Finding Evidence for Updates in Medical Guidelines

Roelof Reinders, Annette ten Teije and Zhisheng Huang
Department of Computer Science, VU University Amsterdam, The Netherlands
mail@roelofreinders.nl, annette@cs.vu.nl, huang@cs.vu.nl

Keywords: Medical Guidelines, Evidence-based Medicine, Medical Guideline Updates

Abstract: Medical guidelines are documents that describe optimal treatment for patients by medical practitioners based on current medical research (evidence), in the form of step-by-step recommendations. Because the field of medical research is very large and always evolving, keeping these guidelines up-to-date with the current state of the art is a difficult task. In this paper, we propose a method for finding relevant evidence for supporting the medical guideline updating process. Our method that takes from the evidence-based medical guideline the recommendations and their corresponding evidence as its input, and that queries PubMed, the world's largest search engine for medical citations, for potential new or improved evidence. We built a prototype and performed a feasibility study on a set of old recommendations, and compared the output to evidence for the newer version. The system succeeded in finding goal articles for 11 out of 16 recommendations, but in total, only 20 out of 71 articles were retrieved. Our ranking method for most relevant articles worked well for small result sets, but for large result sets it failed to rank the goal articles in the top 25 results.

1 Introduction

The field of medical science is very broad. But what it all comes down to in practice, is treating an individual patient suffering from a physical or psychological discomfort, and finding the optimal treatment to cure him or her. In order to help medical practitioners keep a clear view of how a patient should be treated, medical guidelines have been created. These medical guidelines describe the different steps that should be taken in helping a patient who suffers from certain symptoms, from diagnosis to treatment to aftercare. It is then up to the medical practitioner to follow this guideline, and to decide when to diverge from it based on the individual circumstances of the patient.

The concept of guidelines is built on what is known as evidence-based medicine (EBM). Sackett et al. (1996) describe this concept as *the conscientious, explicit, and judicious use of current best evidence in making decisions about the care of individual patients.* What this comes down to, is that medical practitioners should use the current strongest scientific evidence combined with their individual expertise to find the optimal treatment for their patients. Field and Lohr (1990) describe guidelines as *Systematically developed statements to assist practitioner and patient decisions about appropriate health care for specific clinical circumstances*.

Guidelines are usually created and maintained by (semi-)governmental organizations. An example of this is the National Guideline Clearinghouse[1], which contains a collection of guidelines maintained by the U.S. Department of Health & Human Services. Each individual guideline is created by a committee, of which all members must disclose any possible conflicts of interest. These committees have occasional meetings to discuss possible changes and updates to the guidelines.

1.1 Motivation

Because of the number and size of guidelines, these desirable updates can be difficult to identify. Relevant evidence might be overlooked or not fully recognized, causing suboptimal treatment quality. Also, the process of finding and identifying the evidence is a time-consuming task. This can cause the process between research being done and the results transferring into the guidelines to take longer than necessary.

Shelleke et al. (2001) defines several factors that could make updating a guideline desirable. These

[1] `http://www.guideline.gov/`

factors include ‘technical’ improvements found in research, but also more ‘societal’ factors, such as the change of values in a society, or the economical circumstances that could lead to preference for a certain intervention over another. The authors developed a model that should indicate whether a recommendation inside a guideline has to be updated. This is based on two steps: consultation of experts and literature research. If either of these steps indicate that changes are favorable, a panel of experts should judge whether the suggested changes are correct and see to it that they are implemented in the guideline.

Consultation of medical experts is useful, and also expensive in terms of time and knowledge. Because it is focused on human interaction (a guideline expert interviewing a medical expert), it is very difficult to improve upon in terms of resources.

Literature research is currently also expensive in terms of time, but slightly less so in terms of knowledge. A large part of the work, namely the gathering of new relevant articles, could be performed by computers. While the results of this search should still be processed by human experts to determine their relevance, giving automatic support to the task of guideline updating by indentifying relevant new articles (evidence) from PubMed could lead to major time benefits. This project aims to develop a system that can perform this task in an adequate manner.

1.2 Research goals

The goal of this project is to develeop a method that finds suggests evidence for updates in evidence-based medical guidelines, to implement a prototype, and to show the feasability of the method. More specifically, our research should answer the following question:

Is it possible to build an automated system that can improve the process of updating medical guidelines by performing literature search?

The answer will be based on the subquestions:

1. How can we extract useful search terms from a guideline recommendation and its evidence?
2. How can we use the search terms obtained from 1 to construct a relevant PubMed query?
3. How can the search results from our PubMed query be judged and ranked based on their relevance to the recommendation and their scientific strength.
4. How can the final search results and their ranking be evaluated in terms of their use in practice?

The rest of this paper is structured as follows. First, we briefly discuss other studies towards this goal and how we hope to improve on their results. In section 3, we propose our approach to accomplishing this. We will then evaluate our method by running it on recommendations from multiple guidelines. Finally, we describe our interpretation of the results and make suggestions for future research.

2 Related work

We discuss two approaches to perform a similar task from the literature. The first is a system by (Cohen, et al., 2012) that predicts whether a medical article can be used to update a systematic review of a research field. This approach uses a machine learning method based on Support Vector Machines that is trained on a dataset of pre-tagged articles. This led to good results in testing, where over 70% of all updates were recognized while maintaining a low alert rate. Even though this approach shows promise, there are definitely some downsides to it. Firstly, it requires an annotated set of articles, which requires a lot of human effort to assemble. Secondly, the resulting model trained by the SVM algorithm is still a black box. Even though the system has decent results, it is difficult to determine how it got these results and whether the method is generalizable. The second approach is a system developed by (Iruetaguena, et al., 2013). This system takes the referenced articles from a guideline, and then constructs a new set of articles by using the PubMed related articles search. Then for each article in this set, the PubMed related articles are taken again. This was done for multiple guidelines of which an older and a newer version were available. The approach found over 90% of all articles introduced in the new version (high recall), but the resulting set of articles was so large that only 0.07% of all articles found (low precision) were goal articles. Our approach improve on these two methods in multiple ways. We want our system to be generalizable, and not require a manually constructed or tagged set of articles. We keep our list of suggested articles small, so that it is easily processable by humans. We implement a ranking system that puts the most relevant articles high in the list of results.

Other work being done in this field aims at formalizing and digitalizing guidelines so that they are easier for computer programs to process. (Peleg, et al., 2003) describes several languages that are developed specifically for this cause. Our work fits very well to the idea of ‘living guidelines’: guidelines that are updated continuously, as for example described by (Seyfang, et al., 2007).

3 Approach

In this section, we describe the means we used to answer our research question. We first describe the external resources used. Then we propose our method, followed by detailed description of each step of the method.

3.1 MEDLINE and PubMed

MEDLINE is the largest online database of medical scientific articles. It is an online implementation of the MEDLARS (Medical Literature Analysis and Retrieval System) that was launched by the United States National Library of Medicine in 1964 (Rogers, 1964). We will use this database to retrieve articles used for evidence updates. One of the most powerful resources for categorizing medical articles in MEDLINE are the MEdical Subject Headings, or in short, MeSH terms. These terms are used as annotations to all articles in the MEDLINE database, and have been part of the MEDLARS design since its conception (Libscomb,2000). These annotations range from very specific to very broad, and are structured in the form of a tree, where broad terms can have more specific terms as their children. There are different subtrees for different topics, for instance 'Diseases' or 'Organisms', but also meta-terms such as 'Publication Characteristics'[2]. On top of the MEDLINE database, the PubMed search engine was developed. This is a very advanced search engine that offers many options beyond basic keyword search. Most advanced features are accessed by entering special parameters into the search field. Other features include searching for publication dates, journals, MeSH terms, and many others. A full list of search tags can be found in the PubMed Help book (PubMedHelp 2005). One of the features that makes PubMed so powerful, is the automatic recognition of these tags. Plain text that is given as input is automatically parsed by PubMed and annotated with semantic tags that help define the search query. For example, if a certain piece of text is recognized as the name of an author, the *[Author]* tag is automatically included in the query when it is executed. We use this for certain steps in our method.

Another key feature of PubMed, is that it allows access to most of its important features via the Entrez Programming Utilities, or E-Utilities in short. We use three of E-utilities functions, namely `ESearch`, that performs a query on the database, `EFetch`, that requests the summary of a given article, and `ELink`, that finds related articles for a given article. A full documentation of all features can be found in the E-Utilities online help book[3].

3.2 Method

The input for our method consists of a recommendation (in natural language), and the PubMed IDs of the articles that are used to support it. An example of a recommendation, including its evidence is shown below. This example corresponds to recommendation 1 in table 5.

Addition of radiotherapy following local excision of DCIS results in a significantly lower risk of local recurrence (this is valid for all subgroups). with the PubMed IDs 9469327, 12867108, 10683002, and 8292119. Based on this input, we take the following steps:

step 1: Parsing the recommendation

step 2: Processing the evidence

step 3: Constructing a PubMed query and executing the query

step 4: Grading and ranking the results

step 5: Generating the output to the user

We discuss each step in detail below.

3.2.1 Step 1: Parsing the Recommendation

The recommendation is a string of natural language, from which we want to extract as much useful information as possible. We use `ESearch` API from the E-Utilities of PubMed. As we described in section 3.1, the query processing system can automatically recognize certain terms. We make use of this by sending the recommendation string to the `ESearch` API, and extracting the recognized MeSH terms for the query that is returned. We use those MeSH terms for constructing the query (step 3).

3.2.2 Step 2: Processing the Evidence

To process the evidence articles, first a Python dictionary is created, with as its keys the article IDs for each evidence article, and as its value another dictionary containing information on the article that was extracted from its summary which is obtained by performing an `EFetch request`, including its title, abstract, and a list of MeSH terms used to categorize it. We perform a loop over the sets of MeSH terms and use them to create two sets. The goal of this process is to establish which terms are common between

[2] For a full overview of the MeSH tree, visit `http://www.nlm.nih.gov/mesh/trees.html`

[3] `http://www.ncbi.nlm.nih.gov/books/NBK25500/`

the articles, and are therefore useful for annotating the recommendation. The first will be referred to as 'primary terms', and contains terms that:

- Are used to categorize every piece of annotated evidence
- Are tagged as a 'Major Topic' in at least one piece of evidence

The set of 'secondary terms' contain MeSH terms that are used to categorize all but one piece of evidence.

3.2.3 Step 3: Constructing and Executing Queries

At this point, we have two pieces of information:

1. A set of terms that PubMed recognized in the recommendation text.
2. A set of primary terms and a set of secondary terms that were extracted from the evidence.

We use this information to create one or multiple PubMed queries that will bring us new relevant articles. It should be possible to make queries more broad (more answers) or more specific (less answers), depending on the number of results that is desirable. For this goal, we use two methods:

1. Constructing a query by combining sets of terms (result of step 1)
2. Constructing a query by selecting terms (result of step 2)

Constructing Queries Method by Combining Sets of Terms

This method is used in the case where we have a set of primary and a set of secondary terms. These sets are left in tact, but the variation lies in how they can be combined can make the resulting query more specific or more broad (lower level). The different levels of this combination, ranging from the most broad to the most specific, are shown in table 1.

For this method, the program starts by executing a query of level 4 (the most specific). This query is sent to PubMed with the `ESearch` method. If there are not enough search results, it tries again with a query from a broader level. This process continues until enough results are gathered.

For example the recommendation (14 from table 5):

> *"To minimise the need for a second operative staging procedure, intraoperative frozen section assessment can be used to diagnose malignancy and to exclude metastatic disease."*

Level	Query format (output)
0	Disjunction of the union of primary and secondary terms
1	Disjunction of primary terms
2	Conjunction of primary terms
3	(Conjunction of primary terms) AND (Disjunction of secondary terms)
4	(Conjunction of primary terms) AND (Conjunction of secondary terms)

Table 1: Different levels for combining primary and secondary terms.

The following MeSH terms were determined to be primary terms:

{Humans, Frozen Sections, Ovarian Neoplasms, Retrospective Studies, Female}

The following MeSH terms were determined to be secondary terms:

{Sensitivity and Specificity, Adolescent, Predictive Value of Tests, Middle Aged, Aged, 80 and over, Adult, Aged}

Table 2 shows the different queries constructed for each level. In the table the quotation marks, `MeSH Terms` strings added to each term, and search time ranges were removed for the sake of clarity.

Constructing Queries Method by Selecting Terms

The second method for constructing queries, is called 'Querying by Selecting Terms'. This method takes a single set of MeSH terms as its input. These terms are then combined into a conjunction, which is sent to PubMed as a query. If the query does not yield enough results, the least important of the MeSH terms is removed from the list, and the query is sent again. The difficult part here is determining which of the terms are the most and least important. For this task, we developed a method that is based on the MeSH subtrees. The MeSH vocabulary is divided into different categories, each indicating a different part of the medical domain. We ordered the different subtrees in terms of their relevance for constructing a query, the result of which is shown in Table 3. The list of terms is sorted by the relevance of their subtree, and for each query, the least important term is removed if there are not enough results.

3.2.4 Step 4: Grading and Ranking the Results

After a query with a large enough number of results, we have a list of potentially useful articles. To determine whether or not an article is useful for a possible guideline update, we have to determine its scientific strength. For evidence to be very strong, it has

Level	Query
4	Humans AND Frozen Sections AND Ovarian Neoplasms AND Retrospective Studies AND Female AND Sensitivity and Specificity AND Adolescent AND Predictive Value of Tests AND Middle Aged AND Aged, 80 and over AND Adult AND Aged
3	Humans AND Frozen Sections AND Ovarian Neoplasms AND Retrospective Studies AND Female AND (Sensitivity and Specificity OR Adolescent OR Predictive Value of Tests OR Middle Aged OR Aged, 80 and over OR Adult OR Aged)
2	Humans AND Frozen Sections AND Ovarian Neoplasms AND Retrospective Studies AND Female
1	Humans OR Frozen Sections OR Ovarian Neoplasms OR Retrospective Studies OR Female
0	Humans OR Frozen Sections OR Ovarian Neoplasms OR Retrospective Studies OR Female OR Sensitivity and Specificity OR Adolescent OR Predictive Value of Tests OR Middle Aged OR Aged, 80 and over OR Adult OR Aged

Table 2: Example of the different levels of queries as constructed by combining the sets of primary and secondary terms of the evidence.

Rank	Index	Description
1	C	Diseases
2	D	Chemicals and Drugs
3	A	Anatomy
4	B	Organisms
5	N	Health Care
6	M	Named Groups
7	E	Analytical, Diagnostic and Therapeutic Techniques and Equipment
8	F	Psychiatry and Psychology
9	G	Phenomena and Processes
10	H	Disciplines and Occupations
11	I	Anthropology,Education, Sociology and Social Phenomena
12	J	Technology, Industry, Agriculture
13	K	Humanities
14	V	Publication Characteristics
15	L	Information Science
16	Z	Geographicals

Table 3: List of MeSH subtrees ranked by importance

to describe randomized controlled trials, a type of research in which test subjects are separated into multiple groups. (Rosenfeld and Shiffman, 2009) describes a set of criteria that can be used to determine whether an article describes research of this form. Based on these criteria, each article in the result set is tagged with a boolean value that indicates whether it is strong or not. This method is described in detail in (Iruetaguena et al., 2013), and is also implemented in our system.

Next to an article's scientific strength, we want to determine its relevance to the guideline recommendation. Two techniques are applied for this. First, the term frequency/inverse document frequency (tf-idf) value is calculated for each article. This term was first coined by (Stalton and Buckley, 1988). To do this, a corpus of 50,006 article summaries was gathered by requesting related articles to our input articles on PubMed. A dictionary was created containing each word in the abstracts of these articles, as well as their relative number of occurrences. The words in the recommendation are then compared to the words in the abstract for each individual and the sum of the weights for terms that occur in both is taken. The resulting score is a measure for the article's relevance. The second measure to determine article relevance is the *Inverse MeSH distance*. To calculate this measure, the distance in MeSH tree branches between MeSH terms used to categorize each article and the MeSH terms extracted from the recommendation is calculated. This is based on the assumption that terms that are close to each other in the tree are more similar than ones that are far apart. The inverse of this value is taken, so that more similar articles receive higher grades. This computation is made only if two terms are in the same subtree of the MeSH vocabulary.

At this point we have three measures of relevance for an article:

1. The article's scientific strength s
2. The article's abstract's relevance to the recommendation r
3. The inverse MeSH distance to the recommendation d

We use those three measures to rank the articles. The score for each article is calculated using the following formula: $score = (5 \cdot s) + r + d$

s is given more weight than the other variables, because it turned out during testing that the article's scientific strength is a strong factor in indicating whether or not it can be referenced in a medical guideline. Of course other weights can be given to the different components. After the score for each article has been calculated, the list of articles is sorted by this score in descending order and presented to the user.

3.2.5 Step 5: Generating the output

The output of the algorithm is an HTML file presenting the program's results in a clear overview, sorted on ranking and with the calculated grades. The queries that were executed are also displayed, as well as the terms used to generate them from both the recommendation and the evidence. The level of the query is also stated. All articles and queries contain a hyperlink to directly access them on PubMed.

3.3 Implementation

The entire system is implemented in Python. All code was tested and confirmed to be working on Python 2.7.6, using one external library: the `xmltodict` library, that allows XML files to be loaded and interpreted as Python dictionaries. This was used to parse the results of the PubMed responses. The library is available from its website[4]. The source code is available on the author's GitHub account[5].

4 Experiments and Results

4.1 Experimental Set-up

For the feasibility study, several experiments were performed, all with real medical guidelines of which two versions were available: one recent version and an older version from a few years ago. Recommendations from these guidelines that were updated with new evidence between these versions were selected. These recommendations and their corresponding evidence were extracted and used as data to evaluate the system.

For the different types of queries that our program uses, we want to evaluate three metrics.

The *Recall* is the percentage of goal articles found by the query. Goal articles are articles that were added to a recommendation between the two versions of the guidelines, and that were thus used to update the recommendation.

The *Number of results* is the number of search results for each query. This value will be evaluated to get an indication of whether the program has managed to generate queries that are not too broad or too specific. Ideally, this number should lie between 10 and 200 for each query.

The *Top25-Recall* is the percentage of goal articles that were found by the program and that were ranked in the top 25 most relevant results by the ranking algorithm. This value should be compared to the *Recall*: if goal articles are retrieved, but not ranked highly, this would be an indication that the ranking algorithm is under-performing.

4.2 Gathering of Test Data

The recommendations used for these experiments were extracted from the guidelines shown in table 4 From these guidelines, recommendations had to be extracted by hand. This was done by reading both versions of the guideline and looking for sections on the same subject, where the evidence for a recommendation had changed between versions. This usually meant that there was a change in the text, as well as an improvement of the recommendation's grade. For each recommendation, the text of the older version was used as program's input. For the recommendations' evidence, all PubMed IDs were gathered by searching for the referenced articles manually. The PubMed IDs that occur in the new version of the guideline, but not in the old one, are identified as 'goal articles': these are the articles that we want our program to find.

The list of recommendations used is shown in table 5, together with their number of evidence articles in both the old and the new version. The full recommendations and their evidence and goal articles are supplied in `http://www.roelofreinders.nl/guidelineupdate/appendixa.pdf`.

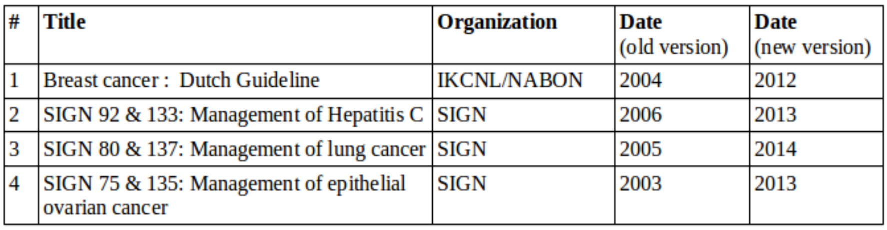
4.3 Finding the optimal search strategy

As described in section 3.2.3, we developed two ways of constructing PubMed queries from sets of MeSH terms. The sets were extracted from the recommendation text and the evidence articles. Both were created in such a way that they can be made more broad or more specific in terms of how many results they return. Now we want to compare how these methods compare to each other in terms of finding the greatest number of goal articles. For our experiments we use the query construction method by combining sets with different input sets of MeSH terms (techniques 1,2, 3), and the query construction method by selecting terms with different input set of MeSH terms (technique 4, 5, 6):

1. Query construction method by combination for just the MeSH terms from the recommendation
2. Query construction method by combination of the primary and secondary evidence set of MeSH

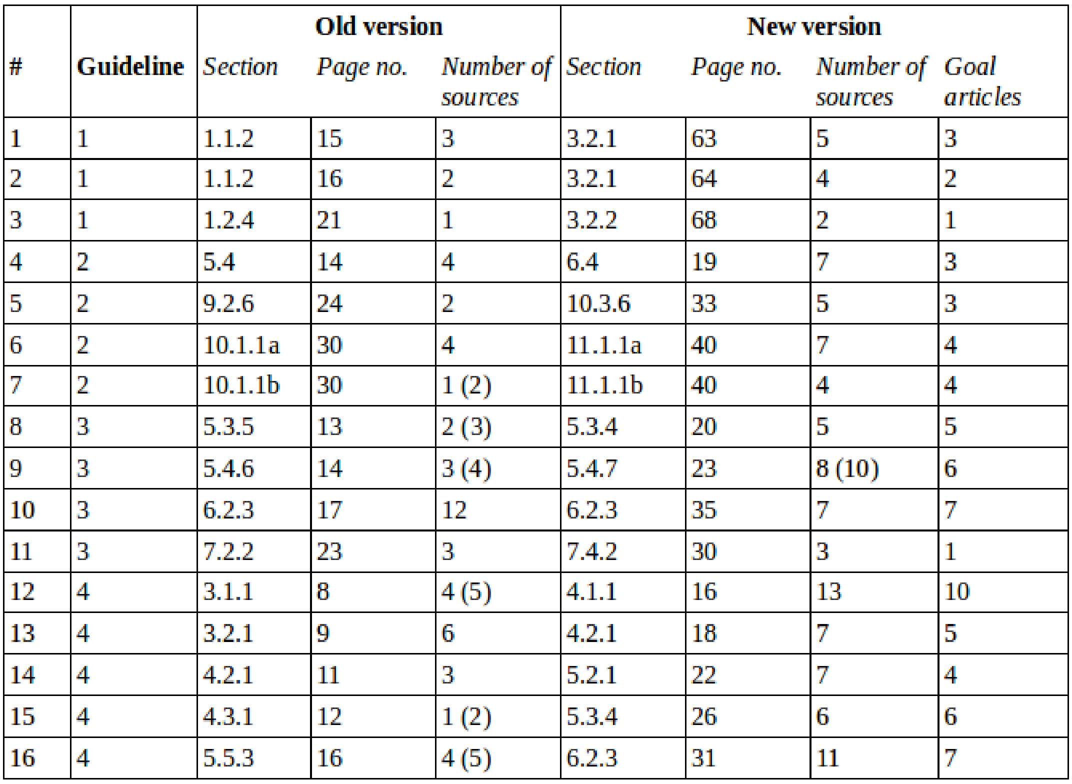
[4] See: `https://pypi.python.org/pypi/xmltodict`

[5] urlhttps://github.com/roelofreinders/guidelineupdate



#	Title	Organization	Date (old version)	Date (new version)
1	Breast cancer : Dutch Guideline	IKCNL/NABON	2004	2012
2	SIGN 92 & 133: Management of Hepatitis C	SIGN	2006	2013
3	SIGN 80 & 137: Management of lung cancer	SIGN	2005	2014
4	SIGN 75 & 135: Management of epithelial ovarian cancer	SIGN	2003	2013

Table 4: The guidelines used to gather test data.



#	Guideline	Old version			New version			
		Section	*Page no.*	*Number of sources*	*Section*	*Page no.*	*Number of sources*	*Goal articles*
1	1	1.1.2	15	3	3.2.1	63	5	3
2	1	1.1.2	16	2	3.2.1	64	4	2
3	1	1.2.4	21	1	3.2.2	68	2	1
4	2	5.4	14	4	6.4	19	7	3
5	2	9.2.6	24	2	10.3.6	33	5	3
6	2	10.1.1a	30	4	11.1.1a	40	7	4
7	2	10.1.1b	30	1 (2)	11.1.1b	40	4	4
8	3	5.3.5	13	2 (3)	5.3.4	20	5	5
9	3	5.4.6	14	3 (4)	5.4.7	23	8 (10)	6
10	3	6.2.3	17	12	6.2.3	35	7	7
11	3	7.2.2	23	3	7.4.2	30	3	1
12	4	3.1.1	8	4 (5)	4.1.1	16	13	10
13	4	3.2.1	9	6	4.2.1	18	7	5
14	4	4.2.1	11	3	5.2.1	22	7	4
15	4	4.3.1	12	1 (2)	5.3.4	26	6	6
16	4	5.5.3	16	4 (5)	6.2.3	31	11	7

Table 5: The recommendations and their evidence extracted from the guidelines. Not every evidence article could be found on PubMed. The number of evidence articles that were retrievable from PubMed are listed in the table, and the actual number of articles is shown between parentheses. The guideline number refers to the guideline number of table 4.

terms

3. Query construction method by combination of the union of the MeSH terms of the recommendation and the primary evidence MeSH terms, and the secondary evidence MeSH terms
4. Query construction method by selecting terms for just the recommendation terms
5. Query construction method by selecting terms for the primary evidence terms
6. Query construction method by selecting terms for the recommendation MeSH terms and the primary evidence MeSH terms combined

Both query construction methods will construct a broader query until the desired number of articles. For the experiments the minimum number of search results was set to 15 for each technique. For the grading and ranking of the results we use at most 1000 results, in other words the maximum number of results from PubMed was set to 1000. The results where sorted by PubMed based on their relevance to the query. This means that result 1001 and onward will be ignored by the ranking algorithm and ignored for its results. This number was chosen because it seems to be on the edge of what a laptop can handle, and we do not want to overburden the PubMed servers by requesting thousands of articles.

The percentage of goal articles found per recommendation in the at most 1000 results, as well as the total number of search results returned by techniques 1 to 3 are shown in table 6. The results for techniques 4 to 6 are shown in table 7.

Overall, 20 out of 71 goal articles were found. Looking at the results more closely, we can note the following findings:

1. For 5 out of 16 recommendations, the system was unable to find any of the goal articles with any of the techniques (recommendations 3, 5, 12, 13 and 15)
2. Technique 2 found the most articles overall, closely followed by technique 3. Both techniques work by the query construction method by combining sets of terms.
3. The techniques 4 to 6, which are based on constructing queries by selecting terms, perform much worse.
4. The query constructing method by combining sets of terms (technique 1-3) performs better overall, but yields many more search results. This indicates that the query captures the recommendations' meaning the best, but is not very specific in doing so.

If we want to explain finding 1, we have to take a closer look at recommendations 3, 5, 12, 13 and 15. For recommendation 3, 5 and 15, this can be explained by the small amount of evidence articles for the recommendation (1, 2 and 1 respectively). Notable about recommendations 12 and 13 is that their updates seem to radically change the recommendation itself. This could be the reason that the system was unable to find any goal articles: the goal articles are simply too different from the original evidence.

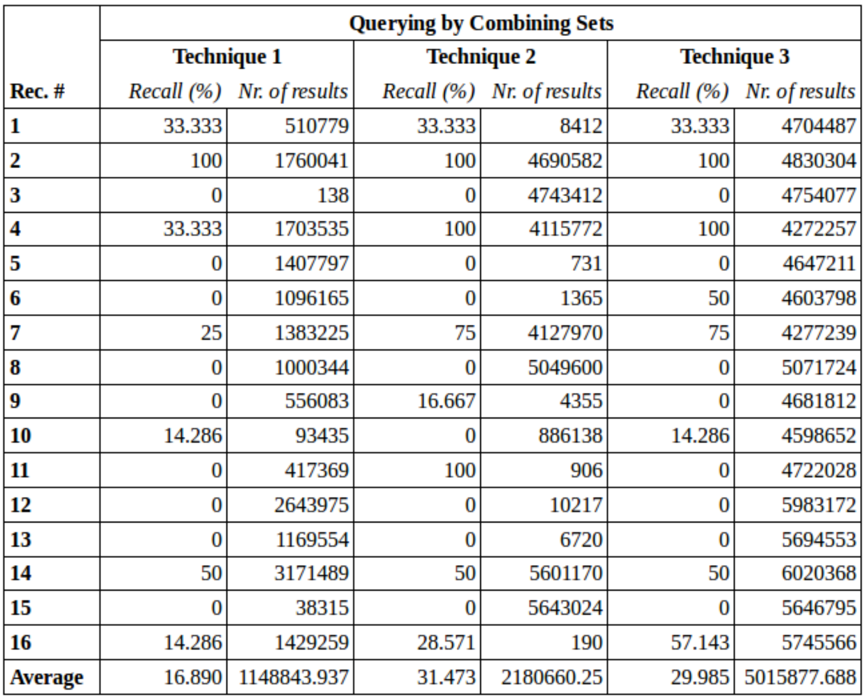
To elaborate on finding 2, we have to look more closely at the techniques used. Both are based on constructing query method by combining sets of terms, and both use the primary MeSH terms extracted from the evidence articles as their input (for technique 3, these are augmented with the MeSH terms of the recommendation). In practice, most queries that reach the threshold of 15 articles are of level 1: they are a disjunction of all primary terms, as explained in section 3.2.3. This explains why the number of search results is so large, as disjunctions are very weak restrictions on the set of articles. The fact that there are still a significant number of goal articles found, indicates that the PubMed search engine is quite potent at sorting articles by relevance to the search terms, since only the first 1000 were used.

We can also see that constructing queries by selecting terms (technique 4-6) reaches one of the goals for which it was designed, which is decreasing the search space. This can be seen by the number of search results, which is much lower on average than when using the constructing queries by combining sets of terms (technique 1-3) . This approach is, however, much worse at finding the goal articles. This indicates that removing search terms in order to broaden the query can lead to a loss in meaning, causing worse results.

The large number of results returned by the queries indicates how volatile queries can be. Even though our approach offers a lot of variation between broad and specific queries, small changes such as removing a term or switching from conjunction to disjunction can result in an explosion in the number of results obtained by the query. This is a difficult problem to solve due to the size of the database, and requires further research.

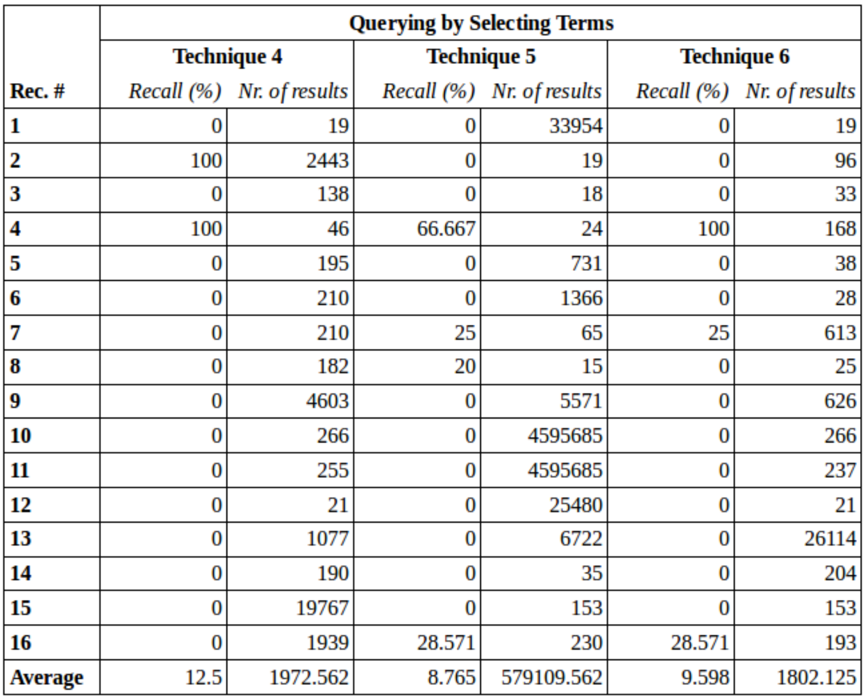
4.4 Results for Ranking

Now that we have an indication of how well our queries perform, we will examine how well the ranking algorithm performs in determining their relevance. To do this, we will take a look at our best performing technique (1-6) for each recommendation,



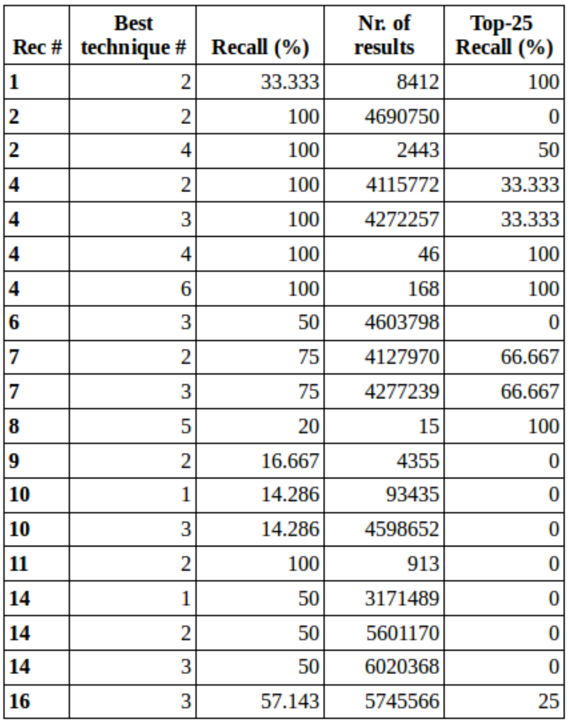
	Querying by Combining Sets					
	Technique 1		**Technique 2**		**Technique 3**	
Rec. #	*Recall (%)*	*Nr. of results*	*Recall (%)*	*Nr. of results*	*Recall (%)*	*Nr. of results*
1	33.333	510779	33.333	8412	33.333	4704487
2	100	1760041	100	4690582	100	4830304
3	0	138	0	4743412	0	4754077
4	33.333	1703535	100	4115772	100	4272257
5	0	1407797	0	731	0	4647211
6	0	1096165	0	1365	50	4603798
7	25	1383225	75	4127970	75	4277239
8	0	1000344	0	5049600	0	5071724
9	0	556083	16.667	4355	0	4681812
10	14.286	93435	0	886138	14.286	4598652
11	0	417369	100	906	0	4722028
12	0	2643975	0	10217	0	5983172
13	0	1169554	0	6720	0	5694553
14	50	3171489	50	5601170	50	6020368
15	0	38315	0	5643024	0	5646795
16	14.286	1429259	28.571	190	57.143	5745566
Average	16.890	1148843.937	31.473	2180660.25	29.985	5015877.688

Table 6: Percentage of goal articles found in the first (at most) 1000 results and the number of search results for each recommendation per technique. Technique 1-3 use ”querying by combining sets” for query construction with respectively the MeSH terms from the recommendation (technique 1), the primary and secondary evidence sets of MeSH terms (technique 2), and the MesH terms from the recommendation, and the primary/secondary evidence sets of MeSH terms (technique 3).



	Querying by Selecting Terms					
	Technique 4		**Technique 5**		**Technique 6**	
Rec. #	*Recall (%)*	*Nr. of results*	*Recall (%)*	*Nr. of results*	*Recall (%)*	*Nr. of results*
1	0	19	0	33954	0	19
2	100	2443	0	19	0	96
3	0	138	0	18	0	33
4	100	46	66.667	24	100	168
5	0	195	0	731	0	38
6	0	210	0	1366	0	28
7	0	210	25	65	25	613
8	0	182	20	15	0	25
9	0	4603	0	5571	0	626
10	0	266	0	4595685	0	266
11	0	255	0	4595685	0	237
12	0	21	0	25480	0	21
13	0	1077	0	6722	0	26114
14	0	190	0	35	0	204
15	0	19767	0	153	0	153
16	0	1939	28.571	230	28.571	193
Average	12.5	1972.562	8.765	579109.562	9.598	1802.125

Table 7: Percentage of goal articles found in the first (at most) 1000 results and the number of search results for each recommendation per technique. Technique 4-6 use ”querying by selecting terms” for query construction with respectively the MeSH terms from the recommendation (technique 4), the primary and secondary evidence sets of MeSH terms (technique 5), and the MesH terms from the recommendation, and the primary/secondary evidence sets of MeSH terms (technique 6).



Rec #	Best technique #	Recall (%)	Nr. of results	Top-25 Recall (%)
1	2	33.333	8412	100
2	2	100	4690750	0
2	4	100	2443	50
4	2	100	4115772	33.333
4	3	100	4272257	33.333
4	4	100	46	100
4	6	100	168	100
6	3	50	4603798	0
7	2	75	4127970	66.667
7	3	75	4277239	66.667
8	5	20	15	100
9	2	16.667	4355	0
10	1	14.286	93435	0
10	3	14.286	4598652	0
11	2	100	913	0
14	1	50	3171489	0
14	2	50	5601170	0
14	3	50	6020368	0
16	3	57.143	5745566	25

Table 8: The best technique, the recall (based on the at most first 1000 results), and the Top25-Recall for each recommendation.

and measure the percentage of articles that are ranked in the top 25 most relevant. We chose the number 25, as this is a reasonable amount that can be processed by a person in approximately an hour. In table 8 is for each recommendation given: the best technique (1-6), the recall for the at most 1000 results (the percentage goal articles in first 1000 results), and the top-25 recall. Recommendations for which we found no goal articles (3, 5, 12, 13 and 15) are omitted.

From these results, we can immediately see the urgency of keeping the number of search results low. In the cases where there are a lot of results, the goal articles have a very high chance to get lost outside of the top of the ranking. This reinforces the findings of (Iruetaguena, et al., 2013), who noted similar results. This indicates that the combination of the Rosenfeld-Shiffman filter combined with tf-idf is perhaps not a suitable way to process large numbers of articles, as the resulting ratings are very close to each other for many articles. For smaller sets of articles, for instance recommendations 4 and 8, the algorithm seems to have performed very well.

The addition of the MeSH distance to these ratings showed little difference. This can have multiple reasons:

- Not all articles are sufficiently annotated with MeSH terms. If an article is not annotated, the MeSH distance will always be 0, resulting in an advantage over other articles.
- The weighting of the MeSH distance was not optimally calibrated. The resulting number was very low and did not have much impact on the article's score.

While the second reason can be solved by further experimentation, the first reason indicates a difficult problem, that can only be solved by more consistent tagging of articles from the side of PubMed. Although the annotation standards have improved over the years, older articles are still poorly tagged which makes them harder to find, although those are probably less relevant for guideline update.

5 Conclusion

5.1 Findings

Giving automatically support to the guideline update process by identifying relevant papers for updating the guideline is a challenging task. Previous attempts have shown some success in finding articles for updates, but these approaches each had their limitations. Cohen et al. (2012) show some success in identifying goal articles with a machine learning approach, but this approach requires a large manually annotated dataset, which is very labor-intensive. Iruetaguena et al.(2013) were able to find these articles, but their result set was too large, and their rating and filtering proved insufficient to filter out goal articles to the top of their ranking.

Our approach focused on extracting as much information as possible from the recommendation text and the supporting evidence articles in the form of MeSH terms. These MeSH terms were then used to construct PubMed queries that could be tuned to be more specific or more broad depending on the number of results. Extracting MeSH terms from the recommendation text and from evidence articles was done by using E-utilities from PubMed. For the evidence we constructed a set of primary MeSH terms, which were shared amongst all articles, and set of secondary MeSH terms, which were shared amongst all but one.

From these sets of MeSH terms, two techniques for querying PubMed were constructed:

1. *Constructing queries by combining sets*, which takes two sets of MeSH terms as input and chooses which logical operator is used amongst

them (conjunction or disjunction) in order to make the query more broad or more specific.

2. *Constructing queries by selecting terms*, which takes the conjunction of a set of MeSH terms, and removes terms in order of importance to make the query more broad. The order of importance is determined by a pre-made ordering of MeSH subtrees.

The techniques were evaluated by taking older and newer versions of four medical guidelines. From these guidelines, recommendations concerning the same subject were selected that were updated between the older and the newer versions. From these recommendations, the set of articles introduced in the newer version of the recommendation were determined as 'goal articles'.

After the execution of a query, each article in the list of search results was rated based on scientific strength and relevance to the recommendation. To determine the strength, articles were judged on the Rosenfeld-Shiffman criteria. To determine relevance, a corpus of 50,006 article summaries was gathered. For each word in this corpus, the tf-idf weight was calculated. This weight was used in combination with a new measure called the inverse MeSH distance. This measure is based on the number of branches separating two terms in the same MeSH subtree. Based on these factors, a ranking is calculated.

We ran the program for each recommendation using different techniques. Overall, constructing queries by combining sets proved to be the most successful method for finding goal articles, particularly when used on the primary and secondary evidence terms. One problem that occurred when using this technique, was that the number of search results was highly volatile. Most of the higher level (more specific) queries yielded 0 results, while the lower level queries yielded hundreds of thousands of results.

The 'constructing queries by selecting terms' technique was successful in making more specific queries, and thus keeping the number of search results in check. This method of searching, however, did find a lot less of the goal articles.

Overall, the combined techniques found at least one goal article for 11 out of 16 recommendations. In total, 20 out of 71 goal articles were found.

The ranking of the articles was successful for queries with not too many results, for which the majority of articles was ranked in the top 25. For the larger lists of results ($>$ 1000 in length), goal articles were often lost in the middle of the ranking.

5.2 Discussion and Future Work

The first thing to state about the results, is that the amount of goal articles found is not necessarily representative of the success of the system. Guidelines are maintained by a panel that judge article relevance based on articles handed to them by an information expert. It could be that the program finds other useful articles that could be used to improve guideline quality other than the ones used by the committee. Based on our own judgment of the search results, this could very well be the case, as the top articles seem mostly relevant to the topic at hand. To make a solid judgment of this would take a medical professional or someone with expert knowledge on the subject. An experiment in a setting in which such an expert would provide feedback on the results could provide a better evaluation of the system.

Another factor that influenced the performance of the system, was the timespan between the two versions of the recommendation that the system was tested on. In our case this was several years; the time between the release of two guideline documents. When the concept of 'living guidelines' becomes a reality, the timespan might however be a lot shorter, for instance a month. Because this significantly reduces the search space, it could also greatly improve the results. Notice that our methods rely on the availability of the MeSH terms in PubMed.

The search strategies we developed each have their own upsides and downsides. Searching by combination works decently for finding goal articles, but the number of search results is not very scalable and tends to explode. This meant a lot of the selection of the articles was left to PubMed, which performed a decent job, but this was not the aim of the research.

Searching by terms on the other hand, scales very well when it comes to limiting the number of search results, but performs much worse when it comes to finding goal articles. This could be because the ordering of the importance between the MeSH subtrees was done based on intuition and not tested severely. This is because the number of possible orderings is explosive. An ordering made by a medical information expert could perhaps offer better results.

When it comes to the number of search results, we believe this is perhaps the most difficult problem to solve. The size of the MEDLINE database makes it difficult to set good restrictions on the number of results when constructing queries. Perhaps a hybrid method that combines aspects of both searching by combination and searching by terms could be used for this.

An extension that could be made to the method is the inclusion and prediction of the evidence rating. This is a letter that indicates how solid the evidence provided is. The rules for assigning this grade are clearly defined, and could be applied automatically. However, this step is outside of the scope of this research.

Looking at the performance of our ranking system, we see that there is room for improvement, as the algorithm seems to fall short when it comes to larger result sets. Perhaps this indicates that simple word comparison, even with tf-idf weighting, is not sufficient for this task. We therefore believe it will be useful if more meta-data would be included in the ranking. Examples of this would be MeSH terms, the number of times an article is cited, or the journal that an article has appeared in. We believe the MeSH distance is a good approach to attaining this in theory, but it ran into several practical problems, such as the lack of tags on a large number of articles. Using meta-data to judge the relevance of evidence would definitely be worth looking into in the future.

Overall, we believe that our system has achieved its goals, and is a good base for further research. We constructed a small set of test data that can be used in the future. During the evaluation, we clearly managed to identify the problems that our approach ran into, and we believe these offer solid ground for future research. We think living guidelines and automated guideline updates are definitely attainable in the future.

Acknowledgements

This work is supported by the European Commission under the 7th framework programme EURECA Project (FP7-ICT-2011-7, Grant 288048).

References

1. Entrez Programming Utilities Help [Internet]. Bethesda (MD): National Center for Biotechnology Information (US); 2010-. Available from: http://www.ncbi.nlm.nih.gov/books/NBK25501/

2. PubMed Help [Internet]. Bethesda (MD): National Center for Biotechnology Information (US); 2005-. PubMed Help. [Updated 2014 Jul 9]. Available from: http://www.ncbi.nlm.nih.gov/books/NBK3827/

3. Cohen, A. M., Ambert, K., & McDonagh, M. (2012). Studying the potential impact of automated document classification on scheduling a systematic review update. *BMC medical informatics and decision making*, 12(1), 33.

4. Field, M. J., & Lohr, K. N. (1990). Clinical practice guidelines: Directions for a new Program. *Washington (DC): Institute of Medicine*.

5. Iruetaguena, A., Garcia Adeva, J.J., Pikatza, J. M., Segundo, U., Buenestado, D., & Barrena, R. (2013). Automatic retrieval of current evidence to support update of bibliography in clinical guidelines. *Expert Syst. Appl.*, 40, 6 (May 2013), 2081-2091.

6. Lipscomb, C. E. (2000). Medical subject headings (MeSH). *Bulletin of the Medical Library Association*, 88(3), 265.

7. Peleg, M., Tu, S., Bury, J., Ciccarese, P., Fox, J., Greenes, R. A., ... & Stefanelli, M. (2003). Comparing computer-interpretable guideline models: a case-study approach. *Journal of the American Medical Informatics Association*, 10(1), 52-68.

8. Rogers, F. B. (1964). The Development of MEDLARS. *Bull Med Libr Assoc.*, 52(1): 150C151.

9. Rosenfeld, R. M., & Shiffman, R. N. (2009). Clinical practice guideline development manual: a quality-driven approach for translating evidence into action. *Otolaryngology–head and neck surgery: official journal of American Academy of Otolaryngology-Head and Neck Surgery*, 140(6 Suppl 1), S1.

10. Sackett, D. L., Rosenberg, W. M., Gray, J. A., Haynes, R. B., & Richardson, W. S. (1996). Evidence based medicine: what it is and what it isn't. *BMJ: British Medical Journal*, 312(7023), 71.

11. Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5), 513-523.

12. Seyfang, A., Martnez-Salvador, B., Serban, R., Wittenberg, J., Miksch, S., Marcos, M., ten Teije, A. & Rosenbrand, K. (2007). Maintaining formal models of living guidelines efficiently. In *Artificial Intelligence in Medicine* (pp. 441-445). Springer Berlin Heidelberg.

13. Shekelle, P., Eccles, M. P., Grimshaw, J. M., & Woolf, S. H. (2001). When should clinical guidelines be updated?. *BMJ: British Medical Journal*, 323(7305), 155.