Aristotle University's Approach to the Technologically Assisted Reviews in Empirical Medicine Task of the 2018 CLEF eHealth Lab

Adamantios Minas[0000−0001−6752−2338], Athanasios Lagopoulos[0000−0002−1979−3915], and Grigorios Tsoumakas[0000−0002−7879−669X]

Aristotle University of Thessaloniki, Thessaloniki 54124, Greece
{adamantcm,lathanag,greg}@csd.auth.gr

Abstract. Systematic reviews are literature reviewing processes that aim to retrieve all relevant content based on a specific topic, in an exhaustive manner. Such reviews are particularly useful in healthcare, where decision making must take into account all possible evidence, and are usually done by constructing a boolean query and submitting it to a database, and then screening the retrieved documents for relevant ones. Task 2 of CLEF 2018 eHealth lab focuses on automating this process on two fronts: Sub-Task 1 is about bypassing the construction of the boolean query, retrieving relevant documents and ranking them by relevance based on a protocol that describes a topic, and Sub-Task 2 is about ranking the documents retrieved by an already constructed query by Cochrane experts. We present our approaches for both sub-tasks, which combine a learning-to-rank model trained on multiple reviews with a model incrementally trained on each individual review using relevance feedback.

1 Introduction

Systematic reviews are a crucial part of Evidence-Based Medicine, which uses any current evidence to support a decision on how a patient will be treated. These reviews aim to find the aforementioned evidence, which should fit some criteria in order to take part in the final decision-making. Systematic reviews can be broken down into a 3-step process:

1. **Document Retrieval:** An expert builds a boolean query that describes their review topic, which is later submitted to a medical database. Boolean queries are queries that define if a document is relevant by the existence (or not) of user-specified terms in the document. By using boolean logic, complex queries with multiple rules can be constructed in order to filter through large amounts of information.
2. **Title and Abstract Screening:** After the possibly relevant documents have been retrieved, they must be screened to find the truly relevant ones. Screening takes part in two stages: in the first stage, experts review each retrieved document's title and abstract, and decide if it is non-relevant, or if it is possibly relevant and must be read in full to decide.

3. **Document Screening:** The second stage of screening is reading the full text of the document that passed through the first screening stage, and deciding if it should take part in the review.

Document screening is the most time-consuming task of this process. Medical databases are expanding rapidly - PubMed counts 26,759,399[1] citations as of 2017. Boolean queries on such databases are bound to retrieve a large amount of documents, hence the need for automation in such a task. This, however, is a complex problem, due to the imbalance of the data (few relevant documents, too many non-relevant documents) and the misclassification cost, where not including a relevant document might have a great toll on the final decision making.

Task 2 [1] of CLEF 2018 eHealth lab [2] focuses on the first two parts of the systematic review process. Our approach consists of phrase extraction and querying for the document retrieval step, as well as a hybrid classification model for the title and abstract screening step, which initially ranks the retrieved documents using Learning-to-Rank (LTR) features and then uses relevance feedback to iteratively re-rank them, based on simple text representations.

The rest of this paper is organized as follows: we briefly describe Task 2 of CLEF 2018 eHealth lab in Section 2, and in Section 3 we analyze our approaches. Section 4 contains the results and the submitted runs, and finally Section 5 concludes and discusses future work.

2 Task Overview

This year, CLEF eHealth's Task 2 was split into two sub-tasks. Sub-Task 1 was about searching in PubMed for relevant documents given a piece of text, while Sub-Task 2 was the same as last year's CLEF eHealth Task 2.

Sub-Task 1 aims to bypass the first part of a Systematic Review - the construction of the boolean query, that would later on be submitted in a database to retrieve possibly relevant documents.

Given 40 topics as training set and 30 as test set, participants were asked to return a ranking with a maximum of 5000 documents per topic. Each topic contained its id, title and objective, as well as a protocol that described that particular topic. Each topic protocol had 6 fields, with another objective field that was slightly different than the topic's one:

1. Objective
2. Type of Study
3. Participants
4. Index Tests
5. Target Conditions
6. Reference Standards

[1] `https://www.nlm.nih.gov/bsd/licensee/2017_stats/2017_LO.html`

For each topic, participants were also provided with a date cut-off. This cut-off was also used in the Boolean Queries that were constructed by Cochrane experts to retrieve relevant documents.

Sub-Task 2 concerns the efficient ranking of the possibly relevant documents retrieved. Given a topic, its query and the documents retrieved, the goal is to rank the documents so that most relevant ones appear first, as well as to find a threshold, after which no documents will be shown to the user. The training set consisted of 42 topics, where each topic contained:

1. A unique topic ID
2. A title
3. An Ovid MEDLINE boolean query, constructed by Cochrane experts
4. The PubMed IDs as returned from the execution of the boolean query

For both tasks, the relevant document PIDs (PubMed IDs) were provided as well, for abstract and content relevance. This enabled the use of algorithms that requested relevance feedback from the user.

3 Our Approach

For both sub-tasks, we used last year's model [3] with some enhancements, as well as some modifications for Sub-Task 1. It consists of two models:

1. An inter-topic XGBoost [4] classifier that is trained on LTR features between a topic and a document and produces an initial ranking of the documents. This inter-topic model is trained on all the training topics.
2. An intra-topic Support Vector Machine (SVM) classifier that is iteratively trained on TF-IDF vectors after asking feedback for documents that are ranked the highest by the inter-topic model. This intra-topic model is trained for each of the test topics using relevance feedback at prediction time.

Algorithm 1 describes the re-ranking algorithm employed by the intra-topic model.

3.1 Sub-Task 1: No Boolean Search

The first step for Sub-Task 1 was to find the initial relevant documents. For each topic, we used its title and objective to create queries that were later submitted to PubMed. To construct the queries, we tokenized both pieces of text, removed the stop-words, and extracted phrases from the resulting word lists. Figure 1 shows an example of this process.

The phrases we extracted were n-grams ($n \in \{2, 3, 4, 5, 6\}$) of the words of each piece of text. Each phrase was then submitted to PubMed, with the date cut-off as given for each topic, for which we retrieved a maximum of 2500 documents.

For the query construction, we also experimented with TextRank [5], an algorithm for keyword extraction. After extracting the keywords from both the

Algorithm 1: The Iterative relevance feedback algorithm of the intra-topic model

Input : The ranked documents R, of length n, as produced by the inter-review model, initial training step k, initial local training step $step_{init}$, secondary local training step $step_{secondary}$, step change threshold t_{step}, final threshold t_{final} (optional)

Output: Final ranking of documents R - $finalRanking$

$finalRanking \leftarrow ()$; `// empty list`

for $i = 1$ ***to*** k **do**

  $finalRanking_i \leftarrow R_i$

$k' \leftarrow k$;

while *not* $finalRanking$ *contains both* ***relevant*** *and* ***irrelevant*** *documents* **do**

  $k' \leftarrow k' + 1$;

  $finalRanking_{k'} = R_{k'}$;

while *not* $length(finalRanking) == n$ ***OR*** $length(finalRanking) == t_{final}$ **do**

  $train(finalRanking)$; `// Train a local classifier by asking for abstract or document relevance for these documents`

  $localRanking = rerank(R - finalRanking)$; `// Rerank the rest of the initial list` R `from the predictions of the local classifier`

  if $length(finalRanking) < t_{step}$ **then**

    $step = step_{init}$;

  else

    $step = step_{secondary}$;

  for $i = k'$ ***to*** $k' + step$ **do**

    $finalRanking_i \leftarrow localRanking_{i-k'}$;

return $finalRanking$;

title and the objective, we created the queries the same way as described above, where the text of each topic was now its keywords. This process did not seem to work well, as it decreased the total recall. We further experimented with the number of maximum allowed documents per query, where we had to trade between recall and number of documents retrieved. The 2500 limit proved to be a good fit, since retrieving more documents would not increase recall significantly, but would require our models to rank many more documents.

After retrieving the possibly relevant documents per topic, we use the inter-topic and intra-topic models to rank them. The LTR features used for the inter-topic model were computed using the title and abstract of each document and the different fields of each topic protocol, as well as the topic's title and objective. Table 1 shows the features employed by our model. For the inter-topic model, we use an Easy Ensemble [6] of 10 XGBoost classifiers, where each classifier is trained on all the relevant documents and a subset of the non-relevant documents, randomly sampled, sampling 5 non-relevant documents per each relevant.

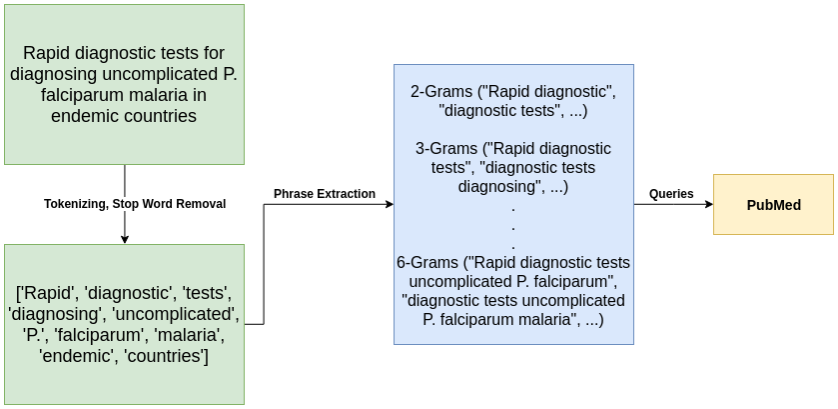

Fig. 1. An example of the query construction pipeline of sub-task 1.

After getting an initial ranking from the inter-topic model, we use the intra-topic model to re-rank up to the first 20,000 documents, and keep the first 5000, as per the task's limit.

3.2 Sub-Task 2: Abstract and Title Screening

For the second sub-task, we also employed last year's model with a few modifications on both the inter-topic and the intra-topic model.

Inter-Topic Model On the Inter-Topic model, we included some semantic information using additional LTR features. Table 2 shows the features, with which we previously experimented, along with the new semantic features. We further advanced our model by removing the stop-words and fixed some minor issues with the BM25 [7] features.

Features 1-24 are the same as last year's submission. We distinguish between two topic fields - the query, which is a list of Medical Subject Headings (MeSH) terms extracted from the topic's Ovid Medline query and the title. MeSH terms are semantic annotations added manually on PubMed documents. The notation used for the LTR features is as follows:

1. t is a topic field
2. d is a document field
3. $c(t_i, d)$ counts the number of times the term t_i appears on the document field d
4. $c(m_i, d)$ counts the number of times the MeSH (Medical Subject Headings) term m_i appears on the document field d
5. $|C|$ is the total number of documents in the collection
6. $df(t_i)$ is the number of documents that contain the term t_i

Table 1. Set of features employed by the inter-topic model for Sub-Task 1.

Features	Description	Topic field	Document field(s)
1, 2	*cos*(tf-idf)	Title	Title, Abstract
3, 4	*cos*(tf-idf)	Objective	Title, Abstract
5, 6	*cos*(tf-idf)	Protocol - Objective	Title, Abstract
7, 8	*cos*(tf-idf)	Protocol - Type of Study	Title, Abstract
9, 10	*cos*(tf-idf)	Protocol - Participants	Title, Abstract
11, 12	*cos*(tf-idf)	Protocol - Index Tests	Title, Abstract
13, 14	*cos*(tf-idf)	Protocol - Target Conditions	Title, Abstract
15, 16	*cos*(tf-idf)	Protocol - Reference Standards	Title, Abstract
17, 18	BM25	Title	Title, Abstract
19, 20	BM25	Objective	Title, Abstract
21, 22	BM25	Protocol - Objective	Title, Abstract
23, 24	BM25	Protocol - Type of Study	Title, Abstract
25, 26	BM25	Protocol - Participants	Title, Abstract
27, 28	BM25	Protocol - Index Tests	Title, Abstract
29, 30	BM25	Protocol - Target Conditions	Title, Abstract
31, 32	BM25	Protocol - Reference Standards	Title, Abstract
33, 34	log(BM25)	Title	Title, Abstract
35, 36	log(BM25)	Objective	Title, Abstract
37, 38	log(BM25)	Protocol - Objective	Title, Abstract
39, 40	log(BM25)	Protocol - Type of Study	Title, Abstract
41, 42	log(BM25)	Protocol - Participants	Title, Abstract
43, 44	log(BM25)	Protocol - Index Tests	Title, Abstract
45, 46	log(BM25)	Protocol - Target Conditions	Title, Abstract
47, 48	log(BM25)	Protocol - Reference Standards	Title, Abstract

7. $levenshtein(m_i, d_j)$ is the levenshtein distance betwen the MeSH term m_i and the term d_j

For features 25 and 26, we applied Latent Semantic Analysis to the TF-IDF vectors of the titles and the abstracts of each document, keeping 200 components. Then for each document in a topic, we computed the cosine similarity of their LSA vectors (topic title - document title and topic title - document abstract).

Features 27 and 28 use Word2Vec [8] vectors, obtained from the BioASQ challenge[2]. These vectors were trained on 10,876,004 abstracts from PubMed, with a vocabulary of 1,701,632 words and a dimensionality of 200. For each piece of text, we sum up all its word vectors and average, which results in a single vector representing the document. Then, we compute the cosine similarities between a topic and a document using these vectors.

Features 29 and 30 use the Word2Vec vectors again, this time to compute the Word Mover's Distance [9] between pieces of text.

[2] `http://bioasq.org/`

Table 2. Set of features employed by the inter-topic model for Sub-Task 2.

ID	Description	Category	Topic field	Document field
1	$\sum_{t_i \in t \cap d} c(t_i, d)$	$T-D$	Title	Title
2	$\sum_{t_i \in t \cap d} \log(c(t_i, d))$	$T-D$	Title	Title
3	$\sum_{t_i \in t \cap d} c(t_i, d)$	$T-D$	Title	Abstract
4	$\sum_{t_i \in t \cap d} \log(c(t_i, d))$	$T-D$	Title	Abstract
5	$\sum_{m_i \in t \cap d} c(m_i, d)$	$T-D$	Query	Title
6	$\sum_{m_i \in t} \sum_{d_j \in d} levenshtein(m_i, d_j)$	$T-D$	Query	Title
7	$\sum_{m_i \in t} \sum_{d_j \in d} levenshtein(m_i, d_j)$ if $levenshtein(m_i, d_j) < v$	$T-D$	Query	Title
8	$\sum_{m_i \in t \cap d} \log(c(m_i, d))$	$T-D$	Query	Title
9	$\sum_{m_i \in t \cap d} c(m_i, d)$	$T-D$	Query	Abstract
10	$\sum_{m_i \in t \cap d} \log(c(m_i, d))$	$T-D$	Query	Abstract
11	$\sum_{t_i \in t} \log(\frac{\lvert C \rvert}{df(t_i)})$	T	Title	-
12	$\sum_{t_i \in t} \log(\log(\frac{\lvert C \rvert}{df(t_i)}))$	T	Title	-
13	BM25	$T-D$	Title	Title
14	BM25	$T-D$	Title	Abstract
15	BM25	$T-D$	Query	Title
16	BM25	$T-D$	Query	Abstract
17	log(BM25)	$T-D$	Title	Title
18	log(BM25)	$T-D$	Title	Abstract
29	log(BM25)	$T-D$	Query	Title
20	log(BM25)	$T-D$	Query	Abstract
21	*cos*(tf-idf)	$T-D$	Title	Title
22	*cos*(tf-idf)	$T-D$	Title	Abstract
23	*cos*(tf-idf)	$T-D$	Query	Title
24	*cos*(tf-idf)	$T-D$	Query	Abstract
	Semantic Features			
25	*cos*(LSA(tf-idf))	$T-D$	Title	Title
26	*cos*(LSA(tf-idf))	$T-D$	Title	Abstract
27	*cos*(Word2Vec)	$T-D$	Title	Title
28	*cos*(Word2Vec)	$T-D$	Title	Abstract
29	WMD(Word2Vec)	$T-D$	Title	Title
30	WMD(Word2Vec)	$T-D$	Title	Abstract
31	*cos*(Doc2Vec)	$T-D$	Title	Abstract

Feature 31 uses document vector representations which we obtained by training a Doc2Vec [10] model on the documents collected from the training set. The model was trained on each document's title and abstract. The vectors for documents not in the model's training set were inferred.

The new semantic features seemed to improve performance, but some of them proved to be better than others. For the final runs, from the semantic features

we kept only 25, 26, 29 and 30, which use the Latent Semantic Analysis and the Word Mover's Distance.

Apart from adding new LTR features, we experimented with a variety of other techniques. First, we tried expanding the title query with more words, to obtain a bigger piece of text, so as to compute more accurate similarities. For each word in the title, we found its K most similar words using cosine similarity on the Word2Vec embeddings and added them to the title. Even for small values of K (e.g. 2) this did not seem to improve performance. We further tested to provide the document vectors (query title, document) from Doc2Vec directly to the inter-topic model, either concatenated or subtracted one from another, which still did not improve performance. Lastly, we experimented with undersampling techniques - specifically Easy Ensemble and SMOTE [11], which did not improve performance either. On the contrary, Easy Ensemble works well for the first sub-task, where the number of non-relevant documents is on average an order of magnitude larger.

Intra-Topic Model For the intra-Topic model, we relaxed the C parameter of the SVM, which controls how "strict" the hyperplane will be in avoiding misclassification to allow for a bigger margin. The intuition for this came from the fact that due to the sheer class imbalance, finding a hyperplane with a bigger margin will probably fit the data better than finding a strict one which may lead to overfitting. This relaxation seemed to improve the model's predictions in our evaluations.

Additionally, we experimented with different SVM kernels, but they proved much slower and less efficient than the linear one. We also added n-grams (2, 3) but they did not give better results either. Finally, we tried to use embeddings for this task as well, by using the average Word2Vec vectors or the document vectors from Doc2Vec as input instead of the simple TF-IDF representations, to no avail.

4 Results

Both sub-tasks of CLEF E-health Task 2 supported both thresholded and non-thresholded runs. Our models, however, do not apply a threshold to the final ranking automatically - instead, we submitted thresholded runs on fixed hand-picked thresholds.

The metrics used for evaluation were multiple and they are described in detail in the task's website[3]. The primary ones (as mentioned in the task's website) are the Mean Average Precision and the Recall, on which we focus below. Note that in the official evaluation script[4], which we used to produce the following results,

[3] `https://sites.google.com/view/clef-ehealth-2018/task-2-technologically-assisted-reviews-in-empirical-medicine`

[4] https://github.com/CLEF-TAR/tar

Mean Average Precision is computed on the whole ranking, without taking the threshold into account.

Table 3 shows our results for sub-task 1. The reranking parameters for the intra-topic model of HybridSVM are:

$$k = 10, step_{initial} = 1, t_{step} = 200, step_{secondary} = 50, t_{final} = 1000$$

The Threshold column refers to the hand-picked threshold mentioned above, and the Train Relevance column refers to which relevances were used for training - abstract or content. For evaluation, content relevance was used as per the competition's guideline. We submitted runs 1, 2 and 3, since we only found that training with abstract relevance gave slightly better results after the submission deadline. This is, however, an interesting observation - since there are more relevant documents at abstract level than at content level, the class imbalance was slightly less effective when training with the abstract relevance, thus producing slightly better results.

Table 3. Sub-Task 1 Results

ID	Average Precision	Recall	Threshold	Train Relevance
1	0.113	0.816	5000	Content
2	0.113	0.809	2500	Content
3	0.113	0.787	1000	Content
4	0.117	0.819	5000	Abstract
5	0.117	0.812	2500	Abstract
6	0.117	0.797	1000	Abstract

Table 4 shows our results for the second sub-task. The only threshold we used is at 1000 documents. The last 4 columns refer to the parameters of the intra-topic's re-ranking algorithm. From the runs shown, we submitted runs 1, 5 and 7.

Table 4. Sub-Task 2 Results

ID	Average Precision	Recall	Threshold	k	$step_{initial}$	t_{step}	$step_{secondary}$	t_{final}
1	0.4	1.0	-	5	1	200	100	2000
2	0.396	1.0	-	10	1	200	50	2000
3	0.393	1.0	-	10	1	200	100	1000
4	0.396	1.0	-	10	1	300	100	2000
5	0.4	0.944	1000	5	1	200	100	2000
6	0.396	0.946	1000	10	1	200	50	2000
7	0.393	0.943	1000	10	1	200	100	1000
8	0.396	0.945	1000	10	1	300	100	2000

5 Conclusion and future work

In this paper, we described our approaches for both sub-tasks of Task 2 of CLEF eHealth 2018. We introduced new features and tweaked last year's models to improve performance, with a tendency towards semantic features.

As future work, we believe that more improvements can be made in both sub-tasks. For Sub-Task 1, the query construction stage could benefit from filtering out words that are not medically relevant, in order to reduce the number of queries and consequently reduce the number of retrieved documents. For the ranking model (sub-tasks 1 and 2), more semantic features could benefit the inter-topic model, while a better strategy for asking feedback in the intra-topic model could boost the metrics. Finally, it would be interesting to apply deep learning techniques to the task, and try to use word embeddings in a more efficient way.

References

1. Evangelos Kanoulas, Rene Spijker, Dan Li, and Leif Azzopardi. CLEF 2018 Technology Assisted Reviews in Empirical Medicine Overview. In *CLEF 2018 Evaluation Labs and Workshop: Online Working Notes, CEUR-WS*, 2018.
2. Hanna Suominen, Liadh Kelly, Lorraine Goeuriot, Evangelos Kanoulas, Leif Azzopardi, Rene Spijker, Dan Li, Aurlie Névéol, Lionel Ramadier, Aude Robert, Guido Zuccon, and Joao Palotti. Overview of the CLEF eHealth Evaluation Lab 2018. CLEF 2018. In *8th Conference and Labs of the Evaluation Forum, Lecture Notes in Computer Science (LNCS), Springer*, 2018.
3. Antonios Anagnostou, Athanasios Lagopoulos, Grigorios Tsoumakas, and Ioannis Vlahavas. Combining inter-review learning-to-rank and intra-review incremental training for title and abstract screening in systematic reviews. In *CLEF 2017 Working Notes, CEUR Workshop Proceedings*, volume 1866, Dublin, Ireland, 2017.
4. Tianqi Chen and Carlos Guestrin. XGBoost. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, pages 785–794, New York, New York, USA, 2016. ACM Press.
5. Rada Mihalcea and Paul Tarau. TextRank: Bringing order into texts. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, EMNLP 2004*, pages 404–411, Barcelona, Spain, 2004.
6. Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou. Exploratory Undersampling for Class Imbalance Learning. *IEEE Transactions on Systems, Man and Cybernetics*, 39(2):539–550, 2009.
7. K. Sparck Jones, Karen Sparck Jones, S Walker, S Walker, S E Robertson, and Stephen E Robertson. A probabilistic model of information retrieval: development and comparative experiments Part 2. *Information Processing and Management*, 36:809–840, 2000.
8. Tomas Mikolov, Greg Corrado, Kai Chen, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. *Proceedings of the International Conference on Learning Representations (ICLR 2013)*, pages 1–12, 2013.
9. Matt J Kusner, Yu Sun, Nicholas I Kolkin, and Kilian Q Weinberger. From Word Embeddings To Document Distances. *Proceedings of The 32nd International Conference on Machine Learning*, 37:957–966, 2015.

10. Qv Le and Tomas Mikolov. Distributed Representations of Sentences and Documents. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014*, volume 32, pages 1188–1196, Beijing, China, 2014.
11. Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 2002.