Journal of Data Intelligence, Vol. 3, No. 4 (2022) 474–504
© Rinton Press

IMPROVED METHODS TO AID UNSUPERVISED EVIDENCE-BASED FACT CHECKING FOR ONLINE HEALTH NEWS

PRITAM DEKA
Queen's University Belfast
pdeka01@qub.ac.uk

ANNA JUREK-LOUGHREY
Queen's University Belfast
a.jurek@qub.ac.uk

DEEPAK P.
Queen's University Belfast
deepaksp@acm.org

False information in the domain of online health related articles is of great concern, which has been witnessed abundantly in the current pandemic situation of Covid-19. Recent advancements in the field of Machine Learning and Natural Language Processing can be leveraged to aid people in distinguishing false information from the truth in the domain of online health articles. Whilst there has been substantial progress in this space over the years, research in this area has mainly focused on the sphere of political news. Health fake news is markedly different from fake news in the political context as health information should be evaluated against the most recent and reliable medical resources such as scholarly repositories. However, one of the challenges with such an approach is the retrieval of the pertinent resources. In this work, we formulate two techniques for the retrieval of the most relevant authoritative and reliable medical content from scholarly repositories which can be used to assess veracity of an online health article. The first technique is an unsupervised method of generating queries from claims which are extracted from an online health article. We propose a three-step approach for it and illustrate that our method is able to generate effective queries which can be used for retrieval of information from medical knowledge databases. The second method involves a filtering approach for extracting the most relevant information for the claims. We show how this can be achieved with the help of state of the art transformer models and illustrate it's effectiveness over other methods.

Keywords: Query Generation, Health Misinformation, Keyword Extraction, Information Retrieval, Transformers

1. Introduction

One of the growing problems in today's world is that of false information. The availability of such information in the online world is staggeringly high that it is seen as a rising threat to democracy and the economy of countries [1, 2]. Due to the abundance of fake information found online, people are even starting to lose faith in governments [1]. One of the main causes of the heightened impact of false information is the fact that online false information spreads much faster [3] through exploiting cognitive biases. This is further aided by the rise of social media [4], which enables people to share unverified facts and opinions in an unregulated manner.

The majority of work on automated online content verification has been focused on detection of political fake news and fake reviews [5, 6, 7, 8, 9, 10, 11, 12], while detection of health misinformation using computational methods is a relatively new area of research. Health misinformation can be defined as "a health-related claim of fact that is currently false due to a lack of scientific evidence"[13]. Since more people are connected via the internet, the exchange of information among people has increased and monitoring the trustworthiness of the exchanged information becomes difficult. Checking health disinformation has become a crucial task as people believe any facts they find online as the truth which may lead to serious consequences. For example, regarding the Covid-19 disease it was reported by The Standard, UK[a] that around 700 people died in Iran due to a fake information which claimed that ingestion of methanol helps in curing the disease. Another incident which raised concerns across the world was when the USA president Donald Trump suggested injecting disinfectant to treat Covid-19 as reported by EuroNews on 24 April, 2020[b]. Such information being spread creates a lot of confusion in the society. Health-related information often caters to deep rooted social as well as personal stigmas, making it sometimes invisible within the public discourse. Much of medical fake news relates to traditional beliefs that have been prevalent for many years regarding health while lacking any supporting scientific evidence. These may be effectively uncovered through contrasting health information against reliable medical sources. For a layperson, verifying health misinformation is a difficult task as it has to be done by someone with knowledge about health and medicine; this aspect makes it very distinct from verification of political fake news. We believe that a very effective strategy to counter health information would be to first identify the key claim within an article. Following this, the most relevant and up to date reliable sources of information relevant to the claim may be searched for supporting or refuting evidence.

Our Contribution. In this work, we first formulate a novel technique towards addressing the task of generating queries from from health-related articles leveraging keywords/phrases in an unsupervised way. The queries are designed for usage to retrieve relevant authoritative medical resources to assess the articles veracity. We propose a three-step method for addressing the unsupervised query generation task, which is evaluated with a new dataset curated for the purpose of this study. We then address the task of retrieving the most relevant information for the claims from a set of documents retrieved using the generated queries. We propose a retrieval approach for this task which uses a transformer model to retrieve the relevant information for the claims. We also show that domain specific transformer models are more effective than generic models through an empirical evaluation using a publicly available dataset for this specific task. A pictorial representation of the method is shown in Fig 1. All the models that we have fine-tuned, along with two working applications based on our method, are publicly available via the repository `https://huggingface.co/pritamdeka`.

2. Related Work

In this section we briefly summarize literature related to our tasks that are located within the domain of health misinformation.

[a] `standard.co.uk`

[b] `https://www.euronews.com/2020/04/24/donald-trump-suggests-injecting-disinfectant-to-treat-covid-19`

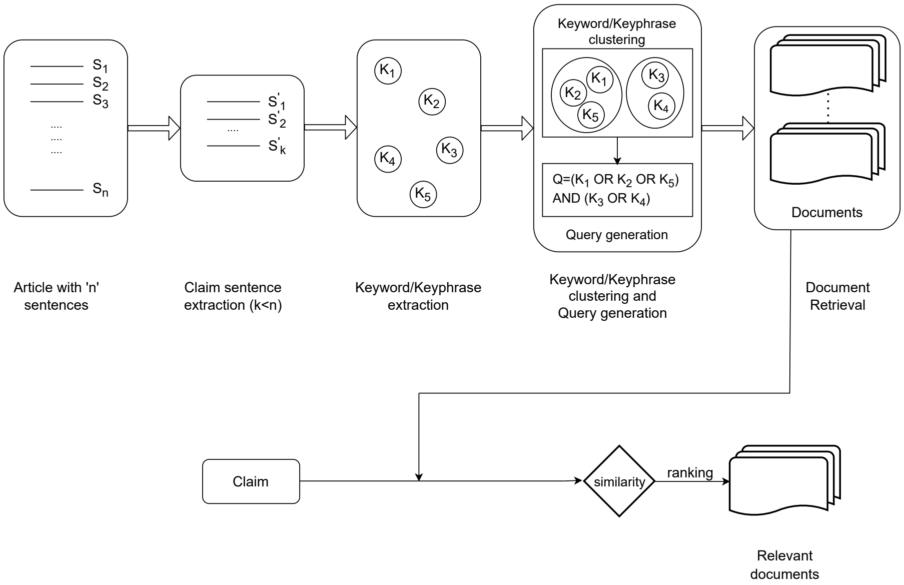

S_1
S_2
S_3
....
S_n
S'_1
S'_2
....
S'_k
K_1
K_2
K_3
K_4
K_5
Keyword/Keyphrase clustering
Q=(K_1 OR K_2 OR K_5) AND (K_3 OR K_4)
Query generation
Documents
Article with 'n' sentences
Claim sentence extraction (k<n)
Keyword/Keyphrase extraction
Keyword/Keyphrase clustering and Query generation
Document Retrieval
Claim
similarity
ranking
Relevant documents

Fig. 1. Representation of the proposed method

2.1. *Health Misinformation*

Although the problem of misinformation in health-related content has been known for a while, using computational methodologies in this field has only started to gain attention in the past few years.

Recent works such as [14] and [15] explored health news article contents in order to classify them as reliable and unreliable. In both the works, various linguistic features and statistical features such as word count are extracted from reliable as well as unreliable news articles. These features are then used to build machine learning classifiers which can then classify articles as reliable or unreliable. The difference between the techniques is that [14] used a text classifying method, fastText [16] as another classifier for the same task. They experimented with both the classifiers and found that both classifiers performed effectively.

The work proposed in [17], [18] and [19] deals with the detection of misinformation found in online health forums. In the first two works, the authors use linguistic features extracted from the comments and titles in these forums along with other features extracted from the users' profiles. [17] proposes an approach for assessing the credibility of users and medical statements posted by them in online health forums. The authors model the relation between credible statements and trustworthiness of users using a probabilistic graphical model. To train a machine learning model for classifying the statements as credible or not, they used a labelled subset of statements whose ground-truth labels were derived from an expert medical database. Similarly, [18] proposes a method for the detection of misinformative comments in online health forums by applying a Random Forest classifier model with linguistic features and user network characteristics as the input features. Both the aforementioned works use supervised methods, however [19] presents an unsupervised method which identifies trustworthy answers by non-expert users for health-related questions in online health forums. The authors use word2vec [20] to compute the embedding vectors for a question and it's answers given by various users. Along with a reliability score of the users, the distance between the question and answer vectors is calculated. The answer having the minimum distance to the question is considered the most trustworthy answer.

Research has also been carried out to deal with health-related misinformation found in Twitter. The work by [21] proposes an approach to classify a health-related tweet as a rumor or not by using various features extracted from tweets such as word count or sentiment along with models such as Naïve Bayes, Random Forest and Random Decision Tree. A similar work [22] proposes an approach to detect untrustworthy users in Twitter who are likely to spread health-related rumors. The authors use sentiment and linguistic features extracted from tweets and other features obtained from users' profiles such as 'follows' relationship information. These features are then used with a logistic regression model to build a classifier for trustworthy and untrustworthy users.

It can be noted that majority of the proposed work consider the medical content verification as a classification task where various type of features extracted from the content or relevant metadata are used as input to machine learning models. The common limitation of those approaches is the fact that they assess the veracity of medical articles independently from the existing authoritative and reliable medical knowledge, except as conveyed through the labellings on the articles. Different from the aforementioned works, in [23] the authors propose a fake medical news detection algorithm, which is based on the principles of evidence-based medicine. For a textual document containing a medical claim, the algorithm retrieves a relevant content from the Turning Research Into Practice (TRIP) database[c], which is then used to verify the claim based on a predefined agreement score. The retrieval process uses collection of medical phrases extracted from the document with the application of the TextRank [24] algorithm and a supervised binary classification model. Such an evidence-based technique offers a more principled approach to medical fact checking in comparison to machine learning based solutions. As discussed by reputed health information providers such as HealthLine[d] , medical facts should be evaluated in consideration with trustworthy news sources and by carefully weighing balance of evidence.

2.2. *Keyphrase extraction*

The goal of keyphrase extraction is to extract certain phrases from a document which are related to the main idea or topic of the document [25]. Over the years, keyphrase extraction has become an important task in IR and NLP applications such as document summarization, opinion mining, and text categorization. According to a popular survey by [25], keyphrase extraction approaches can be categorized into supervised and unsupervised works.

2.2.1. *Supervised approaches*

The works that use a supervised approach consider the task as a binary classification problem or a ranking problem. As a binary classification problem, [26] used a Nave Bayes algorithm to predict keyphrases from candidate phrases by extracting features from the candidate phrases

[c] `https://www.tripdatabase.com/`

[d] `https://www.healthline.com/health-news/learn-to-spot-fake-health-news-with-these-5-tips`

such as tf-idf and distance in the document of the phrases first appearance. A similar work by [27] is based on using a decision tree algorithm to classify phrases as keyphrases based on various features such as number of words in the phrase, first occurrence of the phrase in the document, and the frequency of stemmed phrase. Another work on classifying keyphrases [28] used thesaurus knowledge along with tf-idf as features and a boosting algorithm, Adaboost [29] as the classifier algorithm. The training data was annotated as positive or negative for keyphrases and non-keyphrases respectively.

One of the early notable research works of the ranking approach by [30] used a pre-determined domain-specific glossary database in order to score noun phrases based on the frequency of its occurrence in the document, its composition, and specificity in the domain of the document. The noun phrases with the highest scores were then selected as the keyphrases. Recent works have also focused on using neural networks for extracting ranked keyphrases. [31] used a multi-layer perceptron (MLP) for extracting keyphrases from scientific articles. Noun phrases were first extracted from a document and then scored based on the phrase frequency, phrase links to other phrases and tf-idf. The model was then trained using a MLP and the k-top ranked candidate phrases were then selected as the keyphrases. [32] used hashtags from tweets to first identify candidate phrases and then used word2vec embeddings [20] as input to a deep neural network. Every tweet was assigned a score of 2 (perfectly suitable), 1 (suitable), or 0 (unsuitable) to indicate whether the hashtag of the tweet was a good keyphrase for it.

2.2.2. *Unsupervised approaches*

In the domain of unsupervised approaches, the problem of keyphrase extraction is considered as a ranking problem. Early work includes graph-based approaches such as [24, 33] who used co-occurrence counts between candidate phrases to compute their relatedness. The importance of a candidate phrase depends on how related it is to other candidate phrases. The more related it is, the more important that phrase is and more likely to be a keyphrase. [34] used a semantic relatedness approach instead of co-occurrence counts to find the relatedness of candidate phrases. [35] is another graph-based approach where information from all positions of a words occurrences is incorporated to first score each candidate phrase. After that the PageRank [36] algorithm is used to rank the candidate phrases according to the score.

Another approach in the unsupervised domain is clustering of the candidate keyphrases into various topical clusters such that candidate phrases in one cluster should be related to only to a particular topic. Notable work includes [37] that clusters semantically similar candidates using Wikipedia and co-occurrence-based statistics. The underlying hypothesis is that each of these clusters corresponds to a topic covered in the document, and selecting the candidates close to the centroid of each cluster as keyphrases ensures that the resulting set of keyphrases covers all the topics of the document. [38] proposed TopicalPageRank, which runs TextRank [24] multiple times for a document, once for each of its topics induced by a Latent Dirichlet Allocation [39] so that the main topics of the document are covered by the extracted keyphrases. The final score of a candidate is computed as the sum of its scores for each of the topics, weighted by the probability of that topic in that document. [40] clustered the candidate keyphrases into topics using a Hierarchical Agglomerative Clustering (HAC) algorithm. After that they apply the TextRank [24] algorithm to assign a significance score to each topic. Keyphrases are then generated by selecting a candidate from each of the top ranked topics.

Recent work also includes the usage of word embeddings for the purpose of unsupervised keyphrase extraction. [41] used Sent2Vec [97] and Doc2Vec [43] to produce sentence embeddings. The candidate phrases and the document is first embedded using the pre-trained embedding models of both Sent2Vec and Doc2Vec and then the embedding vectors of candidate phrases are matched for similarity with the embedding vector of the document by taking cosine similarity between them. The keyphrases are then ranked according to the cosine similarity score and the top ranked are selected.

2.3. *Query generation*

As our work is based on query generation from keywords for effective retrieval of documents from medical knowledge database, the most relevant work is Boolean query generation for systematic reviews of biomedical literature. Systematic reviews are used to provide medical practitioners with advice to assist them with their case studies. These reviews are basically distilled from the results of an exhaustive search within a corpus of published research literature such as MEDLINE. To perform these reviews, complex Boolean queries are first formed to retrieve documents from the corpus which are then triaged by experts to find the most relevant literature. However, this whole process is done manually which consumes time. Recently, there have been efforts to automate the manual process of query generation using computational adaptations of the manual approaches which take into account high-level concepts and their synonyms in order to generate queries. In [44], the authors propose a framework to automatically create queries from a given short statement about the topic of the review. They first form a high-level hierarchy of the query terms from the statement. This is done by forming a parse tree and then using POS tags to form the hierarchy. The high-level concepts are then represented by entities from a reference entity repository such as UMLS. This is then followed by an optional step of expanding queries by including entities related to the query terms. After that entities are mapped to keywords to form the final query.

2.4. *Evidence based fact-checking*

The FEVER[45] dataset has been the de facto standard for evidence based fact-checking when construed as a cross-domain task. It relies on Wikipedia and fact-checking websites to provide the evidences. However, approaches for evidence based fact-checking for scientific or health related claims needs domain specific knowledge and only a few datasets have been released very recently catering to such claims. These include the SCIFACT[46], COVID-fact[47] and the HealthVer[48] datasets. Both the COVID-fact and HealthVer datasets deal with Covid-19 related claims and they provide evidences for the claims. The SCIFACT dataset, however, provides a corpus of PubMed abstracts which contains the evidences for a wider range of health related claims. There has been various works based on a shared task using the SCIFACT dataset. These include the work by [49] which uses a longformer[50] model to predict the veracity labels of the claims as well as identifying the evidence statements from within the abstracts. [51, 52, 53] use a three-stage approach for their work consisting of (1) document retrieval, (2) evidence selection, and (3) label prediction using the selected evidences. The difference in their work lies on the method of selection of the evidence statements. [51] uses each sentence independently to select the evidence statements. [52] and [53] encodes the abstracts and use pooling and self-attention to produce sentence representations used for evidence selection.

Our work is based on extracting claims from online health articles and then creating keyword-based queries for further retrieval of information from medical knowledge databases as a means of fact-checking. We then follow a similar task of retrieving the most relevant documents as evidences for the extracted claims.

3. Methodology

In the following sub sections we first define the task formulation of query generation as well as the task formulation for document retrieval and then explain the proposed methods.

3.1. *Task formulation*

Consider a text article D which is known to contain some medical or health related claim/s. Let the article D comprise of n sentences; $D = [s_1, \ldots, s_n]$. Our first task targets unsupervised identification of the keywords/keyphrases from claims within article D which are then used for the creation of queries that can be further used to retrieve relevant medical content containing evidence related to the claim.

$$[D = [s_1, \ldots, s_n]] \xrightarrow[\text{extraction}]{\text{claim}} \{c_1, \ldots, c_k\} \xrightarrow[\text{extraction}]{\text{keyword/keyphrase}} \{f_1, \ldots, f_k\} \xrightarrow[\text{generation}]{\text{query}} \{Q\}$$

The output set contains one query which is formed by clustering the keywords/keyphrases according to their similarity.

The next task focuses on retrieving the relevant documents from the set of retrieved documents by the query. Given a claim c and a set of documents $A = [a_1, \ldots a_n]$, the goal is to find the most relevant documents for the claim.

$$[c, A = [a_1, \ldots, a_n]] \xrightarrow[\text{retrieval}]{\text{relevant document}} [a_1, \ldots a_k], \text{ where } k < n$$

Both the tasks are designed to be the intermediate steps of the evidence based disinformation identification task that determines whether health related claims are true or fake using the existing authoritative medical resources.

3.2. *Proposed Method for query generation*

The proposed solution follows a three-step approach for the generation of the queries. We explain each of those steps in detail in the following subsections..

3.2.1. *Identification of the claim sentences:*

In the first step we identify the sentence from the article which makes a health-related claim that can be fact-checked. The pseudo-code for this step is presented in the Algorithm 1. Here, we use TextRank [24] which is a graph-based ranking algorithm. Each sentence from the document is considered as a single vertex. Two vertices are connected if there is a semantic similarity between the two sentences. Each vertex (sentence) is then scored based on its "importance within the document. This score allows us to detect those sentences from the articles which convey the central meaning of the article. In our case the similarity between two sentences is computed as cosine similarity computed on sentences embeddings calculated with a pre-trained S-BERT [54] language model (lines 11-19). Weight assigned to each edge represents the similarity between sentence pairs. Following the construction of the graph (line 20), the TextRank [24] score, $s_{i_{score}}$ is calculated for each sentence, $[s_1, s_2, s_3, \dots s_n]$ (line 22) which lies between the range [0,1]. In the Toulmin theory of argumentation[55], an argument's conclusion is the claim. Based on this, we propose to find the concluding sentences in an online health article. A conclusion indicator is a word or phrase that indicates that the statement its attached to is a conclusion. Although an exhaustive list of conclusion indicator words doesn't exist, we use the indicator words provided by [56] for the conclusion indicators (e.g., concluded, concludes, in sum). Taking this under consideration, for each sentence we assign a score, $s_{i_{indicator}}$ based on the presence of indicator words in the sentences with 1 if indicator words are present and 0 otherwise (lines 2-10). The TextRank [24] and the indicator scores for each sentence are added and the sentences are ranked according to the decreasing final scores (lines 21-24).

$$s_{i_{final\_score}} = s_{i_{indicator}} + s_{i_{score}}$$

As the output we obtain ranked sentences from the health articles in a descending order based on the final score calculated for each sentence. This allows us to identify the key sentences that defines the subject matter of the article. In our experimental evaluation that will be described in a later section, we observed that the key claims made by the document are always among the sentences obtained on the top of the list.

3.2.2. *Extraction of the keywords/keyphrases from the claim sentences:*

In the second step we extract keywords/keyphrases from the claim sentences in order to create the queries. The pseudo-code for this step is presented in Algorithm 2. For this, the top k ranked sentences are used to extract the keywords/keyphrases. The value for k is chosen to be either 3 or 10% of the length of the article, whichever is greater. Based on the experimental results of the first step, we find that many articles miss out claim related sentences if k is too small but at the same time we find that a large value of k extracts non-relevant sentences. Following the definition by [25], we first identify named entities from the top k sentences which will be used as the candidate keys (lines 1-3). From these candidate keys, the most significant ones will be used as keywords/keyphrases.

Since we are dealing with health articles, we want to extract the specific entities rather than noun phrases or nouns. For this we use scispaCy[57] which was developed specifically for the processing of clinical, biomedical and scientific text. After the entities are extracted, we follow a similar method as the first step to get the most significant entities as the keywords/keyphrases. We use a pre-trained S-BERT model to generate the embedding vectors for the entities and the k sentences (lines 4-5). Note that in the previous step we calculated the embedding vectors for individual sentences, however, in this step we calculate the embedding vector for all k sentences taken together. This is done as we want to extract entities which are the most relevant as measured across sentences. In other words, we consider the top k sentences as the summary of the article (common application of the TextRank [24] algorithm). We then calculate the cosine similarity between each entity vector and the vector

Algorithm 1 Claim extraction

```
    Input: A list of sentences, $S = [s_1, s_2, s_3 \dots s_n]$
    Output: A reordered list of sentences $S'$ from $S$
 1: $W = [w_1, w_2, \dots w_m]$ be the list of conclusion indicator words
 2: for $i \leftarrow 1, n$ do
 3:     for $j \leftarrow 1, m$ do
 4:         if $w_j \in s_i$ then
 5:             $s_{i_{indicator\_score}} \leftarrow 1$
 6:         else
 7:             $s_{i_{indicator\_score}} \leftarrow 0$
 8:         end if
 9:     end for
10: end for
11: Load the S-BERT pre-trained model
12: for $i \leftarrow 1, n$ do
13:     $v_i \leftarrow get\_S - BERT\_vector(s_i)$
14: end for
15: for $i \leftarrow 1, n$ do
16:     for $j \leftarrow i + 1, n$ do
17:         $X \leftarrow cosine(v_i, v_j)$
18:     end for
19: end for
20: $G \leftarrow networkx\_graph\_from\_array(X)$, get the graph from the array $X$ using networkx library
21: for $i \leftarrow 1, n$ do
22:     $s_{i_{score}} \leftarrow TextRank(G)$, getting the TextRank scores for $s_i$
23:     $s_{i_{final\_score}} \leftarrow s_{i_{indicator\_score}} + s_{i_{score}}$
24: end for
25: Sort $s_i \in S$ in descending order of $final\_score$
```

of the k top sentences (lines 6-9). The entities are then ranked according to the decreasing order of similarity score. In order to determine the number of the top ranked entities which should be selected as the keywords/keyphrases, we define the similarity threshold as the root mean square (RMS) or quadratic mean (QM) of all the positive similarity values as follows:

$$RMS = \sqrt{\frac{\sum_{i=1}^{n}(x_i)^2}{n}}$$

where x_i is the set of positive similarity values and n is the total number of positive similarity values. We use RMS rather than arithmetic mean (AM) as $RMS \geq AM$ [58, 59] which implies that there would be a less number of similarity values above RMS in comparison to AM. This makes RMS a more selective measure. As the keywords/keyphrases will be used for generating queries, the choice of keywords/keyphrases should be precise and relevant to the health article. We first calculate the RMS of the similarity scores and then select those similarity scores which are greater than the RMS value (lines 11-15). We then get the corresponding entities for these scores which form the list of keywords/keyphrases (line 16).

3.2.3. *Query generation from the keywords/keyphrases:*

In the third step of our approach, we first group similar keywords/keyphrases within clusters

Algorithm 2 Keyword extraction

```
Input: List of $k$ sentences, $S' = [s'_1, s'_2, s'_3 \dots s'_k]$
Output: A list of keywords/keyphrases, $K$
for all sentences $s'$ in $S'$ do
    extract the entities, $E = [e_1, e_2, e_3 \dots e_n]$
end for
Load the S-BERT pre-trained model
$D \leftarrow get\_S - BERT\_vector(S')$
for all entities $e$ in $E$ do
    $c \leftarrow get\_S - BERT\_vector(e)$
    $similarities \leftarrow cosine(c, D)$
end for
$X \leftarrow root\_mean\_square(similarities)$
for all $s$ in $similarities$ do
    if $s > X$ then
        Get the corresponding entity $e$ and store them in a list
    end if
end for
The final list of entities from above are the keywords/keyphrases
```

and then create queries using the clustered keywords/keyphrases. The overview of the query generation process is shown in Algorithm 3.

For clustering, we use the concept of medical entity linking (MEL). MEL basically means the mapping of medical terms to a well defined vocabulary, usually some knowledge graph. A well defined knowledge graph in the medical domain specifically for the MEL task is UMLS which is short for Unified Medical Language System. UMLS brings together many health and biomedical vocabularies and provides a mapping structure among these vocabularies. It may be viewed as a comprehensive thesaurus or ontology of biomedical concepts. The idea behind using UMLS for our task is to make sure that similar keyphrases are grouped together in clusters. For this we use a pre-trained model on UMLS knowledge graph, SapBERT[60] which utilizes a BERT[61] model to help align entities to their synonyms using the UMLS knowledge graph (line 1-4). Based on the generated embeddings, cosine similarity metric is used to group similar keywords into clusters (lines 5-10). We use the k-means clustering algorithm for the purpose. To define the value of k we use the Silhouette coefficient score (line 11). The silhouette value is a measure of how similar an object is to its own cluster compared to other clusters. The formula is as follows:

$$\text{Silhouette coeff} = \frac{(x - y)}{max(x, y)}$$

where x is the mean intra-cluster distance of all instances within a cluster and y is the mean distance to the instances of the next closest cluster. The silhouette coefficient should be within [-1,1] with negative values meaning the samples might be wrongly clustered and near to +1 means the clusters the sample is far away from the neighboring clusters. A value of 0 indicates that the sample is on or very close to the decision boundary between two neighboring clusters. We vary the range of k for the k-means algorithm and calculate the silhouette score for each value of k. We then take the value of k where the silhouette score is the maximum

Algorithm 3 Query generation

```
    Input: List of p keywords/keyphrases, $K' = [k'_1, k'_2, k'_3 \dots k'_p]$
    Output: A Boolean query, $Q$
 1: Load the SapBERT pre-trained model
 2: for $i \leftarrow 1$ to $p$ do
 3:     $a_i \leftarrow get\_SapBERT\_vector(k')$
 4: end for
 5: for $i \leftarrow 1$ to $p$ do
 6:     for $j \leftarrow i + 1$ to $p$ do
 7:         $sim \leftarrow cosine(a_i, a_j)$
 8:         $clusters \leftarrow k\_means\_clustering(sim)$
 9:     end for
10: end for
11: $k \leftarrow max\_silhouette\_score(clusters)$
12: for all $c$ in clusters do
13:     for $i \leftarrow 1$ to $k$ do
14:         for $j \leftarrow i + 1$ to $k$ do
15:             $q \leftarrow ((c_{i_1}$ OR $c_{i_2} \dots$ OR $c_{i_n})$ AND $(c_{j_1}$ OR $c_{j_2} \dots$ OR $c_{j_m}))$, where $n$ and $m$ are the
                number  of elements in clusters $i$ and $j$ respectively
16:         end for
17:     end for
18: end for
19: $Q \leftarrow (q_1$ OR $q_2$ OR $\dots$ OR $q_N)$ where $N$ is the total number of $q$
```

[62]. In this way we get the optimum value of clusters for each article. The point of clustering semantically similar keywords is to create queries using Boolean operators which will be used to search relevant medical knowledge from a database. The OR operator is used to join terms in one cluster and the AND operator is used to join two clusters[63]. However, if there are more than two clusters then we use a combination of two clusters at a time and then create the queries for each combination (lines 12-18). This is done as it was found that joining more than two clusters by AND to form a query led to very selective and meaningless queries. This is because of the fact there may not be documents which contain terms from all the clusters, as a result of which no document is retrieved. In order to avoid it, we take two clusters at a time. After that we join all the queries by OR to get a single query (line 19) . For example, if there are 4 clusters, using a combination of 2 clusters at a time for the 4 clusters, we get 6 combinations in total, which gives us 6 queries. We then join all the queries using OR to get the single query.

3.3. *Proposed approach for relevant document retrieval*

The generated queries can be used for retrieving the documents from medical databases such as PubMed; this is based on a yes/no assessment of each document's membership in the result set, given the query is boolean. However, in order to retrieve the specific documents relevant to the specific claims, the documents need to be scored in such a way that the most relevant ones are filtered. This can be achieved by calculating a retrieval score, $\lambda()$ using similarities learned in an embedding space.

$$\lambda(c, a) = sim(\delta(c), \delta(a))$$

where $sim()$ can be a cosine or dot product similarity and $\delta()$ is the embedding method. Based on this, we first embed the claim and documents in the same vector space using an S-BERT model and then use cosine similarity to determine the most similar documents to the claims. The pseudo code is shown in Algorithm 5

Algorithm 4 Relevant document retrieval

```
Input: List of $k$ claims, $C = [c_1, c_2, c_3 \ldots c_k]$ and a list of documents $D = [d_1, d_2, d_3 \ldots d_n]$
Output: A ranked list of documents for each claim
Load the S-BERT model
$B \leftarrow get\_S - BERT\_vector(D)$
for all claims $c$ in $C$ do
    $a \leftarrow get\_S - BERT\_vector(c)$
    $similarities \leftarrow cosine(a, B)$
    Sort $D$ in descending order of $similarities$
end for
```

Hypothesizing that domain specific BERT models will perform better at retrieving the PubMed documents, we opted to experiment with such models. It has already been established that fine-tuning BERT models for specific tasks leads to better performance[54]. Based on this we use MS-MARCO[64], a standard dataset for retrieval tasks, as an off-the-shelf dataset for our fine-tuning purpose. We used the PubMedBert[65] model[e] as our base BERT model and fine-tuned it using the MS-MARCO dataset with a MarginMSE loss[66] using a bi-encoder architecture[54]. We also experimented with other domain specific BERT models but PubMedBert yielded the best results which is why we used it for the rest of our experiments.

For the fine-tuning process, we used a triplet training environment where the positives used are included in the MS-MARCO dataset and the negatives are taken from the training data available on the Huggingface Transformers[67] S-BERT repository[f] Sentence embeddings are computed independently for the queries, positives and the negatives. A dot product similarity (denoted by sim) is computed between the (query, positives) and (query, negatives). Training is done by forming triplets from the query, positive and negatives. The objective is to minimize the mean square error loss between $A = |sim(Query, Pos) - sim(Query, Neg)|$ and $B = |gold\_sim(Query, Pos) - gold\_sim(Query, Neg)|$.

$$\mathcal{L}_{(\mathrm{A,B})} = \frac{1}{|A|} \sum_{a \epsilon A, b \epsilon B} (a - b)^2$$

Here A is computed by using the PubMedBert model and B is the gold standard scores.

4. Experiment and results

We describe the details of the experiments done for each of the steps involved in the proposed method and analyse the results in the following sub-sections.

4.1. *Creation of gold standard dataset*

[e] `https://huggingface.co/microsoft/BiomedNLP-PubMedBERT-base-uncased-abstract-fulltext`

[f] `https://huggingface.co/datasets/sentence-transformers/msmarco-hard-negatives/resolve/main/msmarco-hard-negatives.jsonl.gz`

Since there was no available dataset for the specific task we address, we curated our own dataset using medical articles available from the website **www.naturalnews.com** to check the efficiency of the proposed technique. We collected various health articles from this website pertaining to topics such as cancer, coronavirus, influenza, Alzheimer's. In total we collected 88 articles, each with a valid heading. The headings were used in the annotation of the dataset that will be described shortly. However, our proposed method does not assume the existence of a heading hence it is applicable to any online post (e.g. social media post). Each sentence within each article is manually annotated with 0 indicating a non-claim sentence and 1 indicating a relevant claim sentence. The annotation is done based on how relevant that particular sentence is to the heading of the article (i.e., relevance to the key claim). Furthermore, for each article, the list of phrases directly relevant to the claim was manually extracted from its heading.

4.2. *Identification of claim sentences*

In the first step of the approach, we use a pre-trained model of S-BERT[54], `roberta-large-nli-stsb-mean-tokens` as it has the highest semantic textual similarity (STS) performance score mentioned in [54]. For generating the graph in the same step we use the python library `networkx`[g] Also for generating the TextRank [24] scores, we use the pagerank [36] function from the same library. The ranked sentences from the first step of the approach are evaluated with nDCG values using the binary sentence relevance scores (Ref: Sec) as labels. We calculate the overall nDCG value for the whole article and nDCG@k where $k \in \{1, 3, 5, 7\}$ denotes the position. As baselines we implement a few methods which range from using bag of words implementation using tf-idf [68], word embedding vectors using GloVe [69], sentence embedding vectors using Facebook's Infersent[70] and Google's Universal Sentence Encoder(USE)[71]. We experiment using both the conclusion indicator words and without using the indicator words shown in Fig 3 and Fig 2 respectively.

We can see from the figure that USE[71] has a nearly similar performance as S-BERT[54]. This can be attributed to the fact that both these models are based on the transformer architecture[72] and captures the context of text. Interestingly, the bag of words model performs better than the Infersent[70] model. This can be attributed to the fact that our dataset is smaller and domain specific. Since Infersent[70] uses pre-trained word embedding vectors which are not domain specific, the performance is lower than the bag of words model. We can see from the Fig 3 that introducing the indicator words improves the results which indicates that indicator words helps in the identification of the claim sentences.

We also evaluate the performance on another dataset taken from the work by [73]. Although the dataset pertains to the biomedical literature domain, we decided to use it as online health related information contains biomedical terms. The dataset has 1,500 scientific abstracts from the biomedical domain with each sentence annotated by experts indicating whether the sentence presents a scientific claim. The dataset however has some abstracts with no claim sentence. We filter out such abstracts for evaluation purposes. The final evaluation dataset consists of 1344 abstracts with each abstract having at least one sentence annotated as a claim. We use the same experimental setup mentioned above and the results using both indicator words and without it are shown below in Fig 5 and Fig 4 respectively.

[g] `https://github.com/networkx/networkx`

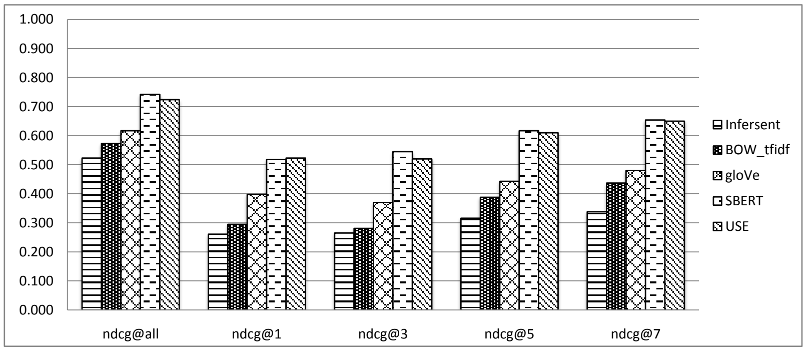


Fig. 2. nDCG comparison without conclusion indicators

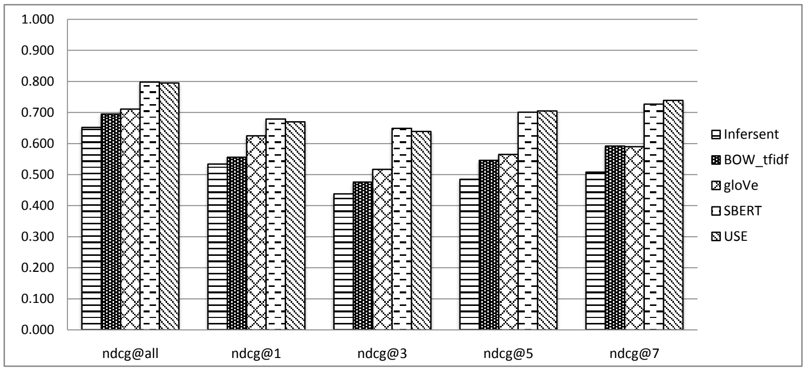


Fig. 3. nDCG comparison with conclusion indicators

It can be seen that adding indicator words improve the nDCG scores which shows that such words are helpful in identifying claim sentences in an unsupervised setting. It is interesting to note from the results that the GloVe embeddings performed better than USE and scores slightly less than S-BERT which shows that it is a strong baseline for such work.

4.3. *Keyword extraction*

In the second step, we use the following S-BERT [54] pre-trained model, `distilroberta-base-paraphrase-v1` for generating the embeddings. We use the following scispaCy model `en_core_sci_lg`[h]for entity extraction. For demonstrational purposes, we show a snapshot of the heading and the various keywords/keyphrases extracted for the respective articles in Table 1.

To evaluate the extracted keywords/keyphrases, we compare them against the keywords/ keyphrases manually extracted from the headings of the articles. It should be noted that the manually extracted keywords/keyphrases from the headings are used just for evaluation purposes, the heading membership of keywords/keyphrases is not a part of the keyword/keyphrase extraction process. As baseline, we use a few state of the art unsupervised keyword/keyphrase extraction models which includes graph-based approaches like TextRank[24], TopicRank[40],

[h] `https://allenai.github.io/scispacy/`

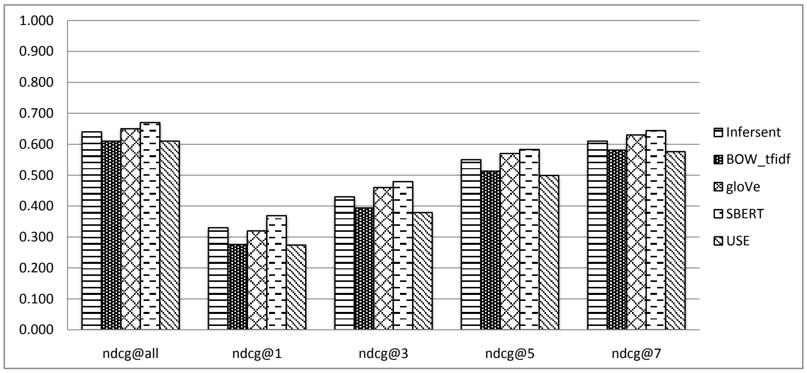


Fig. 4. nDCG comparison without conclusion indicators for the dataset from [73]

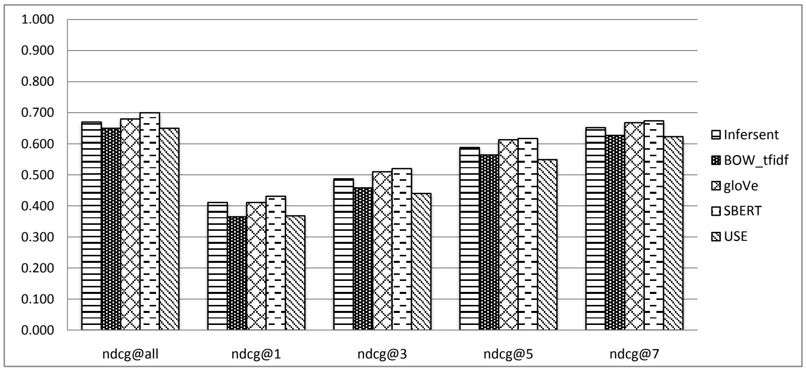


Fig. 5. nDCG comparison with conclusion indicators for the dataset from [73]

SingleRank[74], PositionRank[35] and MultipartiteRank[75]. For implementation of these models, we use the python toolkit by [76]. Since the baseline models require a value of k, where k is the number of keywords/keyphrases to be extracted, we experiment using different values of k. We calculate the average precision, recall and f-measure obtained by the methods across all the articles [77]. We do not consider keywords/keyphrases which may be semantically similar in meaning to the heading keywords/keyphrases. The results are presented in Table 2.

We can observe from the table that there is an increase in the value of each metrics when using our approach. It can be observed from the table that as the value of k increases, precision values get lower and recall values get higher. This is due to the fact that the number of keywords/keyphrases extracted from the headings of most of the articles lie between 2 or 3. So when k increases beyond these numbers, the precision values drop even for the most accurate ranking.

We also use another dataset from [78] in order to evaluate the key phrases extraction method that we have proposed. The dataset consists of 3048 biomedical abstracts from PubMed with author annotated keywords/keyphrases. Precision, recall and f-measure were calculated in the similar way by taking different values of k. The results are shown in Table 3. It can be seen from the table that the method we proposed performs better than the other

Table 1. Extracted keywords/keyphrases from respective articles

Heading	**Extracted Keywords/Keyphrases**
For the ladies: eating walnuts can help halt breast cancer	’walnut’, ’mammary gland cancer’, ’breast cancers’, ’walnut consumption’, ’breast tumors’
Green tea, zinc proving to be better than hydroxychloroquine at fighting coronavirus infections	’zinc’, ’polyphenol’, ’epigallocatechin gallate’, ’coronavirus’, ’asian countries’, ’antiviral nutrients’, ’green tea’
Homeland security scientist confirms that natural sunlight kills coronavirus	’natural sunlight’, ’uvc light’, ’viruses’, ’ultraviolet uv rays’,’coronavirus’, ’wuhan coronavirus’

Table 2. Comparison of our method with baselines for our dataset

		k=5	**k=10**	**k=15**	**k=20**
MultipartiteRank[75]	**P**	0.19	0.12	0.10	0.09
	R	0.39	0.50	0.59	0.62
	F1	0.25	0.19	0.17	0.16
PositionRank[35]	**P**	0.19	0.14	0.11	0.10
	R	0.42	0.57	0.63	0.66
	F1	0.26	0.22	0.18	0.16
SingleRank[74]	**P**	0.17	0.14	0.10	0.09
	R	0.35	0.59	0.61	0.62
	F1	0.22	0.22	0.18	0.15
TextRank[24]	**P**	0.10	0.10	0.09	0.08
	R	0.23	0.42	0.54	0.61
	F1	0.13	0.16	0.15	0.15
TopicRank[40]	**P**	0.16	0.10	0.08	0.07
	R	0.33	0.41	0.43	0.44
	F1	0.21	0.16	0.13	0.12
Our method	**P**	**0.24**	**0.16**	**0.12**	**0.10**
	R	**0.52**	**0.68**	**0.72**	**0.73**
	F1	**0.33**	**0.25**	**0.20**	**0.17**

baselines.

Table 3. Comparison of different keyword/keyphrase extracting algorithm with our method using dataset from [78]

		k=10	**k=15**	**k=20**	**k=25**	**k=30**
MultipartiteRank[75]	**P**	0.25	0.22	0.20	0.18	0.17
	R	0.20	0.26	0.30	0.34	0.37
	F1	0.22	0.22	0.23	0.23	0.23
PositionRank[35]	**P**	0.18	0.17	0.17	0.16	0.16
	R	0.15	0.21	0.27	0.31	0.36
	F1	0.16	0.19	0.21	0.21	0.22
SingleRank[74]	**P**	0.14	0.15	0.15	0.16	0.16
	R	0.12	0.19	0.25	0.30	0.35
	F1	0.13	0.16	0.19	0.21	0.22
TextRank[24]	**P**	0.12	0.12	0.13	0.13	0.13
	R	0.10	0.15	0.20	0.25	0.29
	F1	0.11	0.13	0.16	0.17	0.18
TopicRank[40]	**P**	0.25	0.22	0.20	0.18	0.17
	R	0.19	0.25	0.29	0.32	0.35
	F1	0.21	0.23	0.23	0.23	0.23
Our method	**P**	**0.27**	**0.25**	**0.23**	**0.21**	**0.20**
	R	0.20	**0.28**	**0.34**	**0.39**	**0.43**
	F1	**0.23**	**0.26**	**0.27**	**0.27**	**0.27**

4.4. *Query generation from keywords*

In this step, for clustering keywords/keyphrases based on similarity, we use the pre-trained model of SapBERT[60], `cambridgeltl/SapBERT-from-PubMedBERT-fulltext` from the Huggingface Transformers library[67] to generate the embeddings of keywords/keyphrases. We use the k-means clustering function from the sklearn library.[i] To define the value of k we use the Silhouette coefficient score which is available in sklearn.[j]

For evaluation of the queries, we need some results that can be regarded as desirable. It is intuitive to think that documents generated from headings of articles are likely to be relevant; thus, the titles of retrieved documents (we will use title to refer to these hereon) based on queries generated from the article headings (called heading queries hereon) forms a benchmark to compare the results of our keyword queries against. While this evaluation can potentially be critiqued from obvious directions, we use this as a best effort evaluation, given that we do not have labelled data pertinent to this specific task. We use a python library Biopython[k]which provides access to the Entrez[l] API of NCBI for searching and retrieving titles from the PubMed databases. The Entrez function Efetch has a sort parameter which sorts the retrieved titles according to the given option. For our purpose we have used the "relevance" option which retrieves the most relevant titles for a query.

4.4.1. *Evaluation measures and their comparative motivations*

For evaluating the keyword queries, we calculate the precision and recall for all the articles based on the heading queries. The heading queries are taken as the gold-standard and titles retrieved by them are used as the gold-standard results. We retrieve the top m titles using the heading queries and top n titles using the queries generated by the extracted keywords/keyphrases where $m = 1, 2, 3$ and $n = 3, 5, 10$. The precision and recall is calculated based on the presence of the gold standard results within the range of n. It was found that few of the articles did not retrieve any titles for the heading query. This is because of the terms present in the query are not indexed in PubMed which is why no title is retrieved when the query is passed. For evaluation purposes those articles are not taken into consideration. The results are shown in Table 4.

It was also found that there are a few articles whose heading terms were broad and vague whereas the article body talked about a specific topic which is part of the broad heading term but not the same. For example, an article had the heading terms as "Vitamin A" and "skin cancer". But the article body talked specifically about "squamous cell carcinoma" which is a certain type of skin cancer. This is why the title retrieved by the heading query is not retrieved by the keyword query within the top positions. It maybe retrieved at lower positions. There are also articles whose heading query terms are similar to the keyword query terms but not exactly the same. Due to these reasons, another method of evaluation of the queries was considered. We compute the cosine similarity between the documents retrieved by the heading queries and the keyword queries and based on the cosine similarity values, we calculate the average precision and recall along with the f-score across all the articles; this way, identifying similar but non-identical content will not be penalized. The pseudo-

[i] `https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html`

[j] `https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html`

[k] `https://biopython.org/docs/1.75/api/Bio.Entrez.html`

[l] `http://www.ncbi.nlm.nih.gov/Entrez/`

code is provided in Algorithm 5. The precision and recall is calculated based on the cosine similarity threshold value, since measures such as precision and recall call for discretization. For precision, we first check how many keyword query documents have cosine similarity greater than the threshold with the heading query documents (line 1 to 6). We then check how many of these have a lesser similarity value than the threshold (lines 7 to 12). Based on these, the precision, recall and f-score is calculated (line 13-14).

Algorithm 5 Calculating Precision, Recall and F-score considering the semantic similarity between the documents retrieved by heading queries and keyword queries

Input: List of keyword query documents: $Q = [q_1, q_2, q_3 \dots q_n]$, List of heading query documents: $H = [h_1, h_2, h_3 \dots h_m]$, similarity threshold: t

Output: Precision, recall and f-score for threshold value

```
 1: $relevant = 0$
 2: for all $h_i$ in $H$ do
 3:     if there exist $q_j \in Q$ such that $s(q_j, h_i) > t$ then
 4:         $relevant++$
 5:     end if
 6: end for
 7: $irrelevant = 0$
 8: for all $q_j$ in $Q$ do
 9:     if there does not exist $h_i$ such that $s(q_j, h_i) > t$ then
10:         $irrelevant++$
11:     end if
12: end for
13: $precision = \frac{relevant}{relevant+irrelevant}$, $recall = \frac{relevant}{m}$
    $f\_score = \frac{2*precision*recall}{precision+recall}$
```

In order to check for the similarity between the documents retrieved by the heading query/queries and the keyword queries, we calculate the cosine similarity between their sentence embedding vectors computed using S-BERT [54]. However, a threshold needs to be determined for the cosine similarity above which documents may be considered similar. For calculating the threshold of cosine similarity, we use the BIOSSES dataset [79]. The dataset contains 100 pairs of sentences annotated by 5 annotators. We take the average score of the annotators and according to the definition of the scores, we choose those above 2 as relevant pairs and below 2 as non-relevant pairs. We then use sentence embeddings to calculate the cosine similarity between each pair. After that, for each threshold value from 0.1 to 0.9, we add the relevant pairs above the threshold value with the non-relevant pairs below threshold value and divide it by 100 (total number of sentence pairs). We then choose the value where we get the maximum as the threshold value. Following the method, we get 0.5 as the threshold value for cosine similarity. Taking this as the reference point, we perform a few experiments whose results are shown in Table 5 and Table 6.

We first show the average precision, recall and f-score values for threshold value 0.5 across different values of m and n for three different S-BERT pre-trained models along with three fine-tuned biomedical language models, BioBERT [80], BlueBERT [81] and BiomedRoBERTa [82]. Recently a surge of language models for the biomedical domain has been seen [83]. However, it's been found that BERT models do not perform well for specific tasks without fine-tuning [54]. Therefore, in order to use such models for generating sentence embeddings, we fine-tune them using the S-BERT framework. We use the SNLI [84], the MultiNLI [85] and the STS-b [86] datasets for fine-tuning which is done according to the documentation[m]of S-BERT. We also find the average position value for the first occurrence of the title having similarity value greater than the threshold across all articles. For this experiment also, we find it for various values of m and n. The analysis indicates the improved accuracy of retrieval achieved by the fine-tuned BioBert.

4.4.2. *Analysis of the results*

From Table 4, we can see that as we increase the value of m, the recall decreases for the same value of n. However, as we increase the value of n, the recall increases. This is due to the fact that the more titles are retrieved by the keyword queries, the chances of getting the gold standard results within those titles also increases. But this also leads to a decrease in the precision as can be seen from the table.

Table 4. Average precision and recall for titles retrieved by keyword queries (n=3, 5, 10)

Max titles retrieved by keyword query (n)	**Top 1 title retrieved by heading query (m=1)**		**Top 2 titles retrieved by heading query (m=2)**		**Top 3 titles retrieved by heading query (m=3)**	
	Avg P	**Avg R**	**Avg P**	**Avg R**	**Avg P**	**Avg R**
n=3	0.109	0.329	0.164	0.265	0.185	0.215
n=5	0.073	0.367	0.116	0.310	0.137	0.257
n=10	0.048	0.481	0.078	0.411	0.093	0.346

In order to consider titles which has a semantic similarity with the gold standard results, we first find the cosine similarity threshold value. Based on the experiment with the cosine similarity threshold value, we can see from Table 5 that the average precision value increases as we increase the value of m for each n. We can also see that the average recall increases as the value of n increases. This shows that relevant titles are retrieved by the keyword queries within the various positions of n. Out of the three pre-trained models `stsb-roberta-large` gave the best results and out of the fine-tuned models, the BioBERT model gave the best results. If we compare the f-scores for all the models, it can be seen that the fine-tuned BioBERT model gives the best result. This finding is in line with the recent work [83] which found that BioBERT performs better with biomedical tasks. For our next experiment we compare the `stsb-roberta-large` as well as the fine-tuned BioBERT model.

From the Table 6 we can see that the average position increases as we increase the value of n which is due to the fact for some articles the relevant title is found in a lower position (high value of n means low position). Also, we find the percentage of articles that has the first occurrence of relevant title at the first position. It can be seen as we increase the value of m the percentage decreases. The increase in average precision and decrease in the percentage shows that relevant titles are retrieved but in positions other than the first position. Also it can be seen that the fine-tuned BioBERT gives better results than the pre-trained model as the average position can be seen at higher positions (lower value). This shows that the

[m] `https://www.sbert.net/docs/training/overview.html`

Table 5. Average precision, recall and f-1 score for cosine similarity threshold value 0.5 for different models

Models	**Max title retrieved by heading query**	**Max doc retrieved by keyword query(n=3)**			**Max title retrieved by keyword query(n=5)**			**Max title retrieved by keyword query(n=10)**		
		P	**R**	**F-1**	**P**	**R**	**F-1**	**P**	**R**	**F-1**
stsb-roberta-large	m=1	0.53	0.83	0.64	0.43	0.84	0.57	0.27	0.91	0.41
	m=2	0.68	0.81	0.74	0.59	0.86	0.64	0.44	0.90	0.56
	m=3	0.75	0.79	0.77	0.68	0.85	0.75	0.54	0.90	0.67
paraphrase-distilroberta-base-v2	m=1	0.48	0.75	0.58	0.37	0.80	0.50	0.23	0.82	0.36
	m=2	0.59	0.70	0.64	0.47	0.72	0.57	0.34	0.75	0.47
	m=3	0.65	0.67	0.66	0.54	0.71	0.61	0.42	0.75	0.54
paraphrase-mpnet-base-v2	m=1	0.51	0.77	0.61	0.43	0.82	0.56	0.28	0.84	0.42
	m=2	0.64	0.77	0.69	0.55	0.78	0.64	0.40	0.81	0.53
	m=3	0.71	0.74	0.72	0.64	0.76	0.69	0.52	0.81	0.63
Fine-tuned-Biobert	m=1	0.63	0.86	0.73	0.52	0.87	0.65	0.35	0.88	0.50
	m=2	0.75	0.85	0.79	0.65	0.85	0.74	0.50	0.86	0.63
	m=3	0.79	0.82	0.80	0.70	0.82	0.76	0.57	0.85	0.68
Fine-tuned-Bluebert	m=1	0.55	0.81	0.65	0.48	0.83	0.60	0.30	0.84	0.44
	m=2	0.69	0.80	0.74	0.58	0.82	0.68	0.46	0.85	0.59
	m=3	0.76	0.79	0.77	0.66	0.80	0.72	0.53	0.84	0.65
Fine-tuned-BiomedRoberta	m=1	0.45	0.77	0.57	0.33	0.82	0.47	0.21	0.83	0.33
	m=2	0.56	0.67	0.61	0.45	0.74	0.56	0.39	0.78	0.52
	m=3	0.63	0.64	0.63	0.53	0.71	0.60	0.40	0.77	0.53

fine-tuned model is more suitable for our data than the pre-trained model.

4.5. *Retrieving relevant documents*

As annotating health/medical information retrieved from PubMed requires domain expertise, we use the SCIFACT[46] dataset for evaluation of the relevant documents retrieval results. The dataset consists of a train file, a test file and a validation file. Since the test file is used for a shared task and does not contain annotations, we decided to use the validation file as the test file. It consists of 300 claims and the train file consists of 809 claims. The dataset also has around 5200 documents as a corpus file from where relevant documents are to be retrieved. For each claim there is at least one document annotated as relevant from the corpus. For the

Table 6. Position details of first occurrence of titles greater than threshold

Maximum titles retrieved by heading query	**Maximum titles retrieved by keyword query**	**Average Position of first occurrence of document greater than threshold**		**% of articles having the first occurrence of document at the 1st position**	
		stsb-roberta-large	**fine-tuned-BioBERT**	**stsb-roberta-large**	**fine-tuned-BioBERT**
m=1	n=3	1.01	0.98	69.2 %	73.1 %
	n=5	1.03	1.00	70.5 %	74.3 %
	n=10	1.46	1.07	70.5 %	74.3 %
m=2	n=3	1.04	1.04	62.6 %	68.0 %
	n=5	1.24	1.08	63.3 %	66.7 %
	n=10	1.54	1.22	63.3 %	66.7 %
m=3	n=3	1.04	1.01	60.0 %	65.0 %
	n=5	1.28	1.07	60.2 %	63.0 %
	n=10	1.66	1.26	60.2 %	63.0 %

retrieval purpose, cosine similarity is first computed between each claim and the documents present and a ranked list of documents are produced for each claim based on the similarity score. For the purpose of evaluation of our experiments, we used the BEIR[87] framework to calculate recall@k values where $k = 3, 5, 10$ are the retrieved results for each claim.

4.5.1. *Baselines used*

We compared several baseline models with our model which are hereby explained. DPR[88] is a bi-encoder trained with a single BM25 hard negative and in-batch negatives. The publicly available multi-DPR model[n][o] which has been trained on four QA datasets: NQ [89], TriviaQA [90], WebQuestions [91] and CuratedTREC [92] is used for the experimental results comparison. ANCE [93] is another bi-encoder constructing hard negatives from an Approximate Nearest Neighbor (ANN) index of the corpus, which in parallel updates to select hard negative training instances during fine-tuning of the model. We use the S-BERT ANCE model available in Huggingface[p] for our experiments. USE-QA[q] is a universal sentence encoder[71] model fine-tuned on the SQuAD[94] dataset. Paragraph-Joint[95] uses the BioSentVec[96] model for the retrieval purpose. BioSentVec is a sent2vec[97] model which is trained over PubMed abstracts. It is a 700 dimensional model which is used to compute sentence embeddings for biomedical data. The SPLADE v2[98] model[r] is a DistilberT-base[99] model which is fine-tuned over the MS-MARCO dataset. A publicly available cross-encoder model[s] is then used to retrieve hard negatives from the dataset which are added to the original dataset. The fine-tuned Distilbert model is again further tuned over this new dataset using MarginMSE loss[66]. The final model is the SPLADE v2 model. The TCT[100] model is a dense retrieval model which is based on the ColBERT[101] model. The ColBERT model is a BERT model where both query and document are encoded independently. Every query embedding interacts with all document embeddings by computing maximum similarity (e.g., cosine), and the scalar outputs of these operators are summed across query terms. This enables ColBERT to retrieve top-k results directly from a large corpus. The TAS-B model[102] is knowledge distilled model where the student model is a Distilbert model which uses Balanced Topic Aware Sampling (TAS-Balanced) to compose dense retrieval training batches. The sampling technique compose batches based on queries clustered in topics and then select passage pairs so as to balance pairwise teacher score margins where the teacher model used is a ColBERT model as well as a concatenated BERT model. We use the sentence-transformer TAS-B[t] model for our experiments. The BM25[103] model used in the experiments is the Okapi BM25[104] model implemented with the Python package `https://pypi.org/project/rank-bm25/`. The BM25+CE model is a re-ranker model which first uses Okapi BM25[104] to retrieve the first 100 documents and after that a cross-encoder model is used to rank those 100 documents. The cross-encoder model used is a fine-tuned sentence-transformer ELECTRA[105] base model[u] .

[n] https://huggingface.co/facebook/dpr-question_encoder-multiset-base

[o] https://huggingface.co/facebook/dpr-ctx_encoder-multiset-base

[p] https://huggingface.co/sentence-transformers/msmarco-roberta-base-ance-firstp

[q] https://tfhub.dev/google/universal-sentence-encoder-qa/3

[r] https://github.com/naver/splade/tree/main/weights/distilsplade_max

[s] https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-12-v2

[t] https://huggingface.co/sentence-transformers/msmarco-distilbert-base-tas-b

[u] https://huggingface.co/cross-encoder/ms-marco-electra-base

4.5.2. *Experimental results and analysis*

We first experimented in a zero-shot evaluation setting wherein all the models are used as it is without any further fine-tuning over the SCIFACT dataset. Table 7 shows the zero-shot evaluation results of our bi-encoder model, S-PubMedBert[v] compared with the other models.

Table 7. Zero-shot evaluation results

Model	**recall@3**	**recall@5**	**recall@10**
S-PubMedBert (our model)	0.693	**0.756**	**0.819**
ANCE[93]	0.512	0.561	0.633
DPR[88]	0.264	0.292	0.370
USE-QA[71]	0.324	0.365	0.418
Paragraph-Joint (BioSentVec)[95]	0.560	0.665	0.750
TCT-Colbert[100]	0.486	0.534	0.601
Distilbert-TAS-B[102]	0.554	0.614	0.689
BM25+CE	0.626	0.664	0.706
SPLADE v2[98]	**0.701**	0.734	0.791
BM25 [103]	0.500	0.556	0.625

We can see that the results for some of the models are close to each other, therefore, for analysing the results in a better way, a statistical test was conducted. Since we are comparing different models, the Friedman test[106] is used for the statistical significance test. The null hypothesis is that all models perform similarly. If the null hypothesis is rejected from the previous test, the Nemenyi test[107] is performed to find the statistically significant methods. We conduct the tests for all the recall@k values and the p-values of all models are plotted as heatmaps. We find that the null hypothesis is rejected in all the cases as the models have differences in performance.

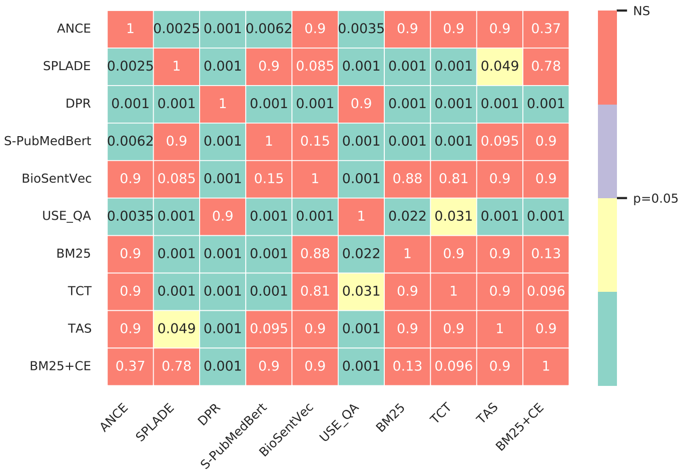
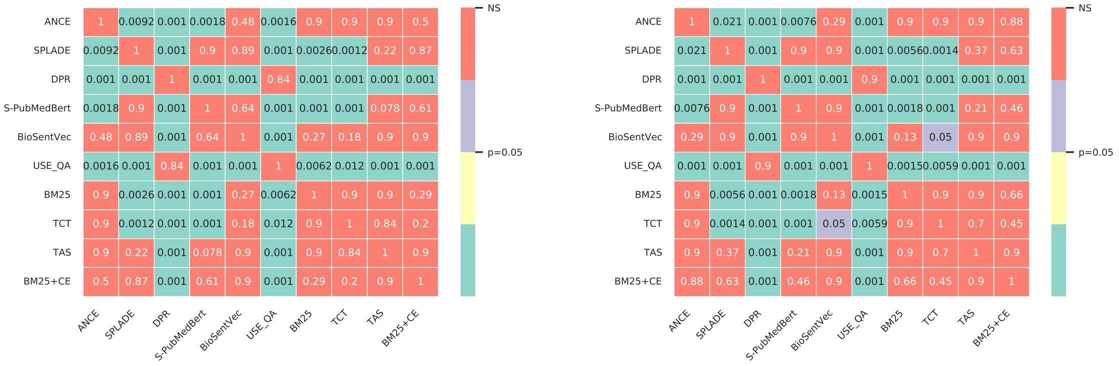


Fig. 6. Heatmap for recall@3 of zero shot evaluation results

For recall@3: The pair-wise p-values from the Nemenyi test are shown in Fig 6. From the figure we can see that DPR and USE-QA have statistically similar results and are less significant than the other methods. We can also see that BM25+CE and BM25 are statistically

[v] `https://huggingface.co/pritamdeka/S-PubMedBert-MS-MARCO`



	ANCE	SPLADE	DPR	S-PubMedBert	BioSentVec	USE_QA	BM25	TCT	TAS	BM25+CE
ANCE	1	0.0092	0.001	0.0018	0.48	0.0016	0.9	0.9	0.9	0.5
SPLADE	0.0092	1	0.001	0.9	0.89	0.001	0.0026	0.0012	0.22	0.87
DPR	0.001	0.001	1	0.001	0.001	0.84	0.001	0.001	0.001	0.001
S-PubMedBert	0.0018	0.9	0.001	1	0.64	0.001	0.001	0.001	0.078	0.61
BioSentVec	0.48	0.89	0.001	0.64	1	0.001	0.27	0.18	0.9	0.9
USE_QA	0.0016	0.001	0.84	0.001	0.001	1	0.0062	0.012	0.001	0.001
BM25	0.9	0.0026	0.001	0.001	0.27	0.0062	1	0.9	0.9	0.29
TCT	0.9	0.0012	0.001	0.001	0.18	0.012	0.9	1	0.84	0.2
TAS	0.9	0.22	0.001	0.078	0.9	0.001	0.9	0.84	1	0.9
BM25+CE	0.5	0.87	0.001	0.61	0.9	0.001	0.29	0.2	0.9	1

(a) Heatmap for recall@5

	ANCE	SPLADE	DPR	S-PubMedBert	BioSentVec	USE_QA	BM25	TCT	TAS	BM25+CE
ANCE	1	0.021	0.001	0.0076	0.29	0.001	0.9	0.9	0.9	0.88
SPLADE	0.021	1	0.001	0.9	0.9	0.001	0.0056	0.0014	0.37	0.63
DPR	0.001	0.001	1	0.001	0.001	0.9	0.001	0.001	0.001	0.001
S-PubMedBert	0.0076	0.9	0.001	1	0.9	0.001	0.0018	0.001	0.21	0.46
BioSentVec	0.29	0.9	0.001	0.9	1	0.001	0.13	0.05	0.9	0.9
USE_QA	0.001	0.001	0.9	0.001	0.001	1	0.0015	0.0059	0.001	0.001
BM25	0.9	0.0056	0.001	0.0018	0.13	0.0015	1	0.9	0.9	0.66
TCT	0.9	0.0014	0.001	0.001	0.05	0.0059	0.9	1	0.7	0.45
TAS	0.9	0.37	0.001	0.21	0.9	0.001	0.9	0.7	1	0.9
BM25+CE	0.88	0.63	0.001	0.46	0.9	0.001	0.66	0.45	0.9	1

(b) Heatmap for recall@10

Fig. 7. Heatmap of Nemenyi test for zero shot evaluation methods

different and from the Table 7 we can see that re-ranking proves to be a better option than just using BM25. It can also be seen that SPLADE performs better than S-PubMedBert at recall@3. Overall, S-PubMedBert's improved performance is seen to be statistically significant over the majority of methods, and that SPLADE's improvements over S-PubMedBert are not statistically significant.

For recall@5: From Fig 7(a) it can be seen that the pair-wise p-value results are similar to recall@3. In this case however, it can be seen that S-PubMedBert performs better than SPLADE which is also apparent from Table 7.

For recall@10: From Fig 7(b), it can be seen that the pair-wise p-value results are similar to that of recall@5. Although the recall values increase at higher positions, statistically there is not much change in the significance results for the methods except for TCT where it is less significant than recall@3 and recall@5.

Overall from the above results, it can be seen that S-PubMedbert and SPLADE perform better than the other methods. DPR and USE-QA have the lowest significance statistically. Also, the BM25+CE model performs better than BM25 which shows that re-ranking BM25 results with a cross-encoder model improves the recall value results.

We also experimented in a setting where we fine-tuned our model over the SCIFACT dataset. For comparison we used several models that were used in the zero-shot comparison. However, some models could not be fine-tuned which is why we did not use those models for the experiment. For fine-tuning, we used triplets where the positive passage was taken from the dataset itself and the negative passages were mined using BM25. For each pair of (query, positive passage) maximum 10 negatives were mined. After that the model was trained using the triplets with a multiple negatives ranking loss function[108] using the S-BERT framework. The fine-tuned model was then evaluated accordingly. All the models which were compared were fine-tuned in the same way as our model.

From Table 8, it can be seen that our fine-tuned model performs better than the other fine-tuned models. However, to find the statistically significant models, a test was conducted similar to the above.

For recall@3: We got statistically significant result difference between ANCE and S-

Table 8. Fine-tuned model evaluation results

Model	**recall@3**	**recall@5**	**recall@10**
Fine-tuned S-PubMedBERT	**0.870**	**0.910**	**0.943**
Fine-tuned ANCE	0.731	0.755	0.787
Fine-tuned DPR	0.651	0.678	0.735
Fine-tuned TCT-Colbert	0.742	0.764	0.791
Fine-tuned Distibert-TAS-B	0.755	0.771	0.795
Fine-tuned SPLADE-v2	0.739	0.755	0.771

PubMedBert, DPR and S-PubMedBert, SPLADE and S-PubMedBert, TCT-Colbert and S-Pubmedbert and TAS-B and S-Pubmedbert. The p-values of S-PubMedBert compared with the other models are all <0.05. Based on the results it can be seen that S-PubMedBert performs better than the other models for recall@3 as shown in Fig 8.

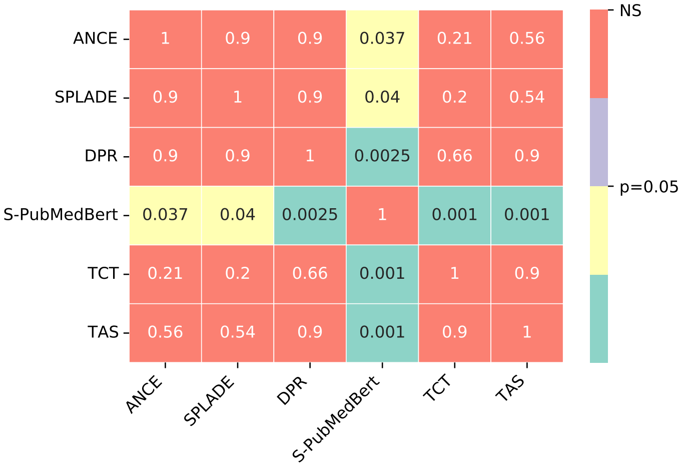


Fig. 8. Heatmap for recall@3

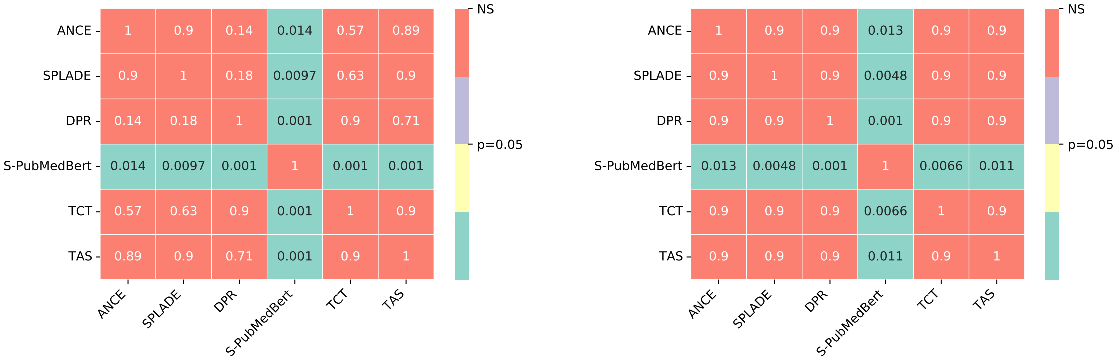
For recall@5: Based on the Nemenyi test, the S-Pubmedbert model was again found to be performing the best. The p-values of S-PubMedBert compared with the other models are all < 0.05. Based on the results we can see that S-PubMedBert performs better than the other models for recall@5 which can be seen from Fig 9(a).

For recall@10: Based on the Nemenyi test, the p-values of S-PubMedBert compared with the other models are all < 0.05. Therefore, from the results we can see that S-PubMedBert performs better than the other models for recall@10 as seen in Fig 9(b).

Based on the above results of the Nemenyi test we can conclude that the fine-tuned S-PubMedBert[w] model over SCIFACT is statistically better than the rest of the fine-tuned models over the same dataset.

From the overall results, it can be seen when using domain specific embeddings, i.e S-PubMedbert, the fine-tuned results show statistical significance than the fine-tuned models using general embeddings. Using a domain specific model yields a much better performance for scientific claim document retrieval task than using a general purpose model which confirms our hypothesis.

[w] `https://huggingface.co/pritamdeka/S-PubMedBert-MS-MARCO-SCIFACT`



	ANCE	SPLADE	DPR	S-PubMedBert	TCT	TAS
ANCE	1	0.9	0.14	0.014	0.57	0.89
SPLADE	0.9	1	0.18	0.0097	0.63	0.9
DPR	0.14	0.18	1	0.001	0.9	0.71
S-PubMedBert	0.014	0.0097	0.001	1	0.001	0.001
TCT	0.57	0.63	0.9	0.001	1	0.9
TAS	0.89	0.9	0.71	0.001	0.9	1

(a) Heatmap for recall@5

	ANCE	SPLADE	DPR	S-PubMedBert	TCT	TAS
ANCE	1	0.9	0.9	0.013	0.9	0.9
SPLADE	0.9	1	0.9	0.0048	0.9	0.9
DPR	0.9	0.9	1	0.001	0.9	0.9
S-PubMedBert	0.013	0.0048	0.001	1	0.0066	0.011
TCT	0.9	0.9	0.9	0.0066	1	0.9
TAS	0.9	0.9	0.9	0.011	0.9	1

(b) Heatmap for recall@10

Fig. 9. Heatmap of Nemenyi test for fine-tuned models

5. Key Findings

Given that we have covered a number of different tasks, techniques and evaluations, we summarize our key findings to convey the bigger picture succinctly.

1. Indicator words play an important role for finding claim sentences in online health related content.

2. Domain specific named entities are better suited as candidate keys than noun phrases/nouns for keyword/keyphrase extraction from biomedical text.

3. Fine-tuned BioBERT using S-BERT performed better than pre-trained generic S-BERT models for biomedical sentence similarity tasks which is in line with the findings by [83].

4. Fine-tuned PubMedBert model performed better in retrieving relevant information which shows that for health/biomedical text retrieval tasks, domain specific transformer models performs better than generic models when fine-tuned over the same data.

5. Based on the above points, we also find that training a biomedical domain specific model with a standard generic dataset for a domain specific task yields better results than training a generic model for the same task using the same dataset.

6. Conclusion and future work

We have proposed a method for creating Boolean queries for the purpose of document retrieval in an unsupervised way, and filtering the results from the result sets of such boolean queries in a claim-specific manner. The experiments and results show that using indicator words along with sentence embeddings, claim related sentences can be extracted from health articles in an unsupervised way. We also showed how medical/health related keywords/keyphrases can be extracted from text and then create queries from it without any supervision using Boolean operators. Our method of keyword/keyphrase extraction performed better than the available keyword/keyphrase extraction algorithms for health/medical related text. We also showed how fine-tuning domain specific transformer models leads to higher efficiency in domain specific tasks. We then show how relevant documents can be retrieved for the claims from the documents which are retrieved using the queries using a dense retrieval approach. Based on extensive experiments and statistical analysis, we show the efficiency of our approach over various baseline approaches using a publicly available dataset.

In future, we plan to use the fine-tuned PubMedBert model to extract evidence statements from the retrieved abstracts. We then plan to use these evidence statements to help refute or support the claims. We also plan to explore more datasets in this field and see how well the model generalises over different datasets in similar domains.

References

1. H. Allcott and M. Gentzkow (2017), *Social media and fake news in the 2016 election*, Journal of economic perspectives, 31(2), pp. 211-36.
2. S. Kogan, T.J. Moskowitz, and M. Niessner (2019), *Fake news: Evidence from financial markets*, Available at SSRN 3237763.
3. S. Vosoughi, D. Roy and S. Aral (2018), *The spread of true and false news online*, Science, 359(6380), pp.1146-1151.
4. K. Nagi (2018), *New social media and impact of fake news on society*, ICSSM Proceedings, July, pp.77-96.
5. V.L. Rubin, N. Conroy, Y. Chen and S. Cornwell (2016), *Fake news or truth? using satirical cues to detect potentially misleading news*, In Proceedings of the second workshop on computational approaches to deception detection, June, pp. 7-17.
6. M. Potthast, J. Kiesel, K. Reinartz, J. Bevendorff and B. Stein (2017), *A stylometric inquiry into hyperpartisan and fake news*, arXiv preprint arXiv:1702.05638.
7. Y. Liu and Y.F.B. Wu (2018), *Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks* In Thirty-second AAAI conference on artificial intelligence.
8. Y. Wang, F. Ma, Z. Jin, Y. Yuan, G. Xun, K. Jha, L. Su and J. Gao, (2018, July), *Eann: Event adversarial neural networks for multi-modal fake news detection*, In Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, pp. 849-857.
9. J. Zhang, L. Cui, Y. Fu, and F.B. Gouza (2018), *Fake news detection with deep diffusive network model*, arXiv preprint arXiv:1805.08751.
10. F. Monti, F. Frasca, D. Eynard, D. Mannion and M.M. Bronstein (2019) *Fake news detection on social media using geometric deep learning*, arXiv preprint arXiv:1902.06673.
11. R. Barbado, O. Araque and C.A. Iglesias (2019), *A framework for fake review detection in online consumer electronics retailers*, Information Processing & Management, 56(4), pp.1234-1244.
12. E. Kauffmann, J. Peral, D. Gil, A. Ferrndez, R. Sellers and H. Mora (2020), *A framework for big data analytics in commercial social networks: A case study on sentiment analysis and fake review detection for marketing decision-making*, Industrial Marketing Management, 90, pp.523-537.
13. W.Y.S. Chou, A. Oh and W.M. Klein (2018), *Addressing health-related misinformation on social media*, JAMA, 320(23), pp.2417-2418.
14. Y. Liu, K. Yu, X. Wu, L. Qing and Y. Peng (2019), *Analysis and detection of health-related misinformation on Chinese social media*, IEEE Access, 7, pp.154480-154489.
15. S. Dhoju, M. Main Uddin Rony, M. Ashad Kabir and N. Hassan (2019, May), *Differences in health news from reliable and unreliable media*, In Companion Proceedings of The 2019 World Wide Web Conference, pp. 981-987.
16. A. Joulin, E. Grave, P. Bojanowski and T. Mikolov (2016), *Bag of tricks for efficient text classification*, arXiv preprint arXiv:1607.01759.
17. S. Mukherjee, G. Weikum and C. Danescu-Niculescu-Mizil (2014, August), *People on drugs: credibility of user statements in health communities*, In Proceedings of the 20th ACM SIGKDD inter-

national conference on Knowledge discovery and data mining, pp. 65-74.

18. A. Kinsora, K. Barron, Q. Mei and V.V. Vydiswaran (2017, August), Creating a labeled dataset for medical misinformation in health forums, In 2017 IEEE International Conference on Healthcare Informatics (ICHI), pp. 456-461.
19. Y. Li, N. Du, C. Liu, Y. Xie, W. Fan, Q. Li, J. Gao and H. Sun (2017, February), *Reliable medical diagnosis from crowdsourcing: Discover trustworthy answers from non-experts*, In Proceedings of the tenth acm international conference on web search and data mining, pp. 253-261.
20. T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado and J. Dean (2013), *Distributed representations of words and phrases and their compositionality*, In Advances in neural information processing systems, pp. 3111-3119.
21. A. Ghenai and Y. Mejova (2017), *Catching Zika fever: Application of crowdsourcing and machine learning for tracking health misinformation on Twitter*, arXiv preprint arXiv:1707.03778.
22. A. Ghenai and Y. Mejova (2018), *Fake cures: user-centric modeling of health misinformation in social media*, Proceedings of the ACM on human-computer interaction, 2(CSCW), pp.1-20.
23. H. Samuel and O. Zaïane (2018, May), *MedFact: towards improving veracity of medical information in social media using applied machine learning*, In Canadian Conference on Artificial Intelligence (pp. 108-120). Springer, Cham.
24. R. Mihalcea and P. Tarau (2004, July), *Textrank: Bringing order into text*, In Proceedings of the 2004 conference on empirical methods in natural language processing, pp. 404-411.
25. K.S. Hasan and V. Ng (2014, June), *Automatic keyphrase extraction: A survey of the state of the art*, In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1262-1273.
26. I.H. Witten, G.W. Paynter, E. Frank, C. Gutwin and C.G. Nevill-Manning (2005), *Kea: Practical automated keyphrase extraction*, In Design and Usability of Digital Libraries: Case Studies in the Asia Pacific, pp. 129-152. IGI global.
27. P.D. Turney (2000), *Learning algorithms for keyphrase extraction*, Information retrieval, 2(4), pp.303-336.
28. A. Hulth, J. Karlgren, A. Jonsson, H. Boström and L. Asker (2001, February), *Automatic keyword extraction using domain knowledge*, In International Conference on Intelligent Text Processing and Computational Linguistics (pp. 472-482). Springer, Berlin, Heidelberg.
29. Y. Freund, R. Schapire and N. Abe (1999), *A short introduction to boosting*, Journal-Japanese Society For Artificial Intelligence, 14(771-780), p.1612.
30. Y.F.B. Wu, Q. Li, R.S. Bot and X. Chen (2005, October), *Domain-specific keyphrase extraction*, In Proceedings of the 14th ACM international conference on Information and knowledge management (pp. 283-284).
31. K. Sarkar, M. Nasipuri and S. Ghose (2010), *A new approach to keyphrase extraction using neural networks*, arXiv preprint arXiv:1004.3274.
32. Q. Zhang, Y. Wang, Y. Gong and X.J. Huang (2016, November), *Keyphrase extraction using deep recurrent neural networks on twitter*, In Proceedings of the 2016 conference on empirical methods in natural language processing (pp. 836-845).
33. Y. Matsuo and M. Ishizuka (2004), *Keyword extraction from a single document using word co-occurrence statistical information*, International Journal on Artificial Intelligence Tools, 13(01), pp.157-169.
34. M. Grineva, M. Grinev and D. Lizorkin (2009, April), *Extracting key terms from noisy and multitheme documents*, In Proceedings of the 18th international conference on World wide web (pp. 661-670).
35. C. Florescu and C. Caragea (2017, July), *Positionrank: An unsupervised approach to keyphrase extraction from scholarly documents*, In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 1105-1115).
36. L. Page, S.Brin, R. Motwani and T. Winograd (1999), *The PageRank citation ranking: Bringing order to the web*, Stanford InfoLab.
37. Z. Liu, P. Li, Y. Zheng and M. Sun (2009, August), *Clustering to find exemplar terms for keyphrase*

extraction, In Proceedings of the 2009 conference on empirical methods in natural language processing, pp. 257-266.

38. Z. Liu, W. Huang, Y. Zheng and M. Sun (2010, October), *Automatic keyphrase extraction via topic decomposition*, In Proceedings of the 2010 conference on empirical methods in natural language processing, pp. 366-376.
39. D.M. Blei, A.Y. Ng and M.I. Jordan (2003), *Latent dirichlet allocation*, the Journal of machine Learning research, 3, pp.993-1022.
40. A. Bougouin, F. Boudin and B. Daille (2013, October), *Topicrank: Graph-based topic ranking for keyphrase extraction*, In International joint conference on natural language processing (IJCNLP), pp. 543-551.
41. K. Bennani-Smires, C. Musat, A. Hossmann, M. Baeriswyl and M. Jaggi (2018), *Simple unsupervised keyphrase extraction using sentence embeddings*, arXiv preprint arXiv:1801.04470.
42. M. Pagliardini, P. Gupta and M. Jaggi (2017), *Unsupervised learning of sentence embeddings using compositional n-gram features*, arXiv preprint arXiv:1703.02507.
43. J.H. Lau and T. Baldwin (2016), *An empirical evaluation of doc2vec with practical insights into document embedding generation*, arXiv preprint arXiv:1607.05368.
44. H. Scells, G. Zuccon and B. Koopman (2021), *A comparison of automatic Boolean query formulation for systematic reviews*, Information Retrieval Journal, 24(1), pp.3-28.
45. J. Thorne, A. Vlachos, C. Christodoulopoulos and A. Mittal (2018), *Fever: a large-scale dataset for fact extraction and verification*, arXiv preprint arXiv:1803.05355.
46. D. Wadden, S. Lin, K. Lo, L.L. Wang, M. van Zuylen, A. Cohan and H. Hajishirzi (2020), *Fact or fiction: Verifying scientific claims*, arXiv preprint arXiv:2004.14974.
47. A. Saakyan, T. Chakrabarty and S. Muresan (2021), *COVID-fact: Fact extraction and verification of real-world claims on COVID-19 pandemic*, arXiv preprint arXiv:2106.03794.
48. M. Sarrouti, A.B. Abacha, Y. Mrabet and D. Demner-Fushman (2021, November),*Evidence-based Fact-Checking of Health-related Claims*, In Findings of the Association for Computational Linguistics: EMNLP 2021, pp. 3499-3512.
49. D. Wadden, K. Lo, L.L. Wang, A. Cohan, I. Beltagy and H. Hajishirzi (2021), *LongChecker: Improving scientific claim verification by modeling full-abstract context*, arXiv preprint arXiv:2112.01640.
50. I. Beltagy, M.E. Peters and A. Cohan (2020), *Longformer: The long-document transformer*, arXiv preprint arXiv:2004.05150.
51. R. Pradeep, X. Ma, R. Nogueira and J. Lin (2020), *Scientific claim verification with VerT5erini*, arXiv preprint arXiv:2010.11930.
52. X. Li, G.A Burns and N. Peng (2021), *A Paragraph-level Multi-task Learning Model for Scientific Fact-Verification*, In SDU@ AAAI.
53. Z. Zhang, J. Li, F. Fukumoto and Y. Ye (2021), *Abstract, Rationale, Stance: A Joint Model for Scientific Claim Verification*, arXiv preprint arXiv:2110.15116.
54. N. Reimers and I. Gurevych (2019), *Sentence-bert: Sentence embeddings using siamese bert-networks*, arXiv preprint arXiv:1908.10084.
55. S.E. Toulmin (2003), *The uses of argument*, Cambridge university press.
56. C. Stab and I. Gurevych (2017), *Parsing argumentation structures in persuasive essays*, Computational Linguistics, 43(3), pp.619-659.
57. M. Neumann, D. King, I. Beltagy and W. Ammar (2019), *Scispacy: Fast and robust models for biomedical natural language processing*, arXiv preprint arXiv:1902.07669.
58. P.W. Gwanyama (2004), *The hm-gm-am-qm inequalities*, College Mathematics Journal, pp.47-50.
59. P.S. Bullen, D.S. Mitrinovic and M. Vasic (2013), *Means and their inequalities* (Vol. 31). Springer Science & Business Media.
60. F. Liu, E. Shareghi, Z. Meng, M. Basaldella and N. Collier (2020), *Self-alignment pretraining for biomedical entity representations*, arXiv preprint arXiv:2010.11784.
61. J. Devlin, M.W. Chang, K. Lee and K. Toutanova (2018), *Bert: Pre-training of deep bidirectional transformers for language understanding*, arXiv preprint arXiv:1810.04805.

62. P.J. Rousseeuw (1987), *Silhouettes: a graphical aid to the interpretation and validation of cluster analysis*, Journal of computational and applied mathematics, 20, pp.53-65.
63. F. Fatehi, L.C. Gray and R. Wootton (2014), *How to improve your PubMed/MEDLINE searches: 3. advanced searching, MeSH and My NCBI*, Journal of telemedicine and telecare, 20(2), pp.102-112.
64. P. Bajaj, D. Campos, N. Craswell, L. Deng, J. Gao, X. Liu, R. Majumder, A. McNamara, B. Mitra, T. Nguyen and M. Rosenberg (2016), *Ms marco: A human generated machine reading comprehension dataset*, arXiv preprint arXiv:1611.09268.
65. Y. Gu, R. Tinn, H. Cheng, M. Lucas, N. Usuyama, X. Liu, T. Naumann, J. Gao and H. Poon (2021), *Domain-specific language model pretraining for biomedical natural language processing*, ACM Transactions on Computing for Healthcare (HEALTH), 3(1), pp.1-23.
66. S. Hofstätter, S. Althammer, M. Schröder, M. Sertkan and A. Hanbury, A (2020), *Improving efficient neural ranking models with cross-architecture knowledge distillation*, arXiv preprint arXiv:2010.02666.
67. T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz and J. Davison (2019), *Huggingface's transformers: State-of-the-art natural language processing*, arXiv preprint arXiv:1910.03771.
68. G. Petasis and V. Karkaletsis (2016, August), *Identifying argument components through textrank*, In Proceedings of the Third Workshop on Argument Mining (ArgMining2016) (pp. 94-102).
69. J. Pennington, R. Socher and C.D. Manning (2014, October), *Glove: Global vectors for word representation*, In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532-1543.
70. A. Conneau, D. Kiela, H. Schwenk, L. Barrault and A. Bordes (2017), *Supervised learning of universal sentence representations from natural language inference data*, arXiv preprint arXiv:1705.02364.
71. D. Cer, Y. Yang, S.Y. Kong, N. Hua, N. Limtiaco, R.S. John, N. Constant, M. Guajardo-Cspedes, S. Yuan, C. Tar and Y.H. Sung (2018), *Universal sentence encoder*, arXiv preprint arXiv:1803.11175.
72. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser and I. Polosukhin (2017), *Attention is all you need*, In Advances in neural information processing systems, pp. 5998-6008.
73. T. Achakulvisut, C. Bhagavatula, D. Acuna and K. Kording (2019), *Claim extraction in biomedical publications using deep discourse model and transfer learning*, arXiv preprint arXiv:1907.00962.
74. X. Wan and J. Xiao (2008, July), *Single Document Keyphrase Extraction Using Neighborhood Knowledge*, In AAAI (Vol. 8, pp. 855-860).
75. F. Boudin (2018), *Unsupervised keyphrase extraction with multipartite graphs*, arXiv preprint arXiv:1803.08721.
76. F. Boudin (2016, December), *Pke: an open source python-based keyphrase extraction toolkit*, In Proceedings of COLING 2016, the 26th international conference on computational linguistics: system demonstrations, pp. 69-73.
77. D.M. Powers (2020), *Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation*, arXiv preprint arXiv:2010.16061.
78. Z. Gero and J.C. Ho (2019, September), *NamedKeys: Unsupervised Keyphrase Extraction for Biomedical Documents*, In Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics, pp. 328-337.
79. G. Soğancoğlu, H. Öztürk and A. Özgür (2017), *BIOSSES: a semantic sentence similarity estimation system for the biomedical domain*, Bioinformatics, 33(14), pp.i49-i58.
80. J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C.H. So and J. Kang (2020), *BioBERT: a pre-trained biomedical language representation model for biomedical text mining*, Bioinformatics, 36(4), pp.1234-1240.
81. Y. Peng, S. Yan and Z. Lu (2019), *Transfer learning in biomedical natural language processing: an evaluation of BERT and ELMo on ten benchmarking datasets*, arXiv preprint arXiv:1906.05474.

82. S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey and N.A. Smith (2020), *Don't stop pretraining: adapt language models to domains and tasks*, arXiv preprint arXiv:2004.10964.
83. P. Lewis, M. Ott, J. Du and V. Stoyanov (2020, November), *Pretrained Language Models for Biomedical and Clinical Tasks: Understanding and Extending the State-of-the-Art*, In Proceedings of the 3rd Clinical Natural Language Processing Workshop, pp. 146-157.
84. S.R. Bowman, G. Angeli, C. Potts and C.D. Manning (2015), *A large annotated corpus for learning natural language inference*, arXiv preprint arXiv:1508.05326.
85. A. Williams, N. Nangia and S.R. Bowman (2017), *A broad-coverage challenge corpus for sentence understanding through inference*, arXiv preprint arXiv:1704.05426.
86. D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio and L. Specia (2017), *Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation*, arXiv preprint arXiv:1708.00055.
87. N. Thakur, N. Reimers, A. Rücklé, A. Srivastava and I. Gurevych (2021), *BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models*, arXiv preprint arXiv:2104.08663.
88. V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen and W.T. Yih (2020), *Dense passage retrieval for open-domain question answering*, arXiv preprint arXiv:2004.04906.
89. T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee and K. Toutanova (2019), *Natural questions: a benchmark for question answering research*, Transactions of the Association for Computational Linguistics, 7, pp.453-466.
90. M. Joshi, E. Choi, D.S. Weld and L. Zettlemoyer (2017), *Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension*, arXiv preprint arXiv:1705.03551.
91. J. Berant, A. Chou, R. Frostig and P. Liang (2013, October), *Semantic parsing on freebase from question-answer pairs*, In Proceedings of the 2013 conference on empirical methods in natural language processing, pp. 1533-1544.
92. P. Baudiš and J. Šedivỳ (2015, September), *Modeling of the question answering task in the yodaqa system*, In International Conference of the cross-language evaluation Forum for European languages (pp. 222-228). Springer, Cham.
93. L. Xiong, C. Xiong, Y. Li, K.F. Tang, J. Liu, P. Bennett, J. Ahmed and A. Overwijk (2020), *Approximate nearest neighbor negative contrastive learning for dense text retrieval*, arXiv preprint arXiv:2007.00808.
94. P. Rajpurkar, J. Zhang, K. Lopyrev and P. Liang (2016), *Squad: 100,000+ questions for machine comprehension of text*, arXiv preprint arXiv:1606.05250.
95. X. Li, G. Burns and N. Peng (2020), *A paragraph-level multi-task learning model for scientific fact-verification*, arXiv preprint arXiv:2012.14500.
96. Q. Chen, Y. Peng and Z. Lu (2019, June), *BioSentVec: creating sentence embeddings for biomedical texts*, In 2019 IEEE International Conference on Healthcare Informatics (ICHI) (pp. 1-5). IEEE.
97. M. Pagliardini, P. Gupta and M. Jaggi (2017), *Unsupervised learning of sentence embeddings using compositional n-gram features*, arXiv preprint arXiv:1703.02507.
98. T. Formal, C. Lassance, B. Piwowarski and S. Clinchant (2021), *SPLADE v2: Sparse lexical and expansion model for information retrieval*, arXiv preprint arXiv:2109.10086.
99. V. Sanh, L. Debut, J. Chaumond and T. Wolf (2019), *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*, arXiv preprint arXiv:1910.01108.
100. S.C. Lin, J.H. Yang and J. Lin (2021, August), *In-batch negatives for knowledge distillation with tightly-coupled teachers for dense retrieval*, In Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021), pp. 163-173.
101. O. Khattab and M. Zaharia (2020, July), *Colbert: Efficient and effective passage search via contextualized late interaction over bert*, In Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, pp. 39-48.
102. S. Hofstätter, S.C. Lin, J.H. Yang, J. Lin and A. Hanbury (2021), *Efficiently Teaching an Effective Dense Retriever with Balanced Topic Aware Sampling*, arXiv preprint arXiv:2104.06967.

103. S. Robertson and H. Zaragoza (2009), *The probabilistic relevance framework: BM25 and beyond*, Now Publishers Inc.
104. S.E. Robertson, S. Walker, S. Jones,M.M. Hancock-Beaulieu and M. Gatford (1995), *Okapi at TREC-3*, Nist Special Publication Sp, 109, p.109.
105. K. Clark, M.T. Luong, Q.V. Le and C.D. Manning (2020), Electra: Pre-training text encoders as discriminators rather than generators, arXiv preprint arXiv:2003.10555.
106. M. Friedman (1937), *The use of ranks to avoid the assumption of normality implicit in the analysis of variance*, Journal of the american statistical association, 32(200), pp.675-701.
107. P.B. Nemenyi (1963), *Distribution-free multiple comparisons*, Princeton University.
108. M. Henderson, R. Al-Rfou, B. Strope, Y.H. Sung, L. Lukcs, R. Guo, S. Kumar, B. Miklos and R. Kurzweil (2017), *Efficient natural language response suggestion for smart reply*, arXiv preprint arXiv:1705.00652.