Medical Domain Document Classification via Extraction of Taxonomy Concepts from MeSH Ontology

Mihailo Škorić, Mauro Dragoni

Medical Domain Document Classification via Extraction of Taxonomy Concepts from MeSH Ontology | Mihailo Škorić, Mauro Dragoni | Infotheca | 2019 | |

10.18485/infotheca.2019.19.1.3

http://dr.rgf.bg.ac.rs/s/repo/item/0004946

Medical domain document classification via extraction of taxonomy concepts from MeSH ontology

UDC 004.82:025.43MESH
DOI 10.18485/infotheca.2019.19.1.3

Mihailo Škorić
mihailo.skoric@rgf.bg.ac.rs
University of Belgrade
Belgrade, Serbia

Mauro Dragoni
dragoni@fbk.eu
Fondazione Bruno Kessler
Trento, Italy

ABSTRACT: This paper is a result of a task presented to attendants of *Keyword Search in Big Linked Data* summer school, that was organized by Vienna University of Technology, under the *Keystone COST* action in the summer of 2017. It presents a specific approach to the classification via creation of minimal document surrogates based on the US National medical library's MeSH ontology, which is derived from the Medical Subject Headings thesaurus. In a series of previously classified medically related texts, which are the bases for the task, all of the significant terms are located and replaced with taxonomical references from the MeSH ontology. Extracted references are used for the classification within the ontology using a rather simple algorithm and the results are evaluated in compresence to previous manual classification of the same documents.

KEYWORDS: document classification, MeSH, ontology, information extraction.

PAPER SUBMITTED: 21 April 2019
PAPER ACCEPTED: 30 August 2019

1 Introduction

1.1 About the task

This paper describes an attempted solution of an assign given during a one-day hackathon by the lecturers of the summer school Keystone 3^{rd} training school: *Keyword Search in Big Linked Data*,[1] organized in Vienna from 21–25 August, 2017 under the COST action IC1302 Keyword search in Big Linked Data. The task was to classify 10.000 given documents originating from the digital collection of the US National medical library (Figure 1), whereas the classification to be used was predefined in the MeSH (Medical Subject Headings) ontology (Dragoni, 2017).

> *...Goserelin in the adjuvant treatment of breast cancer An update of the Zoladex Early Breast Cancer Research Association (ZEBRA) trial was presented by Professor R Blamey (Nottingham City Hospital, UK). Goserelin was found to be better ... Results were presented by the Austrian Breast and Colorectal Cancer Study Group comparing ...*

Figure 1. A fragment of one of the documents to be classified.

The classification presented in this paper was done according to the medical subjects from the MeSH ontology, version for 2016.[2] Ontology can be queried via web,[3] where you can get predefined queries, the ones for classes and predicates being aong them, or where other data can be obtained with the new SPARQL queries.

The documentation, RDF triplets and the case example download are available online,[4] and there is also an option for previewing the predicate via access point, where the predicates can be seen in tabular form, as well as their descriptions and XML labels, which is especially important if a local copy of the MeSH ontology is used[5] Ontology consists of 56,309 medical concepts, described and systematically classified in a hierarchical tree (Figure 2).

The concepts from the ontology are hierarchically collated, and each has an assigned identifier consisting of blocks of digits, separated by colons, that describe the parent concepts in descending order, from highest to lowest in the hierarchy. In this case, the classification classes relate to the second level

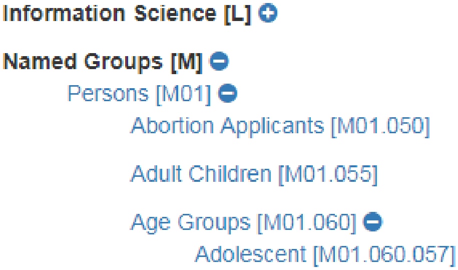
[1] Big Linked Data (on-line)

[2] The classification of documents from a medical domain based on ontologies is the subject of research by a number of teams, using different approaches, but as an ontology, MeSH is most often used.

[3] Access point (on-line)

[4] RDF triplets (on-line)

[5] MeSH ontology (on-line)



Information Science [L] ⊕

Named Groups [M] ⊖

Persons [M01] ⊖

Abortion Applicants [M01.050]

Adult Children [M01.055]

Age Groups [M01.060] ⊖

Adolescent [M01.060.057]

Figure 2. View of a hierarchical tree cut.

of the tree hierarchy – there are a total of 1,718 - and are recognized by two-block identifiers, for example [M01.055] Adult children, where the first block – M01 indicates that it has a parent node [M01] Persons, and the second block 055 is unique among the sibling nodes (Figure 2).

1.2 Classification introduction

The problem of classifying documents in general occurs in two variants: classification in a well-known, restricted domain of classes, and in an unknown one. For both, the problem is reduced to calculating the similarity of documents, usually with use of the so-called Dice[6] index or coefficient.

$$sim(D_i, D_j) = \frac{2|S_i \bigcap S_j|}{|S_i| + |S_J|} \tag{1}$$

Dice's equation tells us that if S_i is a set of terms from document D_i, and S_j is a set of terms from document D_j, then this index can be defined as a double the number of common terms divided with the total number of terms in both documents (if S is a set, $|S|$ is a number of terms in that set). If documents do not have any common terms then $sim(D_i, D_j)$ is equal to 0 which reflects the minimal similarity of the two documents, and if two documents have exactly the same set of terms assigned then $sim(D_i, D_j)$ is equal to 1 which reflects maximum similarity. When a domain of classes is known and limited, the problem is reduced to finding the appropriate class

[6] Lee Raymond Dice – American biologist (1887-1977)

with the largest sim(document, document class) value, i.e. the class most similar to the document that is the subject of the classification.[7]

The problem that arises in calculating the coefficient of similarity between texts is high computer cost, which must be paid either in processing power or high execution time. For this reason, the first step in classifying (and indexing) is most often the creation of a document surrogate. Usually, documents are translated into the word-vector space, or a frequency index. Sometimes words in the index are further derived using stemming or lemmatization, and sometimes by replacing synonyms or hypernyms, in order to further reduce the surrogates and speed up the execution time. For a qualitative classification process, it is necessary to create a surrogate which properly represents the document.

(Trieschnigg et al., 2009) and (Elberrichi et al., 2012) tested a few methods based on MeSH classification based on either MeSH ontology or thesaurus. Created classifiers used:

- MeSH Thesaurus only ('Thesaurus-oriented' classifiers);
- Training set to build explicit models for each MeSH concept ('Concept-oriented' classifiers);
- Manually created document annotations, like ordinary text classifiers, to determine the appropriate concept ('K-Nearest Neighbor' classifier);
- Hybrid and hand-refined systems that combine multiple approaches – 'Hybrid' classifiers.

In both papers, it was concluded that the K-Nearest Neighbor Classifier (KNN) produces the best results, but despite its advantages, it is significantly slower than the thesaurus-based classifiers, and with the growth of a set of test documents, its performance further decreases, which was not desirable in solving our task.

In this paper, we experiment with a simple classification approach to evaluate the importance of timely management of large amounts of data, as well as the usable value of semantics stored in the MeSH ontology. The goal was to create a classifier that would be quick and simple, in order to solve the problem of the large amount of text that needed to be classified. A drastic summarization of documents and the classes themselves was applied. Classes (concepts of the second level of ontology) were reduced to a single term – their name. On the other hand, the documents were reduced only to the occurrences of terms (concept names from MeSH ontology) that, with

[7] IR notes (on-line)

the simple mapping (stored in their identifiers in the ontology itself), are identified with a term that denotes a class, their parent object. This greatly facilitates and speeds up similarity calculations, as each class now has only one term. In this way, the document will always be classified by the Dice index into the class whose (only) term occurs most often in it, thus avoiding a large amount of computation and reducing the task to finding the most frequent term in the surrogate of the text.

2 Experiment setting

The aim of the experiment was to test the possibility and success of classification of medical documents based on taxonomy from the MeSH ontology and a rule-based system managing the appearance of terms related to concepts from the MeSH ontology in the documents to be classified. The course of the experiment can be divided into five follow-up steps.

1. **Extraction of taxonomy from MeSH ontology using SPARQL query**. This stage was necessary in order to snap together the list of identifiers and determine the taxonomic position of the concepts used in the documents, as well as the relative position of their nodes in the hierarchy.
2. **The conversion of documents into vectors of identifiers using concepts occurring within them**. This stage allows the assignment of attributes that are directly and inextricably linked to the classes in which these documents should be classified.
3. **Noise removal**. This stage should enable and provide better results for the document classification.
4. **Document classification based on their identifier vectors and a simple set of rules**.
5. **Evaluation of document classification performance for each of the sets used**. This stage allows us to reflect on and compare different classification rules, as well as to determine whether some rulesets can (and to what extent) be considered successful.

In the following chapters, these experiment stages will be described in more detail, in order to get a better insight into the methods used and the results obtained.

2.1 Extraction of taxonomy of concepts from MeSH ontology

Extracting the matrix of the concept names and their identifiers in the classification tree is done using another SPARQL query. Since in this ontology there are triples consisting of the concept, predicate and object of that predicate, which reflects the position in taxonomy, this part of the task is reduced to the extraction of a subject and object for each of these triplets.

First, it was necessary to find the name of the predicate that reflects the position in the taxonomy in the form **[A-Z][0-9][0-9]([.][0-9][0-9][0-9])***.[8] A simple SPARQL query was used, with one ontology concept (mesh2016: D049916) inputted, and it lists all predicates and objects of the MeSH 2016 ontology triple, whose concept is a part of (Figure 3). The query is illustrated on the concept *mesh2016:D049916.*

```
PREFIX mesh2016: <http://id.nlm.nih.gov/mesh/2016/>
SELECT DISTINCT ?predikat ?objekat
FROM <http://id.nlm.nih.gov/mesh/2016>
WHERE  mesh2016:D049916 ?predikat ?objekat
ORDER BY ?class
```

Figure 3. SPARQL query used to provide taxonomy of all concepts and MeSH 2016 ontologies.

Based on the query, a set of results is obtained containing, inter alia, the output from which it is concluded that the required predicate is *meshv:treeNumber*because it contains the syllables describing the hierarchy (Figure 4). The data is used in a subsequent query that aims to derive all the names of the concepts and their *meshv:treeNumber* values.

The concept names are derived from the *rdfs:label*, followed by the *mesh:treeNumber* of the same concept. The returned concepts are sorted by the size of the name from the longest to the shortest, in order for them to be searched in the documents without the risk of longest matches not being recognized due to previous recognition of shorter ones (Figure 5).

The result of this query is a CSV file whose rows contain the name (*rdfs:label*) and the taxonomic reference (*treeNumber*) of each concept (Fig-

[8] This regular expression describes a construction that consists of a mandatory part (capital letter, number, digit) and an optional part (dot, number, number, number) that iterates.

```
rdf:type; meshv:TopicalDescriptor
rdfs:label; Polyplacophora
meshv:identifier; D049916
meshv:dateEstablished; 2006-01-06
meshv:historyNote; 2006
meshv:publicMeSHNote; 2006
meshv:treeNumber; mesh2016:B01.050.500.644.600
```

Figure 4. Some of the SPARQL Queries 1 outputs, among which are the desired predicate and object

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX meshv: <http://id.nlm.nih.gov/mesh/vocab#>
PREFIX mesh2016: <http://id.nlm.nih.gov/mesh/2016/>
SELECT DISTINCT ?naziv ?treeNumber
FROM <http://id.nlm.nih.gov/mesh/2016>
WHERE ?koncept rdfs:label ?naziv .
?koncept meshv:treeNumber ?treeNumber
ORDER BY DESC(STRLEN(?naziv))
```

Figure 5. SPARQL query used for provision of a list of the name and position of all the concepts from the MeSH 2016 ontology.

ure 6). This file will be used in the next step, where the concept names are located in the documents and replaced with the node identifiers from the taxonomic tree. It should be noted that both one-word and multi-word units (e.g. Gram-Negative Bacteria) can be found. However, having in mind the order of applying the replacements (from the longest to the shortest term), there will be no wrongful replacement and recognition of only a part of the term.

2.2 Conversion of documents into concept vectors

This stage consists of two steps. First, in all documents, the previously listed concepts are found and replaced with corresponding identifiers, and then the remaining text is removed in order to transform the documents solely into the list of the identifier vector. No normalization of the document or concepts was done, which is sustainable for English which does not have a rich flexible

```
Ganglia;A08.340
Neurons;A08.675
Malleus;A09.246.397.247.524
Cochlea;A09.246.631.246
Eyelids;A09.371.337
Choroid;A09.371.894.223
Tissues;A10
Chorion;A10.615.284.473
Muscles;A10.690
```

Figure 6. Examples of lines from CSV document containing the names and identifiers of concept nodes.

system, but for a morphologically rich language, such as Serbian, previous lemmatization or other kind of normalization of both resources is necessary.

Finding and replacing ontology concepts in the text. As we wanted to find something in the documents (names of the concepts) and then replace it with something else (corresponding taxonomic identifiers), having those two things listed together in a previously generated file, it was possible to directly transform the list from the file directly to C# function that would do it.

This is achieved by through transformation of the CSV file. Character **;** was replaced with string **", "** and strings **doc = doc.Replace("** and **");** were pasted onto the beginning and the end of each row respectively (Figure 7).[9]

For replacement in all the documents to be classified, a second C# code has been prepared. It loads the classification documents one at a time and applies the script generated in the previous step so that the concepts are found by name and replaced by an ontology node identifier. This stage is the longest and the most time consuming because our experiment involves the application of 56,309 term replacements over 10,000 documents, giving a total of 563,090,000 transformations. Further research can go towards using

[9] In hindsight, a similar approach can also generate a different type of substitution based on loops or regular expressions, which would speed up replacements and reduce the effects of multiple parsings of the same document.

```
doc = doc.Replace("Ganglia", "A08.340");
doc = doc.Replace("Neurons", "A08.675");
doc = doc.Replace("Malleus", "A09.246.397.247.524");
doc = doc.Replace("Cochlea", "A09.246.631.246");
doc = doc.Replace("Eyelids", "A09.371.337");
doc = doc.Replace("Choroid", "A09.371.894.223");
doc = doc.Replace("Tissues", "A10");
doc = doc.Replace("Chorion", "A10.615.284.473");
doc = doc.Replace("Muscles", "A10.690");
```

Figure 7. The section of a find and replace script based on the previously generated CSV file (Figure 6).

finite state machines and transducers to solve this problem, which is more complex to implement but performs faster in processing this type.

Transformation of documents into concept vectors (surrogate creation) After the documents were successfully annotated with the identifiers of concepts that appeared in them, it was necessary to clear the documents from the rest of the unpaired text. To prevent this from working individually for each document, they are merged into one, ~230MB, in size, with new lines as the border between the documents. Information of importance – document names (**[0-9]+[.]txt**), identifiers in them (**[A-Z][0-9][0-9]([.][0-9][0-9][0-9])***) and tags for new row (**[\r\n]+**) - are found using regular expression (**[A-Z][0-9][0-9]([.][0-9][0-9][0-9])***)|(**[0-9]+[.]txt**)|(**[\r\n]+**), док се све остало уклања.while everything else is removed. This reduced the file size over 450 times (new size: ~0.5MB).

Upon completion of the transformation, a new file is formed. In it, each new line represents a new document: it begins with the title of the document (without extension) followed by a semicolon, and all the concept identifiers found in it separated by commas (Figure 8).

We will illustrate the transformation of one of the starting documents by steps on a simple example. In a document fragment from 1, some terms were identified (Figure 9), and then replaced by identifiers (Figure 10). It is noted that in the *Colorectal*, part of the word was recognized, and a string **Color** was mistakenly replaced by **G01.590.540.199**.[10] This happened because

[10] Such an error could have been avoided by previous tokenization of the text.

```
2875592;M01.060.116
2875593;D13.444.308,D13.444.308
2875594;C04.557.465.625.650.510,D13.444.735,D13.444.735
2875595;A01.236,A01.236;A01.236
2875596;D13.444.735
2875598;
2875599;D02.455.612
```

Figure 8. A fragment of a file that contains the names of documents and identifiers in them.

neither the terms *colorectal cancer* nor *colorectal* are found as terms in the ontology version used. Figure 11 shows the final surrogate of the text from Figure 1.

> ... **Goserelin** in the adjuvant treatment of breast cancer An update of the Zoladex Early **Breast** Cancer **Research Association** (ZEBRA) trial was presented by Professor R Blamey (Nottingham City Hospital, UK). **Goserelin** was found to be better ... Results were presented by the **Austrian Breast** and **Color**ectal Cancer Study Group comparing ...

Figure 9. A fragment of the original document with the concepts found in MeSH ontology marked.

> ... **D06.472.699.327.740.320.340** in the adjuvant treatment of breast cancer An update of the Zoladex Early **A01.236** Cancer **H01.770.644 F02.463.425.069** (ZEBRA) trial was presented by Professor R Blamey (Nottingham City Hospital, UK). **D06.472.699.327.740.320.340** was found to be better... Results were presented by the **Z01.542.088 A01.236** and **G01.590.540.199**ectal Cancer Study Group comparing ...

Figure 10. A fragment of the original document with the concepts found in MeSH ontology replaced with identifiers.

...D06.472.699.327.740.320.340;A01.236;H01.770.644;F02.463.425.
069;D06.472.699.327.740.320.340;A01.236;G01.590.540.199;...

Figure 11. Final surrogate of a document fragment.

2.3 Noise removal

Before moving onto classification, it was necessary to detect possible noise in the form of identifier attribution errors, highly frequent concepts or the ones that appear in an excessive number of documents, which are therefore not discriminatory. For each class, a total number of repetitions was calculated, resulting in an uneven distribution (Figure 12).

The first identifiers added to the list of stop words and removed are those relating to geographic locations (listed in ontology under class Z01 - geographic locations as well as homonyms like the term back, which appears in documents 11,440 times, apparently not always to denote parts of the human body. Figure 13, however, shows the uneven distribution of frequency classes even after this removal.

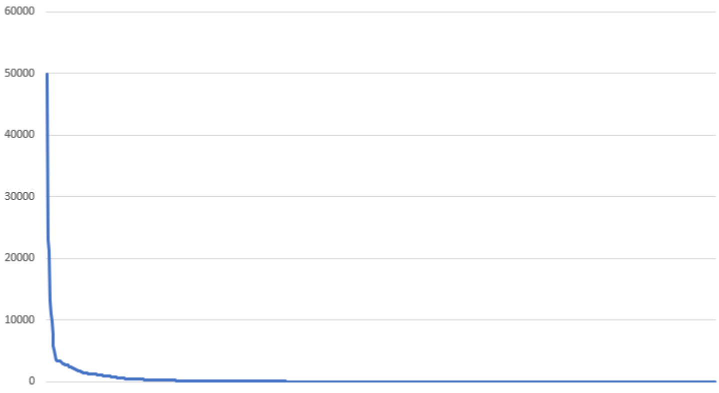


Figure 12. Frequencies distribution before noise removal.

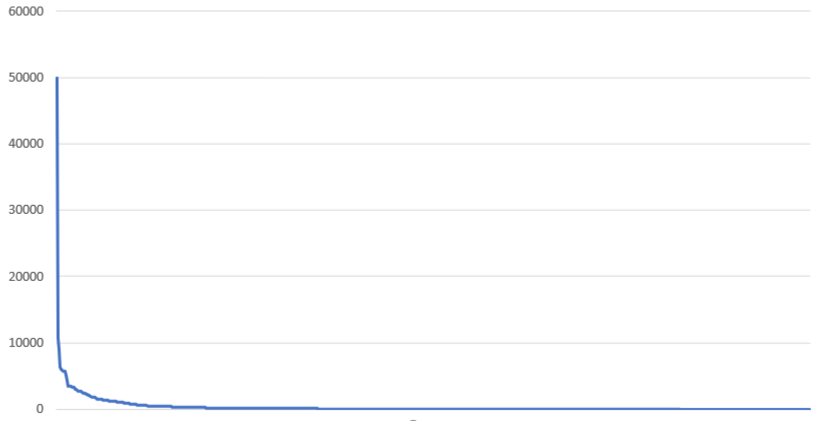


Figure 13. Frequencies distribution after noise removal.

2.4 Classification of documents using identifiers

Two processing procedures were applied to the documents subject to classification, resulting in two test sets. The control method was to replace the identifiers with their class, a more general hierarchical designation. The experimental method also considered the length of the identifiers, so they were replaced by a certain number of iterations of the parent class depending on their length, to test the assumption that the use of more specialized terms was more important for determining the class of documents. So, instead of reducing the identifiers to the first two blocks of digits, their length was taken into account or the depth of each of the concepts in the tree. For example: identifier **D04.345.295.750.650.700** has been replaced using an appropriate regular expression with **D04.345, D04.345**, which is equivalent to the appearance of two concepts belonging to class **D04.345** in that document. The way to map the length of concepts into the corresponding number of repetitions is given in Table 1.The table shows that terms are identified with a maximum of four repetitions (if they have more than eight blocks) of a term denoting a class. After applying these steps, only class identifiers now appear in document surrogates, which should be easily counted.

After the test sets have been successfully created, a simple program is prepared for document classification, which requires a file with inputs indicating the classes (the first two blocks of digits) that are recognized as

number of surplus blocks (over 2)	0	1	2	3	4	5	6	7	8	9	10
number of resulting class iterations	1	1	2	2	2	3	3	3	4	4	4

Table 1. Mapping the number of surplus syllables and the number of resulting class identifiers.

input. The program simply counts the classes that occur in the surrogate of the document and returns the one that occurs most frequently. If there are multiple classes that occur in the document with the same frequency, the class that first appears returns, which is logical because the order of occurrence of the terms is retained in the surrogates. Also, it is necessary for each document to be assigned an identifier, so if a document does not have an identifier assigned, it is assigned one of the most general – **H02.403** – which designates medicine.

When a sequence of identifiers in each document is reduced to one class, it is taken as a result of the classification and forwarded for evaluation.

3 Evaluation and results comparison

Test set	Taking identifier length into account	Precision @10	Average mean Precision	Recall	F-measure
1	yes	0.58	0.0060	0.0696	0.0108
2	no	0.46	0.0057	0.0648	0.0103

Table 2. Test sets used for classification and their results.

Experiments were performed on documents from the TREC Clinical Decision Support 2016 set.[11] The aim was to classify documents based on terms that denote concepts in the MESH ontology and that appear in those documents. The annotation used in the evaluation was manually conducted by an expert team. The usual metrics of average mean precision, recal[12] and F-measure

[11] TREC Clinical Decision Support 2016 (on-line)

[12] Relevant documents not included in the ranking were taken as false negatives.

were applied, as well as the precision@10 metric, which reflects the success of returning relevant documents in the first 10 results of a query.

Table 2 shows us that taking the length of the identifier into account yielded a slight improvement in results (5% improved precision and F-measure, 7% improved recall and 12% improved precision@10).

Considering the values obtained, it can be immediately noticed that the precision is unusually low relative to recall. By applying a more detailed analysis of the data, a very large number of false positives was observed, thus explaining the decreased precision of the given strategy. This result is not surprising, as it concurs with current standards in the field of classification of medical records (Calí et al., 2017). A major problem with concept-oriented information retrieval in the biomedical sphere is the large number of miss-classified documents, leading to a very low response rate. Low precision is thus acceptable in this paper because it is offset by a higher response rate and many relevant documents are returned in the highest positions, with precision@10 values, as high as 58%. Still, there is room for progress here.

4 Conclusion

In this paper we presented an approach to document classification which is based on the creation of the minimal surrogates of those documents. Within medical documents, specific terms are located and replaced with taxonomical references. Extracted references are used for classification using MeSH ontology and a simple algorithm and evaluated against a team of experts.

Preliminary results demonstrated the suitability of the proposed approach within a very complex task. Future work will focus on the decrease of false positive results in order to boost the overall performance of the system.

The classification based on ontologies does not depend on the domain in which it is applied, but it certainly depends on the resources available, specifically the ontology or taxonomy used for the classification (Rakesh et al., 2001).Once established, the system may find wider application. When it comes to the classification of (medical) documents for the Serbian language, it is necessary to prepare resources first. In this regard the International Classification of Diseases in Serbian - *MKB 10* (*Međunarodna klasifikacija bolesti*) (Kolonja et al., 2016) could certainly be of use, where a number of terms is associated with English and Latin equivalents, allowing for the extension of the search for concept names and their retrieval in documents. However, rich Serbian language morphology should be taken into account and preparation of additional lexical resources specific to the field of medicine would be required in order to normalize text before classification or indexing, which would help to identify more taxonomic terms in documents (Stanković et al., 2015).

5 Acknowledgements

This work was created within the Keyword Search in Big Linked Data summer school, organized as part of the Keystone COST action from 21 to 25 August 2017 at the University of Technology in Vienna.

References

Rakesh et al., Agrawal. “Multilevel taxonomy based on features derived from training documents classification using fisher values as discrimination values”, U.S. Patent No. 6,233,575, 2001

Calí, Andrea, Dorian Gorgan and Martin Ugarte. *Semantic Keyword-Based Search on Structured Data Sources: COST Action IC1302 Second International KEYSTONE Conference, IKC 2016, Cluj-Napoca, Romania, September 8–9, 2016, Revised Selected Papers*, Vol. 10151, 2017

Dragoni, Mauro. “3rd KEYSTONE Summer School”, 2017, URL `http://ifs.tuwien.ac.at/keystone.school/slides/Dragoni_SemanticSearch.pptx`

Elberrichi, Zakaria, Malika Taibi and Amel Belaggoun. “Multilingual Medical Documents Classification Based on MesH Domain Ontology”. *International Journal of Computer Science Issues* Vol. 9 (2012)

Kolonja, Ljiljana, Ranka Stanković, Ivan Obradović, Olivera Kitanović and Aleksandar Cvjetić. “Development of terminological resources for expert knowledge: a case study in mining”. *Knowledge Management Research & Practice* Vol. 14, no. 4 (2016): 445–456

Stanković, Ranka, Cvetana Krstev, Ivan Obradović and Olivera Kitanović. “Indexing of Textual Databases Based on Lexical Resources: A Case Study for Serbian”. In *Semantic Keyword-based Search on Structured Data Sources*, Cardoso, Jorge, Francesco Guerra, Geert-Jan Houben, Alexandre Miguel Pinto and Yannis Velegrakis, 167–181. Cham: Springer International Publishing, 2015

Trieschnigg, Dolf, Piotr Pezik, Viv Lee, Franciska de Jong, Wessel Kraaij et al.. “MeSH Up: Effective MeSH Text Classification for Improved Document Retrieval”. *Bioinformatics (Oxford, England)* Vol. 25 (2009): 1412–8