- (6) **Semantic-BERT** is similar to Fragment-BERT, but the rank fusion is performed with respect to keyword groups rather than across all keywords. Keyword groups are identified based on similarity as computed by a word2vec model trained on PubMed. The top-ranked MeSH term for each Keyword group is then returned as the suggestion.

All Neural methods use our fine-tuned dual-encoder model described in previous work [17].

3 SYSTEM OVERVIEW

3.1 MeSH Term Suggestion Web Tool

We start by describing the web service and associated front-end that exposes the implemented MeSH Term suggestion methods. The architecture of the system is provided in Figure 1. The system consists of (1) the MeSH Term Suggestion API, which wraps the library implementing the suggestions methods, and described in Section 2, (2) the Web front-end, which allows users to enter their keyword queries and receive back the suggestions, (3) the Big Brother logging service, which captures and stores users interactions for subsequent analysis.

Apart from direct usage through the web front-end, we also provide an API for MeSH Term suggestions.

The MeSH Term suggestion API exposes to users the POST method to call the API that, provided a query, returns a list of MeSH Term suggestions using one of the implemented methods. The input format is shown in Figure 2, while the output of the call is shown in Figure 3. The API output includes the original keyword query input, the suggestion type (i.e. the method used to generate the suggestion), and the MeSH Terms suggested for each keyword or keywords group.

The web front-end is shown in Figure 4. Users can submit a single keyword or keywords combination and choose to use any of the methods outlined in Section 2. Upon submission of a query, the tool returns a list of candidate MeSH Terms that the user can copy or use the inbuilt tool to add to their free-text query to form a new query with MeSH terms (which they can eventually copy).

The interaction logging service, Big Brother [12], is integrated into our tool's front-end and captures all interactions of the users

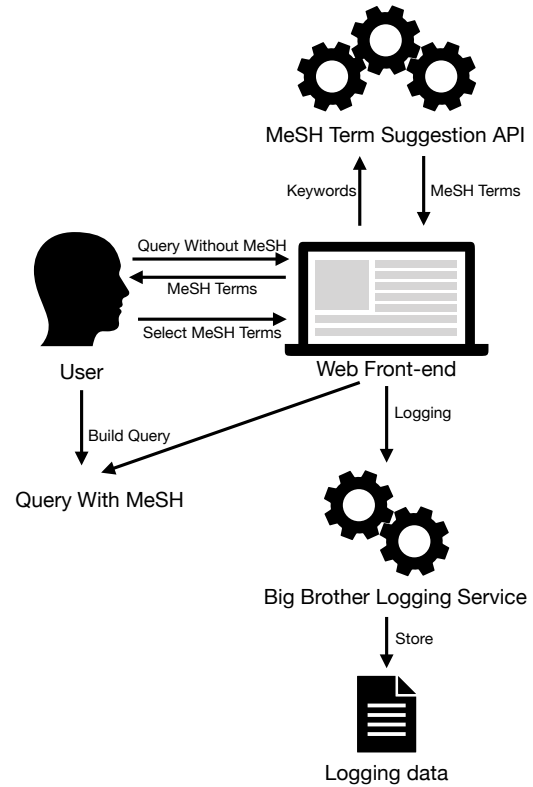


Figure 1: Architecture of web-based MeSH Suggestion tool

```
1 {"Keywords": [K1, K2, K3, ..., Kn],
2   "Type": Semantic/Atomic/Fragment}
```

Figure 2: Input format of the API POST call.

```
1 [
2 {"Keywords": [K1],
3   "Type": Semantic/Atomic/Fragment},
4 "MeSH_Terms": {0: M1, 1: M2, ..., 9: M10}
5
6 {"Keywords": [K2, K3],
7   "Type": Semantic/Atomic/Fragment},
8 "MeSH_Terms": {0: M1, 1: M2, ..., 9: M10}
9 ...
10 {"Keywords": [Km, ... Kn],
11   "Type": Semantic/Atomic/Fragment},
12 "MeSH_Terms": {0: M1, 1: M2, ..., 9: M10}
13 ]
```

Figure 3: Output format from the API POST call.

with the web page. The logging service can help with the future investigation of MeSH Term suggestion methods through user studies.

3.2 MeSH Term Suggestion Library

Along with the web service API and web front-end described above, we also provide a Python-based library package, `meshsuggestlib`, that implements the methods described in Section 2. The package

MeSH Term Suggestion Tool

Input Keywords Separated By \$ Sign
trisomy 21\$mosaicism\$tricuspid regurgitation

Semantic-BERT
 Fragment-BERT
 Atomic-BERT
 ATM
 MetaMap
 UMLS

SUGGEST

trisomy 21[t,lab] OR mosaicism[t,lab] OR tricuspid regurgitation[t,lab] OR Trisomy[MeSH]

Keyword: trisomy 21, mosaicism

MeSH Terms:

- **ADD** 0: Trisomy
- **ADD** 1: Mosaicism
- **ADD** 2: Karyotyping
- **ADD** 3: Chromosome Aberrations
- **ADD** 4: Chromosome Disorders
- **ADD** 5: Chromosomes, Human, Pair 21
- **ADD** 6: Sex Chromosome Aberrations
- **ADD** 7: Chromosomes, Human, 21-22 and Y
- **ADD** 8: Translocation, Genetic
- **ADD** 9: Monosomy

Keyword: tricuspid regurgitation

MeSH Terms:

- **ADD** 0: Tricuspid Valve Insufficiency
- **ADD** 1: Tricuspid Valve
- **ADD** 2: Mitral Valve Insufficiency
- **ADD** 3: Mitral Valve
- **ADD** 4: Mitral Valve Stenosis
- **ADD** 5: Tricuspid Valve Stenosis
- **ADD** 6: Aortic Valve Stenosis
- **ADD** 7: Aortic Valve Insufficiency
- **ADD** 8: Echocardiography, Transesophageal
- **ADD** 9: Heart Valve Diseases

Figure 4: Example Mesh Term suggestion using our Web tool.

also makes available classes that can be extended for the implementation of new MeSH Term suggestion methods. Finally, the package includes data and associated auxiliary code for evaluating MeSH Term suggestion methods. These inclusions allow others to quickly implement, validate and compare new MeSH Term suggestion methods. For example, the results for the Semantic-BERT MeSH Term suggestion methods on the CLEF TAR 2017 dataset [6], which we show in Table 2, can be obtained by running the following commands:

```
python -m meshsuggestlib
--model_dir model/checkpoint-80000/
--method Semantic-BERT
--dataset CLEF-2017
--output_file out.tsv
--email sample@gmail.com
--interpolation_depth 20
--depth 1
```

	Input Name	Description
Basic	<i>method</i>	Predefined MeSH Term Suggestion method or new method
	<i>dataset</i>	Pre-defined Dataset or data folder name
	<i>mesh_file</i>	MeSH Term file path
Neural	<i>mesh_encoding</i>	[Optional] path of encoded MeSH Terms
	<i>tokenizer_name_or_path</i>	Tokenizer for Neural Methods
	<i>model_dir</i>	Neural Model path or name
	<i>q_max_len</i>	query keyword maximum length after tokenization
Group	<i>p_max_len</i>	MeSH Term maximum length after tokenization
	<i>semantic_model_path</i>	Path of w2v Model for semantic grouping
	<i>interpolation_depth</i>	Cut-off of each keyword for interpolation
PubMed	<i>depth</i>	Cut-off for number of MeSH Term retrieved for each group
	<i>output_file</i>	Path of query result output
	<i>date_file</i>	Path of date restriction file for each topic
Evaluate	<i>email</i>	Email for calling E-utilities API for literature retrieval
	<i>evaluate_run</i>	Whether evaluate the output result
	<i>qrel_file</i>	Path to file containing relevance judgments

Table 1: Input option for library.

		2017			2018		
Dataset		P	F1	R	P	F1	R
Lexical	Method						
	Original	0.0303	0.0323	0.7694	0.0226	0.0415	0.8629
	ATM	0.0225	0.0215	0.7109	0.0306	0.0535	0.8225
	MetaMap	0.0323	0.0304	0.7487	0.0335	0.0590	0.8085
Neural	UMLS	0.0325	0.0300	0.7379	0.0326	0.0573	0.7937
	Atomic-BERT	0.0252	0.0243	0.7778	0.0283	0.0479	0.8452
	Semantic-BERT	0.0254	0.0243	0.7784	0.0309	0.0526	0.8404
	Fragment-BERT	0.0343	0.0325	0.7414	0.0388	0.0690	0.8034

Table 2: Effectiveness of MeSH Term suggestion methods in terms of precision(P), F1 and recall (R). No statistical significance is detected between the Original query and those obtained by other methods (two-tailed t-test with Bonferroni correction, $p < 0.05$).

Similarly, these results can be evaluated with simplicity using the following template command:

```
python -m meshsuggestlib
--evaluate_run
--output_file out.tsv
--qrel qrel.qrels
```

Table 1 reports a full list of input options for `meshsuggestlib`. For the neural models we implemented, it is possible to change the underlying model checkpoint used, although currently only dense retrievers (bi-encoders) are supported. Nevertheless, it is possible for researchers to extend the package by implementing new MeSH Term suggestion methods, or adding new evaluation datasets; we show how one can add a new suggestion method in Section 4.2.

4 CASE STUDIES

Next, we report on a small-scale validation of the methods we implement in `meshsuggestlib` and the associated web tool; then, we describe how the library can be expanded by implementing new MeSH Term suggestion methods.

4.1 Evaluation of Methods

We evaluate all implemented methods on the CLEF Tar 2017 [6] and 2018 [7] datasets. For each topic in the dataset, we stripped the original Boolean query of the Boolean operators and the MeSH terms, so to obtain a keyword query which was then used as input

for the MeSH Term suggestion methods. We then attach the suggested MeSH Terms to the query and use this to retrieve documents from the PubMed index. Evaluation is performed with respect to how effective the query was for retrieval – the better the query, the more effective the MeSH Term suggestion method. Note, this is a retrieval task, not a ranking task, as queries and the underlying retrieval system are Boolean. Also note that the original query is likely to outperform the automatic queries: this is because these queries have undergone careful manual intervention by information specialists. We refer to our previous work for more details of the evaluation setup [17].

Results are reported in Table 2, where we also include the results obtained on the original Boolean query (which includes MeSH terms added by information specialists).

Results differ from our recent evaluation of these methods (see [17]) because: (1) We issue our constructed query to PubMed's E-Utilities API [10] to retrieve documents for evaluation; some PubMed articles may be changed or updated; thus may be filtered out by the Boolean keywords or date restrictions. (2) For the Lexical methods, we use the PubMed API for ATM, and the UMLS and Metamap for the other methods; the implementations and the data used by these methods may have received updates between the two undertakings of the experiments. For Neural methods, the encoder integrated in the library has been retrained and thus may differ from that originally used in previous work because of small differences in initial weights and training process. (3) The evaluation is conducted using the `ir_measures` toolkit [9] instead of `Trec_eval` because of its better fit into our Python library – these two tools have minor differences in how recall is computed. Despite these aspects, the trend we observe from the results is the same, and the differences between our reproduced results and previous experiments are marginal.

4.2 Add a new MeSH Term Suggestion Method

`meshsuggestlib` allows researchers to implement new suggestion methods. If these methods are based on the neural architecture used by our methods, then it is sufficient to change the library input parameters `tokenizer_name_or_path` and `model_dir` to direct the library to the new dense retriever models. If instead the underlying retrieval logic differs, to add a new method it is sufficient to implement the search function `user_defined_method` in the `NeuralSuggest` class. This function takes keywords, the retriever models and lookups as input and returns a list of keywords and MeSH Term IDs pairs as output. At inference, the use of the method 'NEW' will automatically call this function.

5 CONCLUSION & DISCUSSION

This demonstration contributes useful tools for the MeSH Term suggestion task: a library that implements common lexical baselines and neural methods, and a web service with associated web front-end that allows end-users to use these methods to augment their queries for systematic review literature search. The tool also allows to collect usage and interaction logs, thus allowing researchers to further their understanding of MeSH Term choices and the query formulation process [11]. The library also integrates an evaluation pipeline, including the implementation of accessory methods for standard datasets in this context: this lowers the barrier for others to research new MeSH Term suggestion methods.

Several improvements are currently planned for the tool. A key feature to further streamline use of the tool is the automatic decomposition of Boolean queries and the related extraction of keywords, which are then used as input to the MeSH Term suggestion methods. Another avenue of improvement is integrating the library into existing Boolean query visualisation tools, like SearchRefiner [13], which allows users to interpret how the choices made with respect to the MeSH terms suggested affect retrieval and effectiveness.

Acknowledgement. This research is supported by the Australian Research Council (DP210104043). The authors of the work also wish to thank Dr Harrison Scells for providing instruction and suggestions.

REFERENCES

- [1] 2019. Introduction: What is MeSH? <https://www.nlm.nih.gov/bsd/disted/meshutorial/introduction/02.html> [Online; accessed 20. Jan. 2020].
- [2] Alan R Aronson. 2001. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program.. In *Proceedings of the AMIA Symposium*. American Medical Informatics Association, 17.
- [3] Olivier Bodenreider. 2004. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic acids research* 32, suppl_1 (2004), D267–D270.
- [4] Beth G Carlin. 2004. Pubmed automatic term mapping. *Journal of the Medical Library Association* 92, 2 (2004), 168.
- [5] Scells Harrison and Zuccon Guido. 2018. Generating Better Queries for Systematic Reviews. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval* (Ann Arbor, MI, USA) (SIGIR '18). ACM, New York, NY, USA, 475–484.
- [6] E. Kanoulas, D. Li, L. Azzopardi, and R. Spijker. 2017. CLEF 2017 Technologically Assisted Reviews in Empirical Medicine Overview. In *CLEF'17*.
- [7] Evangelos Kanoulas, Rene Spijker, Dan Li, and Leif Azzopardi. 2018. CLEF 2018 Technology Assisted Reviews in Empirical Medicine Overview. In *CLEF 2018 Evaluation Labs and Workshop: Online Working Notes, CEUR-WS*.
- [8] Grace E. Lee and Aixin Sun. 2018. Seed-driven Document Ranking for Systematic Reviews in Evidence-Based Medicine. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval* (Ann Arbor, MI, USA) (SIGIR '18). ACM, New York, NY, USA, 455–464. <https://doi.org/10.1145/3209978.3209994>
- [9] Sean MacAvaney, Craig Macdonald, and Iadh Ounis. 2022. Streamlining Evaluation with `ir-measures`. In *European Conference on Information Retrieval*. Springer, 305–310.
- [10] Eric Sayers. 2010. A General Introduction to the E-utilities. *Entrez Programming Utilities Help [Internet]*. Bethesda: National Center for Biotechnology Information (2010).
- [11] Harrison Scells, Connor Forbes, Justin Clark, Bevan Koopman, and Guido Zuccon. 2022. The Impact of Query Refinement on Systematic Review Literature Search: A Query Log Analysis. In *Proceedings of the 2022 ACM SIGIR International Conference on Theory of Information Retrieval*. 34–42.
- [12] Harrison Scells, Jimmy, and Guido Zuccon. 2021. Big Brother: A Drop-In Website Interaction Logging Service. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, Canada) (SIGIR '21). Association for Computing Machinery, New York, NY, USA, 2590–2594. <https://doi.org/10.1145/3404835.3462781>
- [13] Harrison Scells and Guido Zuccon. 2018. searchrefiner: A Query Visualisation and Understanding Tool for Systematic Reviews. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 1939–1942.
- [14] H. Scells, G. Zuccon, B. Koopman, A. Deacon, S. Geva, and L. Azzopardi. 2017. A Test Collection for Evaluating Retrieval of Studies for Inclusion in Systematic Reviews. In *SIGIR'2017*.
- [15] Shuai Wang, Hang Li, Harrison Scells, Daniel Locke, and Guido Zuccon. 2021. MeSH Term Suggestion for Systematic Review Literature Search. In *Proceedings of the 25th Australasian Document Computing Symposium* (Virtual Event, Australia) (ADCS '21). Association for Computing Machinery, New York, NY, USA, Article 8, 8 pages. <https://doi.org/10.1145/3503516.3503530>
- [16] Shuai Wang, Harrison Scells, Justin Clark, Bevan Koopman, and Guido Zuccon. 2022. From Little Things Big Things Grow: A Collection with Seed Studies for Medical Systematic Review Literature Search. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Madrid, Spain) (SIGIR '22). Association for Computing Machinery, New York, NY, USA, 3176–3186. <https://doi.org/10.1145/3477495.3531748>
- [17] Shuai Wang, Harrison Scells, Bevan Koopman, and Guido Zuccon. 2022. Automated MeSH Term Suggestion for Effective Query Formulation in Systematic Reviews Literature Search. arXiv:2209.08687 [cs.IR]