DEMIR at CLEF eHealth 2019: Information Retrieval based Classification of Animal Experiments Summaries

Nizar Ahmed[1] Alirıza Arıbaş[2] and Adil Alpkoçak[3]

[123] Dokuz Eylul University, Izmir, Turkey
[1]nizar.ahmed@ceng.deu.edu.tr
[2]ali.aribas@ceng.deu.edu.tr
[3]alpkocak@ceng.deu.edu.tr

Abstract. Information retrieval searching systems recently become powerful for retrieving full text results according to a particular query (or else a document query). Elastic search is an open source information retrieval searching system that is built on Apache Lucene, and works as a distributed search and analytics engine at the same time. Therefore, this engine can also be used as one of machine learnings' approaches to solve some challenges such as document classification problem. This study is published as working-notes paper for CLEF eHealth 2019 Task 1 on Multilingual Information Extraction and it proposes a *k*-nearest neighbor (*k*-NN) and Threshold (*t*-NN) approaches to classify animal experiment summaries into its correct ICD-10 codes. After that, another two methods are proposed to control and adjust the retrieved labels of the documents results to assign ICD-10 codes for the issued query document. These approaches register high precision, recall and f-measure after we experiment it with the development dataset.

Keywords: Elasticsearch, *k*-Nearest Neighbor *k*-NN, Threshold -Nearest Neighbor *t*-NN, Multi-label classification.

1. Introduction

Information retrieval systems proved its efficacy during time by improving the correctness of the retrieved search results and minimizing the retrieval time [1]. Elasticsearch is an open source information retrieval searching system that is built on Apache Lucene, and works as a distributed search and analytics engine. This system showed its power since released in 2010 and become the most popular search model for full-text searching, log analytics, security intelligence and business analytics [2].

Using Elasticsearch is not limited only on information retrieval searching purposes but also it deals with machine learning applications. Accordingly, machine learning now becomes a core and natural extension to the search and some analytical capabilities of Elasticsearch [3]. Many researches employed Elastic search in their machine learning models but for statistical purposes (using Kabana statistical tool) such as M. Bajer [4] and J. Bai [3].

Our research is maintained according to CLEF 2019 eHealth Task 1 challenge [5]. The main requirements in their tasks is to discover the semantic indexing of NTPs using codes from the German version of the International Classification of Diseases (ICD-10). In our point of view, this task considered a multi-label classification problem since each text document is assigned/classified to at least one label of the ICD-10 codes.

This paper suggests accumulating Elasticsearch with a machine learning model to classify the text documents into one of ICD-10 codes. Our approach first suggests the work with *k*-nearest neighbor *k*-NN and a threshold approach *t*-NN for retrieving the document result set of Elastic search step. After that it proposes the work with two other approaches to predict the ICD-10 codes for the query document (such as calculating the raw or similarity frequency of the retrieved label set). To the best of our knowledge, this method considered the first to apply with a multi-label problem and well thought-out more challenging.

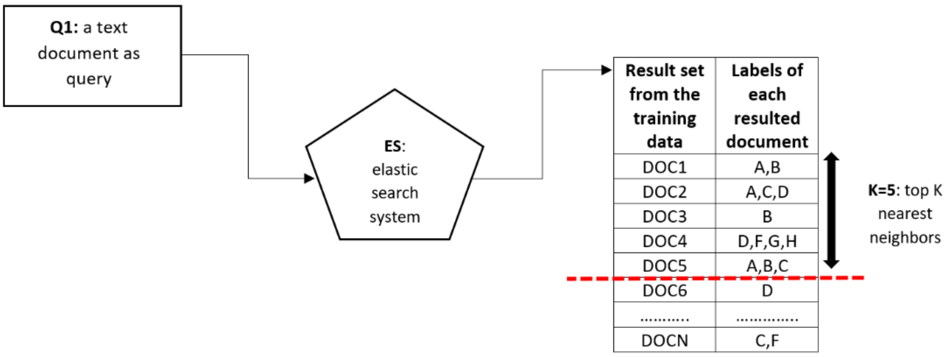
The rest of this paper is organized as follows: section two provides all the details about our methodology. Then, section three shows experimental set up. After that, section four contains our results with its discussion. Finally, section five gives the conclusion of this study.

2. Methodology

In this work, we propose an IR (Information Retrieval) mechanism for classifying the animal test information with ICD-10 codes. Animal test information is written in German language that is a non-technical summaries (NTPs) of animal experiments. This problem considered a multi-label machine learning task since each text document is assigned/classified to at least one label of the ICD-10 codes. Our methodology passes through two main phases.

2.1 The First Phase: Approaches responsible for controlling the results of the Elasticsearch IR system:

We use Elasticsearch platform to retrieve documents (from the training data) that are similar to a particular query document. The retrieved (resultant) documents may share the same class/es as the query document (for example one of the test/development files). We propose two main approaches to identify the result set of documents as a consequence of a particular query file:


Q1: a text document as query
ES: elastic search system
K=5: top K nearest neighbors

Result set from the training data	Labels of each resulted document
DOC1	A,B
DOC2	A,C,D
DOC3	B
DOC4	D,F,G,H
DOC5	A,B,C
DOC6	D
………..	…………..
DOCN	C,F

Fig. 1. *k*-NN method on the Elastic search results.

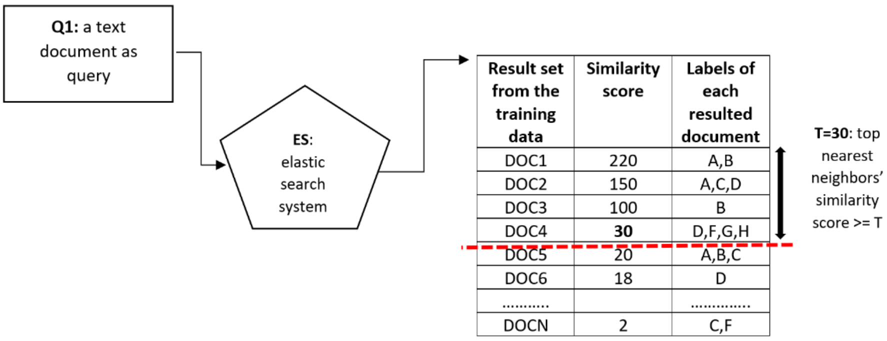
k-NN Method: (*k* nearest neighbor of the result set)

In this method we control the retrieved result set of the Elasticsearch by considering the top *k* documents which would be the nearest neighbors of the query document. We believe that changing *k* parameter will be responsible for controlling precision and recall scores. For example, imagine that we have Q1 as query document and after issuing this query in the Elasticsearch the following training data results are retrieved as follows:

Accordingly, if we set *k*=5, only the top 5 ranked documents will be considered and their label set will be taken into account while we calculate the predicted label set in the second phase.

t-NN Method: (Threshold based method)

After issuing the query document on the Elasticsearch system, the result set is retrieved with a ranking that depends on the similarity score of each document. *t*-NN method depends on controlling the retrieved result set of the Elasticsearch according to the similarity score parameter. Hence, we tune a specific similarity score (*t*: threshold value) so that any result set equal or greater than this value will be taken into account. For example, if we have *t*=30, it means that any resultant document with a similarity score equal or greater than 30 will be considered in the solution.


Q1: a text document as query
ES: elastic search system
T=30: top nearest neighbors' similarity score >= T

Result set from the training data	Similarity score	Labels of each resulted document
DOC1	220	A,B
DOC2	150	A,C,D
DOC3	100	B
DOC4	**30**	D,F,G,H
DOC5	20	A,B,C
DOC6	18	D
………..		…………..
DOCN	2	C,F

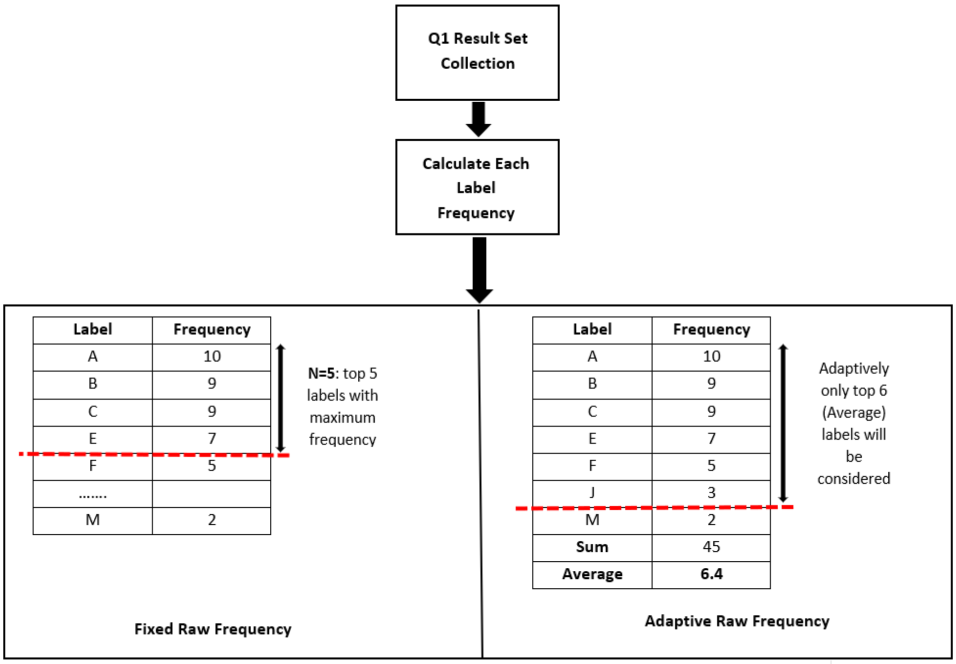
Fig. 2. *t*-NN method on the Elastic search results.

2.2 The Second Phase: Approaches responsible for predicting the class/es of the query document:

After producing the result set (i.e. retrieving the resultant documents in response to the query document), we should now calculate the majority label set of the results. We believe that these labels could be used as the predicted label of the query in hand. Moreover, taking into account the proper value of *k* or *t* in the first phase which plays an important role in label prediction process. There are two approaches in this phase:

Raw frequency of the label set:

In this approach, we calculate the frequency (normal count) of each label produced in the result set. After that we consider the top N labels with highest frequency (for example top 10 or 5 … etc.). Selecting the prober N value of the top labels controls the degree of precision and recall as well. For example, selecting N = 10 will produce a high range of predicted labels, therefore the recall score will be higher than that if we select N= 5. And vice versa for the precision score. Furthermore, another method is considered using the adaptive way (rather than the fixed one) to control the label set by averaging the sum of the label raw frequency N and use it as a cut-off value (i.e. top N will be considered).


Q1 Result Set Collection
Calculate Each Label Frequency
N=5: top 5 labels with maximum frequency
Fixed Raw Frequency
Adaptively only top 6 (Average) labels will be considered
Adaptive Raw Frequency

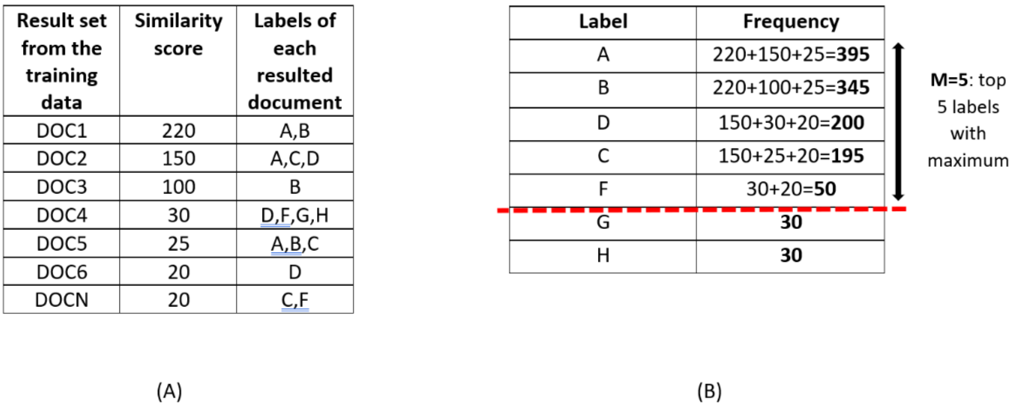
Label	Frequency
A	10
B	9
C	9
E	7
F	5
…….	
M	2

Label	Frequency
A	10
B	9
C	9
E	7
F	5
J	3
M	2
Sum	45
Average	**6.4**

Fig. 3. The raw frequency method with its two sub-approaches.

Similarity frequency of the label set:

Instead of counting the raw frequency of each label of the result set, we consider the similarity score of the resultant document to be the factor of each label. For example, if the result document (Doc.1) of the Elasticsearch has a similarity score of 20 and this document is related with the label A, B, C, then each label in Doc.1 will be related to this score 20A,20B and 20C as you can see from figure 4 A. So that when we calculate the frequency of each label from the total result set, see figure 4 B, we will consider the similarity score not the normal count of the labels. In addition, we explore the adaptive way to control the label set by averaging the sum of the label similarity frequency M and use it as a cut-off value (i.e. top M will be considered).



Result set from the training data	Similarity score	Labels of each resulted document
DOC1	220	A,B
DOC2	150	A,C,D
DOC3	100	B
DOC4	30	D,F,G,H
DOC5	25	A,B,C
DOC6	20	D
DOCN	20	C,F

(A)

Label	Frequency
A	220+150+25=**395**
B	220+100+25=**345**
D	150+30+20=**200**
C	150+25+20=**195**
F	30+20=**50**
G	**30**
H	**30**

M=5: top 5 labels with maximum

(B)

Fig. 4. Calculating label frequency according to similarity and considering the top 5 labels (fixed method)

As a result to the great difference between the values of the labels' frequencies, the adaptive way will not be effective unless we divide the average by a particular factor (several values used as parameters and taken into consideration in our case). For example, the average from the table B of figure 4 is calculated as: 390+345+200+195+50+30+30=1240/7=177.14. So, considering the top 177 labels will be exhaustive to predict the label of a particular query document. As a result, we divide the average by several values let's say 30 (a value more than 10) then: 177/30=5.9 so that only the top 6 labels will be considered as the predicted label. This approach is maintained only by experiments and it seems that it affects the progress of recall and precision very well as we will see in the result section.

In general, all the approaches mentioned in this section were preserved, explored and proved in an experimental environment. Eventually, they seem satisfying after we see the recall and precision scores getting higher with each parameter taken into consideration.

3. Experimental Setup

A total Number of 8793 German text documents are used in the experiments: 7543 training, 842 development set and 403 as testing set. All of the training and development set are annotated by either one of the ICD-10 codes or without label. On the other hand, we don't reward or penalizing for unannotated NTSs. Moreover, all the test set document released in CLEF 2019 eHealth Task 1 challenge without providing their gold-truth.

Our experiments started by preparing the training documents to represent the collection of corpora that will be retrieved as the search results of a query document. And the development/test sets are used as the query documents that we need to predict their ICD-10 labels. Therefore, the experiments begin from phase one by issuing a query document as an input to the Elasticsearch platform. Then a result set returned consequently with their related labels and some other information such as similarity scores for each document. We used Elasticsearch default settings for similarity, analyzer, stemmer and stopwords in German. Elasticsearch weighting schemas consist of term frequency, inverse document frequency and field-length norm for calculating similarity score [8]. The following points describe all the experiments that held by our system and each considers one approach of phase one with another in phase two in sequence (like described previously in methodology section).

3.1 Experiments representing *k*-NN method from phase one and both methods of the second phase:

Experiment 1: *k*-NN with top N of the *fixed* raw frequency approach to predict the label set.

In this test we explore seven values of *k* (*k*= 15, 20, 50, 100, 200, 500 and 1000) to consider only the top *k* resulted documents. Those values are used to tune *k* parameter and record precision and recall scores to see which value will be the most suitable one. After retrieving the resulting documents , we conduct the raw frequency method of phase two to predict the label set. We explored several values to consider the top N labels as the predicted class: N = 10 to 2 (9 values).

Experiment 2: *k*-NN with top N of the *adaptive* raw frequency approach to predict the label set.

In this experiment we try thirteen values of *k* (*k*=5,6,7,8,9,10,15, 20, 50, 100, 200, 500 and 1000) to consider only the top *k* resulted documents. Then we work with the adaptive approach that will choose the proper value of the top raw label frequency as described in the methodology section.

Experiment 3: *k*-NN with top M *fixed* similarity frequency of the label set.

In this test we try eight values of *k* (*k*=5,6,7,8,9,15, 20 and 50) to consider only the top *k* resulted documents. After retrieving the resulting documents, we conduct the similarity frequency method of phase two to predict the label set. We explored several values to consider the top M labels as the predicted class: M = 10 to 2 (9 values).

Experiment 4: *k*-NN with top M of the *adaptive* similarity frequency approach to predict the label set.

In this experiment we try nine values of *k* (*k*= 5,6,7,8,9,10,15, 20 and 50) to consider only the top *k* resulted documents. Then we work with the adaptive approach that will choose the proper value of the top raw label frequency as described in the methodology section.

3.2 Experiments represent *t*-NN method from phase one and both methods of the second phase:

Experiment 5: *t*-NN with top N of the *fixed* raw frequency approach to predict the label set.

In this approach we choose to work with three threshold values of *t* (*t*= 10, 20 and 30) to take the top resultant documents which its similarity is greater or equal to *t*. After that we apply the raw frequency method of phase two to predict the label set. We test several values to select the top N labels as the predicted class: N = 10 to 2 (9 values).

Experiment 6: *t*-NN with top N of the *adaptive* raw frequency approach to predict the label set.

Likewise, we select four values of the threshold *t* (*t*=10,20,25 and 30). Then we work with the adaptive approach that will choose the proper value of the top raw label frequency.

Experiment 7: *t*-NN with top M *fixed* similarity frequency of the label set.

Similar to experiment 5, we select three threshold values (*t*=10,20 and 30) for phase one and the similarity frequency method of phase two to predict the label set (top N labels as the predicted class: N = 10 to 2).

Experiment 8: *t*-NN with top M of the *adaptive* similarity frequency approach to predict the label set.

Finally, we choose eight threshold values (T= 10,20,30,40,50,60,70 and 80) in phase one, and the adaptive approach that will select the proper value of the top similarity label frequency in phase two.

4. Results and Discussion

For evaluating our approach, we depend on three state of art evaluation metrics: precision, recall and F-measure [6] [7]. The following tables summarize the results of all the experiments applied on the development dataset within the two main phases (i.e. *k*-NN and *t*-NN), with each point of phase two that explains each method for predicting the labels. These tables hold only the best values of precision, recall and F-measure scores in each experiment mentioned in the last section.

Table 1. Evaluation of all *k*-NN experiments with raw and similarity frequency label prediction techniques using development set.

Experiment No and Description	*Exp. 1: k*-NN with *fixed* raw frequency	*Exp. 2: k*-NN with *adaptive* raw frequency	*Exp. 3: k*-NN with *fixed* similarity frequency	*Exp. 4: k*-NN with of the *adaptive* similarity frequency
k and top N/M labels values	*k*=15, N=2	*k*=8, N=adaptive	*k*=5, M=2	*k*=5, M=adaptive
Precision	0.562	0.549	**0.808**	0.672
Recall	0.503	0.545	0.707	**0.817**
F-measure	0.531	0.547	**0.754**	0.738

**k*: stands for *k* nearest neighbors of Elasticsearch result set. *N: Top N raw frequency. *M top M similarity frequency.

Table 2. Evaluation of all *t*-NN experiments with raw and similarity frequency label prediction techniques using development set.

Experiment No and Description	*Exp. 5: t*-NN with fixed raw frequency	*Exp. 6: t*-NN with adaptive raw frequency	*Exp. 7: t*-NN with fixed similarity frequency	*Exp. 8: t*-NN with of the adaptive similarity frequency
t and top N/M labels values	t= 20, N = 2	t= 25, N = adaptive	t= 30, M = 3	t= 80, M = adaptive
Precision	0.558	0.541	0.722	**0.843**
Recall	0.479	0.420	0.816	**0.838**
F-measure	0.516	0.473	0.767	**0.841**

**t*: stands for the threshold value according to similarity score value of the retrieved result set.

We noticed that working with similarity frequency for predicting the label set outperforms considering the raw frequency with both *k*-NN and *t*-NN main approaches. More specifically, as you can see from experiment 8 of table 2 (i.e. *t*-NN and adaptive similarity frequency for label prediction techniques), a highest precision, recall and F-measure has been recorded. Furthermore, this score demonstrates that setting *t* = 80 and working with the adaptive way for predicting the labels using similarity frequency will guarantee that the query will neither be more specific nor more exhaustive. Else, at *t*=80, precision, recall and f-measure are meeting at this point so that they are more moderate and stable. See figure 5.

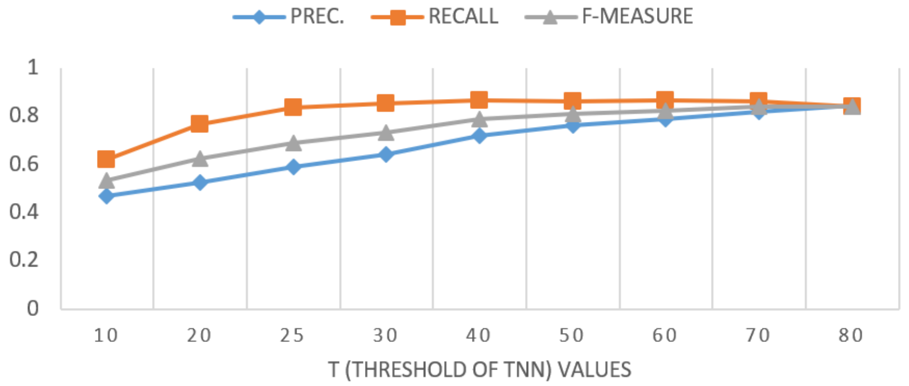


Fig. 5. meeting point of precision, recall and F-measure at *t*=80 with *t*-NN adaptive similarity frequency

Finally, we choose three methods to apply them on the testing data: Exp3, Exp7 and Exp8. We got these results with the best evaluation scores in the development set amongst the others (As you can see from the bold text in Table 1 and Table 2). The following Table 3 shows the precision, recall and F-measure scores after we run our system on the test data.

Table 3. Evaluation of raw and similarity (weighted) frequency label prediction techniques using test set.

Experiment No and Description	*Exp. 3:* *k*-NN with *fixed* similarity frequency	*Exp. 7:* *t*-NN with *fixed* similarity frequency	*Exp. 8:* *t*-NN with of the *adaptive* similarity frequency
k, *t* and top N/M labels values	*k*=5, N=3	*t*=10, M=3	*t*=30, M=adaptive
Precision	0.46	**0.49**	0.46
Recall	**0.50**	0.44	**0.49**
F-measure	0.48	0.46	0.48

As a result, we compared our results with development and test set. Our system works better with larger set, since development includes more documents than test set. The more number of the retrieved result set, the more ICD-10 codes return. Some query returns too much, some too few. The parameters, *t* and *k*, controls the size of result set. We run the test set queries using parameters shown in Table 3. But this settings didn't work well in some conditions. For example, when threshold $t = 80$, more than 40% of test set documents retrieved no results. When we decrease the value of *t* from 80 to 10 and then 30, it returns document with lower similarity score and hence it may lead to misclassification.

5. Conclusion

CLEF 2019 eHealth Task 1 announced a challenge that is concerned with discovering the semantic indexing of NTPs using codes from the German version of the International Classification of Diseases (ICD-10). This task considered a multi-label classification problem since each text document is assigned/classified to at least one label of the ICD-10 codes. This paper proposes an information retrieval paradigm that depends on Elasticsearch result set to classify unseen (query documents) to its correct ICD-10 code. There are two main phases proposed in this study: the first one responsible for controlling the results of the Elasticsearch IR system (depending on *k*-NN and *t*-NN approaches) and the second responsible for predicting the class/es of the query document (depending on fixed or adaptive raw frequency as well as similarity frequency). Our results show that working with *t*-NN approach in phase one and the adaptive similarity frequency in phase two records the highest precision, recall and f-measure

References

[1] M. S. Divya and S. K. Goyal, "ElasticSearch: An advanced and quick search technique to handle voluminous data.," *COMPUSOFT, An international journal of advanced computer technology,* p. 171, 2013.

[2] C. Gormley and Z. Tong, Elasticsearch: The definitive guide: A distributed real-time search and analytics engine., O'Reilly Media, Inc., 2015.

[3] J. Bai, "Feasibility analysis of big log data real time search based on hbase and elasticsearch.," in *In 2013 ninth international conference on natural computation (ICNC),* 2013.

[4] M. Bajer, "Building an IoT data hub with Elasticsearch, Logstash and Kibana.," in *In 2017 5th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW),* 2017.

[5] L. Kelly, . H. Suominen, . L. Goeuriot , M. Neves , E. Kanoulas , D. Li , L. Azzopardi , R. Spijker , G. Zuccon , H. Scells and J. Palotti, "Overview of the {CLEF eHealth} Evaluation Lab 2019," in *Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Tenth*

International Conference of the CLEF Association (CLEF 2019)., Berlin Heidelberg, Germany, 2019.

[6] . J. Read, P. Bernhard and H. Geoffrey , "Multi-label classification using ensembles of pruned sets.," in *In 8th IEEE international conference on data mining*, 2008.

[7] . I. Triguero and V. Celine , "Labelling strategies for hierarchical multi-label classification techniques.," *Pattern Recognition,* pp. 170-183, 2016.

[8] Gormley, C., & Tong, Z. (2015). Elasticsearch: The definitive guide: A distributed real-time search and analytics engine. " O'Reilly Media, Inc.".