

Semantic Search in a Personal Digital Library

Dmitriy Malakhov¹✉, Yuri Sidorenko¹, Olga Ataeva²,
and Vladimir Serebryakov²

¹ Lomonosov Moscow State University, Moscow, Russia
mda.develop@gmail.com

² Dorodnicyn Computing Centre of RAS, Moscow, Russia

Abstract. The article offers a solution to the problem of semantic search in a personal digital LibMeta library. It also describes the L-tag-based semantic search model. The article provides an algorithm to build up a keywords and clusters hierarchy by the means of iterative clustering and keywords extraction. This hierarchy is used to generate abstracts and extract L-tags from texts.

Keywords: Semantic search · Information search · Digital library · LibMeta · Clustering · Keywords extraction · L-tag

1 Introduction

It is traditionally believed that digital library resources consist of bibliographic records of traditional libraries and digital copies of the documents described by these records. Still technological development redefines the concept of both the libraries and their resources (that go beyond bibliographical records and their digital representation) and pushes the semantics of these resources to the forefront, which may call for various library resources classifications to be implemented. There have been developed different application field indexes that allow us to more specifically define resource topics. As a rule these means are either not enough to describe semantics, or are not sufficient to the new rules of library resources description, which leads to both more complex descriptions and more efforts taken to introduce new description means that correspond to the current demands.

With the new facilities provided by developing technologies a library user can make better use of all the digital library resources means. He or she has the ability to describe their field of interest in the standard terms of the subject field with dictionary thesauruses and ontologies at hand. This allows a user to organize and describe both his or her own collections and resources, as well as to make more detailed resource and field descriptions thus defining their terms more specifically.

The LibMeta [1] personal open semantic digital library is characterized by a flexible store of metadata for its own resources and types of described information resources. This way to describe the resources of the library makes the descriptions of its resources and objects universal and unattached to any subject field or users' field of interest. The description structure ensures maintained connections between different types of the recourses.

The flexibility of the resources description is insured by the use of OWL ontologies for metadata storage. This format provides a number of benefits, such as:

- the ability to perform SPARQL queries;
- getting additional knowledge with logical inference;
- easier integration with other libraries;
- the ability to change schemes to match new requirements.

Semantic search is the search of documents by their contents. The LibMeta library allows to conduct semantic search on metadata with the use of SPARQL queries, still it doesn't allow to conduct semantic search on book texts. The ultimate goal of this work is to boost the quality of LibMeta library services with the help of the ability to conduct semantic search on the library book texts.

Thus it is necessary to realize the semantic search system for the LibMeta library book texts. This system should use a search query in a natural language, taking semantics into account, to look for the relevant book texts. The semantics is supposed to be supported by synonym and hyponym dictionaries.

This article develops the research published in CEUR [2]. The third part dedicated to the semantic search model has been replaced: instead of the S-tag model description this article introduces the L-tag semantic search model, which is based on new principals. Part 8 that deals with the semantic search system architecture has been reworked as well.

2 The Semantic Search Organization

There are different ways to organize semantic search on texts. The latest years have seen semantic texts annotation gaining more popularity. There exist a variety of ways to solve the issue of semantic annotation. Each of them focuses on attaching a set of semantically close tags to a document or a part of a document. These tags help you to find these documents later on. Besides you can look for documents using the regular full-text search and then take these tags into account while working with the document, thus obtaining more information [3]. It is common to use names, places, organizations names or other subjects as the tags [4].

RDF stores that contain a set of concepts and connections between them are often used to describe the tags. Some methods use information from Wikipedia as from a large source of knowledge as well [5]. The semantic annotation methods has lately been more and more addressing the use of the massive interconnected Linked Open Data cloud [4, 6]. For example, The National Archives of Great Britain (42 TB) has been annotated with the use of GATE annotation tool [7].

Semantic annotation is not the only way to organized search. There exist findings based on classic full-text search, which broaden it, adding the ability to use synonyms within the query. Thus there had been made up an ontology based on terms from various articles with the help of UDC [8], which was then used to broaden a user's query. The approach that deals with the information on syntax, morphology and punctuation appears to be of special interest as well [9]. Unfortunately, the approaches described above haven't been implemented and are not commonly used.

Let us then focus on an alternative way to organize semantic search with the use of synonym and hyponym dictionaries.

There have been conducted a lot of experiments on using synonym and hyponym dictionaries to improve the full-text search quality. It is widely known that the usage of synonyms and hyponyms makes searching results excessive and less accurate [10].

What makes this approach special is that only the meaningful parts of a text are indexed. Depending on the objective, these parts can feature paragraphs, sentences, word-combinations or a man-made tags (for example: a hashtag). Changing the size of the meaningful part lets us control the accuracy and fullness. For example, when the fullness is small and we see sentences being indexed, we can try to index combinations of sentences.

3 The L-tag-based Semantic Search

3.1 The L-tag

Suppose we are given the following elements:

- A set of documents D , where a document stands for whatever exists.
- An alphabet A , where an alphabet stands for any finite nonempty set.
- A set of terms T , where a term stands for an ordered multiset of elements from A .
- A set of L-tags LT , where an L-tag stands for an ordered multiset of elements from T .

For example: an alphabet can stand for any alphabet of a natural language. Words and word-combinations of the chosen natural language alphabet are the terms of this alphabet. A search query composed of a sequence of words and word-combinations, made of the natural language alphabet A is an L-tag on a T-set, where the T-set is a multiset of words and word-combinations in the natural language of the alphabet A .

Let us consider that each L-tag represents informational requirements and can be used by the user to represent this requirements. Let us also suggest that the users of the system will not represent different informational requirements using the same L-tag.

Informational requirements stand for a weighty set of documents which is necessary for the user to solve a problem, where the weight of a document is defined by its relevance for the problem-solving process. The ultimate goal of the information search is the assessment of the document weight within the informational requirements represented by the query. The definition suggests that the informational requirements can cross, be embedded or coincide. This feature will be used further to define the similarity function.

We are usually certain to know which documents are featured in the informational requirements described by an L-tag. Further in the text there will be introduced the semantic function and the similarity function, which are meant to value the informational requirements described by different L-tags.

3.2 The Similarity Function

We can define the similarity function as the following:

$$\text{sim}:(l_1, l_2) \rightarrow R \quad (1)$$

where l_1, l_2 are L-tags and R is a set of real numbers.

The similarity function (1) should satisfy the following rules:

1. The similarity function (1) is always greater than or equal to zero. Besides the function is not limited.
2. If the informational requirements of the L-tag l_1 does not fully include the informational requirements of the L-tag l_2 then the similarity function equals zero.
3. The more the informational requirements of the L-tag l_1 include the informational requirements of the L-tag l_2 , the bigger is the similarity function (1).

The ultimate goal of the similarity function (1) is to determine how much the informational requirements of one L-tag include the informational requirements of the other L-tag. Based on the similarity function (1) estimation we can make conclusions about the interrelationships between the informational requirements described by two L-tags.

It is obvious that the similarity function is not generally symmetric, as it does not imply that when the informational requirements of one L-tag include the informational requirements of the other L-tag it stays all the same the other way round.

Let us note that the similarity function (1) sets the L-tags hierarchy, based on the informational requirements hierarchy. The similarity function can be differently estimated depending on the understanding of the way the informational requirements are represented by an L-tag. It is important that the similarity function (1) estimations are consistent with the rules stated above.

3.3 The Similarity Function Estimation Example

Let us consider the following L-tags:

- l_1 : 'A paper'.
- l_2 : 'A scientific paper'.
- l_3 : 'A scientific paper on computer science'.
- l_4 : 'A scientific paper on physics'.

We can say that the informational requirements of the L-tag l_1 include the informational requirements of the L-tag l_2 , which include the informational requirements of both the L-tag l_3 and the L-tag l_4 .

According to the similarity function (1) rules, it should possess the following properties:

$$\text{sim}(l_1, l_1) \geq \text{sim}(l_1, l_2) \quad (2)$$

$$\text{sim}(l_1, l_2) \geq \text{sim}(l_1, l_3) \quad (3)$$

$$\text{sim}(l_1, l_2) \geq \text{sim}(l_1, l_4) \quad (4)$$

$$\text{sim}(l_2, l_1) = 0 \quad (5)$$

Property (2)–(4) follow the second similarity function (1) rule, when property (5) follows the first rule.

Let us note that the similarity function (1) rules provide no regulations for the value of the function for the (l_2, l_3) and (l_2, l_4) pairs:

$$\text{sim}(l_2, l_3) ? \text{sim}(l_2, l_4) \quad (6)$$

For example, we can estimate the similarity function (1) the following way:

$$\text{sim}(l_1, l_1) = \text{sim}(l_2, l_2) = \text{sim}(l_3, l_3) = \text{sim}(l_4, l_4) = 1 \quad (7)$$

$$\text{sim}(l_1, l_2) = 0.6 \quad (8)$$

$$\text{sim}(l_2, l_3) = 0.7 \quad (9)$$

$$\text{sim}(l_2, l_4) = 0.6 \quad (10)$$

$$\text{sim}(l_1, l_3) = \text{sim}(l_1, l_2) * \text{sim}(l_2, l_3) = 0.42 \quad (11)$$

$$\text{sim}(l_1, l_4) = \text{sim}(l_1, l_2) * \text{sim}(l_2, l_4) = 0.36 \quad (12)$$

In any other cases the similarity function (1) estimation equals zero.

3.4 The Semantic Function

The semantic function can be defined as follows:

$$\text{sem}:(d, lt) \rightarrow R \quad (13)$$

where d is a document, lt is an L-tag and R is a set of real numbers.

The semantics function (13) should satisfy the following rules:

1. The semantic function (13) is always greater than or equal to zero. Besides the function is not limited.
2. The semantic function (13) is equal to zero if the informational requirements of the L-tag lt are not satisfied with the document d .
3. The more is the weight of the document d in the informational requirements of the L-tag lt , the bigger is the semantic function (13).

The ultimate goal of the semantic function (13) is to define to determine how much the informational requirements of an L-tag correspond to the information in the document. Based on the semantic function (13) estimation we can make conclusions about the informational requirements described by an L-tag.

The semantic function (13) can be differently estimated depending on the understanding of the way the informational requirements are represented by an L-tag and of the way this or that information from the document can satisfy this or that informational requirement. The semantic function (13) estimation should be consistent with the rules stated above.

3.5 The Semantic Function Estimation Example

Let us consider the following L-tags:

- l_1 : 'A paper'.
- l_2 : 'A scientific paper'.
- l_3 : 'A scientific paper on computer science'.
- l_4 : 'A scientific paper on physics'.

Let us consider the following documents:

- d_1 : carries the information of the document being a paper.
- d_2 : carries the information of the document being a scientific paper.
- d_3 : carries the information of the document being a scientific paper on computer science.
- d_4 : carries the information of the document being a scientific paper on physics.

For example, we can estimate the semantic function (13) the following way:

$$sem(d_1, l_1) = sem(d_2, l_2) = sem(d_3, l_3) = sem(d_4, l_4) = 1 \quad (14)$$

$$sem(d_2, l_1) = 0.6 \quad (15)$$

$$sem(d_4, l_1) = 0.36 \quad (16)$$

$$sem(d_3, l_1) = 0.42 \quad (17)$$

$$sem(d_3, l_2) = 0.7 \quad (18)$$

$$sem(d_4, l_2) = 0.6 \quad (19)$$

In any other cases the semantic function (13) estimation equals zero.

3.6 The Semantic Function Interpolation

As seen from the previous examples of the similarity function (1) estimation and the semantic function (13) estimation, the latter can be excessive (15–19).

Suppose we are given a set of points $(d, lt) \in P$, where d is a document, lt is an L-tag. Suppose the semantic function (13) is known in the points of P and is unknown in the other

points, when the similarity function (1) is known everywhere. Then we can interpolate the semantic function (13):

$$\begin{cases} sem(d, lt), (d, lt) \in P \\ \max_{(d, lt_1) \in P} (sem(d, lt_1) * sim(lt_1, lt)), (d, lt) \notin P \end{cases} \quad (20)$$

Interpolation usually means calculating a value of a function in a point based on the value of the function in the vicinity of this point. In our case to estimate the semantic function (13) we use the values of the semantic function (13) in the vicinity of the point. This point should on the one hand be sufficiently close to that point, and on the other hand have a sufficiently big value of the semantic function (13). Interpolation can as well be conducted differently in case it satisfies the semantic function (13) rules.

The semantic function (13) interpolation helps to make its estimation process easier. Instead of estimating the semantics function (13) in all points (14–19) it is enough to estimate it in the points (14). For the cases (15–19) the estimation can be obtained through interpolation (20).

Thus, with the similarity function (1) estimation for all the L-tags and with the semantics function (13) estimation for P , we can estimate the weight of a document in the informational requirements of any L-tag.

It is crucial to understand that the more points belong to the set P , the more precise the semantics function (13) estimation is. This is why the size of the set P is a compromise between the difficulty and the quality of the semantics function (13) estimation.

3.7 The Semantic Search Model

Let us consider that an L-tag lt describes a document d , if:

$$sem(d, lt) > 0 \quad (21)$$

Suppose we are given:

- A set of L-tags LT .
- A set of documents D , described by the L-tags from LT .
- The similarity function (1) on the data set LT .
- A set of points $(d, lt) \in P$, where $d \in D, lt \in LT$.
- The semantic function (13) on the data set P .
- The semantic function (13) interpolation on the data set D and LT .

The semantic search model can be defined as implementation of the information search model:

- Each document should be described by a set of L-tags. Suppose an L-tag lt describes a set of document d , if the value of the semantic function (13) for the pair (d, lt) equals zero.
- A query can feature a set of L-tags LT that expresses the informational requirements of a user.

- The relevance function fixed on a data set of documents D and a data set of queries LT can be estimated by the semantic function interpolation (20).

Thus, the suggested model epitomizes the information search concept. What makes this model specific is that it implies singling out the main and most meaningful parts of a document, expressed by L-tags, and estimating the value of these L-tags for the document with the help of semantic function interpolation (20). Besides, the similarity function (1) estimation can be done with the use of synonyms from a thesaurus, which enables us to compare L-tags both syntactically and semantically. All this lets us claim this model to be the semantic search model.

Using the model enables us to reduce the semantic search objective to the objective of defining similarity function (1) on the data set LT and to the objective of defining the semantic function (13) on the data set P . This lets us single out or assign the meaning-reflecting L-tags beforehand by calculating the semantic function (13) and then use these L-tags when searching.

Obtaining the relevant documents data set $\{d\} \subseteq D$ for a certain query $q \in LT$ can feature three stages:

4. Obtaining a data set of L-tags $\{lt\} \subseteq LT$:

$$sim(q, lt) > 0 \quad (22)$$

5. Obtaining a set of documents $\{d\} \subseteq D$:

$$sem(d, lt) > 0, lt \in \{lt\} \quad (23)$$

6. Sorting the set of documents $\{d\}$ by their meanings:

$$sem(d, q), d \in \{d\} \quad (24)$$

4 Extracting L-tags from Texts

As shown above, we can carry out semantic search using L-tags if they are extracted from book texts and had their semantic function (13) estimated. Let us now consider the way to automatically extract L-tags from texts.

Text keywords stand for words and word-combinations that convey the main idea of the text and differ it from the other texts of the collection.

To match the content of a text, an L-tag should include keywords of this text. An L-tag can be a keyword, a sentence or a paragraph that includes a keyword. In this case the value of the semantic function (13) should depend on the L-tag keywords significance. Thus the objective to extract L-tags can be reduced to keyword text scanning.

As a keyword of a text should separate this text from the other texts, keywords depend on both the text they belong to and the whole collection of texts which opposes that text. If we divide a set of texts into semantically close text groups, we can see a keyword as a marking indicator that characterizes a text and the corresponding text group.

From another point of view, if we use keywords as indicators to divide texts, we can boost the speed and quality of such division by cutting indicators space and filtering noise.

Thus, clustering as the process of dividing texts into groups can use marked keywords, when the keywords marking algorithm can use the results of clustering. With clustering it is possible to divide a great multitude of texts into groups (clusters) and then look for keywords for the included texts according to those groups. This process can be repeated several times, alternating clustering with keywords extracting.

As a result we get a hierarchical structure of the documents in the collection and the corresponding keywords hierarchy.

5 Extracting Keywords

Suppose we are given a data set of texts D on the data set of terms T . A term stands for a word or a word-combination. The data set D is divided into a number of clusters C . The objective is to scan the data set D and then extract the keywords that characterize the cluster of this text.

Keyword marking traditionally involves two stages. The first stage features marking keywords candidates. At this stage stop words are deleted, parts of speech or the candidates that are not featured in Wikipedia headlines are filtered. The second stage is meant to check how close the candidates are to the text semantic-wise. This check is conducted with the help of supervised and unsupervised machine learning algorithms [11].

What makes our objective differ from the standard keywords extraction objective is the fact that the keywords should depend on both the document under scanning and some other documents that belong to the same field. The proposed approach can be upgraded with the help of the standard problem-solving algorithms [11].

Let us consider a simple way to solve the problem. Suppose we have a random text $d \in D$. For every cluster $c \in C$ and term $t \in T$. Let us estimate the probability of $d \in c$ when $t \in d$:

$$P(d \in c | t \in d) = \frac{|(t \in d \wedge d \in c)|}{|(t \in d)|} \quad (25)$$

where

- $|(t \in d \wedge d \in c)|$ is the number of documents from the cluster that feature the term t .
- $|(t \in d)|$ is the number of documents that feature the term t .

Suppose there is a threshold N , then we say that the term t characterizes a cluster c if:

$$P(d \in c | t \in d) > N * \max_{c_i \in C} (P(d \in c_i | t \in d)).$$

Thus, for every document d all terms that characterize the cluster of a document and are included in this document are seen as keywords K_d if the estimation $P(d \in c | t \in d)$ is high enough.

In order to extract L-tags we need to estimate the semantic function (13) value. The value of the semantic function (13) for a random document d and an L-tag lt is estimated as follows:

$$\sum_{k \in K_d \wedge k \in lt} P(d \in c | k \in d) \quad (26)$$

6 Clustering

Suppose we are given a data set of documents D on the data set of keywords K . The objective is to find out the best number of clusters for the data set of documents D to be used while conducting the clustering later on.

We will use the clustering method k-means++ [12] which enables us to divide a set of documents into clusters k in linear time.

The clustering quality factor with a parameter k is the value of Q_k , which is equal to the sum of roof-mean-square deviations of the centers of the obtained clusters for N iterations. Thus the smaller is Q_k , the more stable and profound is the clustering.

If k is too big or too small, it will affect the quality of the follow-up keyword extraction. That is why when choosing the parameter k it is necessary to set the up-value limit k_1 and down-value limit k_2 .

The best value of k is somewhere in between of k_1 and k_2 . You should chose that value of k which minimizes the value of Q_k for N iterations.

For example: Suppose $k_1 = 8$, $k_2 = 16$, $n = 10$. Then it will take 80 iterations to find the best clustering stability-wise, with the minimum value of Q_k .

7 Abstracting

After conducting a semantic search on book texts we face the problem of the search results representation. We can use the whole text of a book that meets the search query, an annotation of a book or an abstract that has been automatically compiled beforehand. The idea to generate an abstract of a book on a user's demand seems to be more preferable.

Abstracting is a process of building up a summary (abstract) of a document. Abstracting is used for visual search results representation. Abstracts can be static and dynamic.

Static abstracts are used to represent summarized information about the whole document. A static abstract is a one-time generated abstract which does not depend on a user's search requirements.

Dynamic abstracts are generated at the moment when a user gives a query and feature summarized information about the relevant for the query parts of a text.

Abstracts can be generated with the use of a dominant-based algorithm [13]. Such approaches are quite common. The dominants may feature those extracted L-tags which are sentences.

8 The System Architecture

Figure 1 demonstrates the interaction scheme of the semantic search on library data. We can emphasize the following components:

- **The data uploading module** receives data from two sources. The metadata source supplies the module with bibliographic records. The metadata reaches the RDF store where a user can get it with the help of SPARQL queries. RDF store is Jena-based. The document source provides book texts which are saved in the file system.
- **Hierarchical clustering and keyword extraction module** launches clustering of all the book texts in the system. The clustering results are the ground for keywords extraction. Then for each cluster starts the process of clustering on a data set of keywords, which results in new keywords being extracted for each cluster. These two processes are performed alternately, until there is a text hierarchy and a keywords data set.
- **Indexes and Abstract generation module.** L-tags include sentences with keywords. According to the extracted keywords, L-tags get extracted and the semantic function (13) gets estimated. A set of the most valuable L-tags is formed for each document to be used later when generating its abstract. The extracted L-tags are indexed into Postgres DBMS. Each L-tag gets a list of documents linked to it.

To interact with the system a user should put a query in a natural language. Before being fulfilled, the query enriches itself with a lot of synonyms and hyponyms from the thesaurus. The L-tag search is conducted with the help of GIST and GIN. The similarity function (1) is estimated with the Okapi BM25 relevance-defining algorithm [14]. After the query the system looks for the relevant L-tags with the lists of linked documents. A dynamic abstract is formed for each document. This abstract represents an ordered set of the document L-tags which are relevant to the user's query.

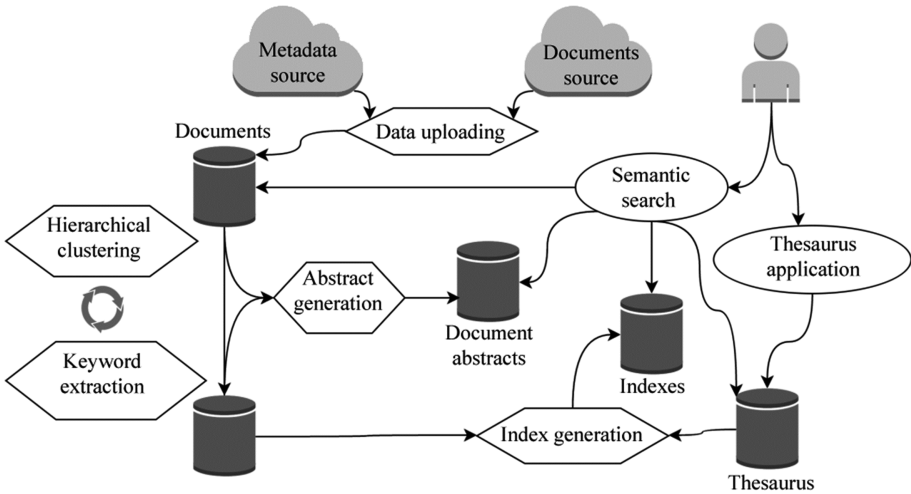


Fig. 1. Interaction scheme

A user can also search for documents with SPARQL queries in RDF store.

To improve the results of the search a user can edit the thesaurus that features synonyms and hyponyms.

9 Conclusion

The research featured the realization of a prototype semantic search system on bibliographic data and book texts.

Taking the semantic LibMeta library as an example we demonstrated the scientific relevance on the research. Implementing the proposed strategies helps to improve the library services quality.

There have been considered different ways of text semantic search realization. There has been implemented the L-tag model, the similarity function (1) and the semantic function (13). There has also been covered the semantic function interpolation (20). The query search objective has been reduced to the objective of similarity function (1) and semantic function (13) estimation. The proposed model was implemented when conducting search on the Postgres DBMS.

The work focused of the L-tag extracting algorithm which needs a data set of extracted keywords to operate.

There has been demonstrated a keyword hierarchy building process with the help of iterative alteration of clustering and keywords extraction. The proposed keywords extraction algorithm enables us to use the information on a cluster of a document. Clustering is performed with the k-means++ algorithm.

To visually represent the semantic search on texts results there has been demonstrated the approach to static and dynamic abstracting.

The proposed algorithms can be improved by the existent solutions, still they were deliberately simplified in the prototype framework. The further research will involve:

- Improving quality of the keywords extracting and the abstract generating algorithms.
- Experiments to improve the clustering quality.
- Experiments on L-tags extraction with the use of UDC hierarchy.
- Allocated keywords extraction and clustering on the Hadoop cluster.
- Distributed L-tag search system.

References

1. Ataeva, O.M., Serebryakov, V.A.: Personal digital LibMeta library as an open linked data integration environment. In: RCDL-2014 (2014). http://ceur-ws.org/Vol-1297/042-47_paper-8.pdf
2. Malakhov, D.A., Sidorenko, Y.A., Ataeva, O.M., Serebriakov, V.A.: Semantic search as a means of interaction with the digital library. In: DAMDID/RCDL, pp. 148–155 (2016). <http://ceur-ws.org/Vol-1752/paper14.pdf>

3. Giannopoulos, G., Bikakis, N., Dalamagas, T., Sellis, T.: GoNTogle: a tool for semantic annotation and search. In: Aroyo, L., Antoniou, G., Hyvönen, E., Teije, A., Stuckenschmidt, H., Cabral, L., Tudorache, T. (eds.) ESWC 2010. LNCS, vol. 6089, pp. 376–380. Springer, Heidelberg (2010). doi:[10.1007/978-3-642-13489-0_27](https://doi.org/10.1007/978-3-642-13489-0_27)
4. Bontcheva, K., Tablan, V., Cunningham, H.: Semantic search over documents and ontologies. In: Ferro, N. (ed.) PROMISE 2013. LNCS, vol. 8173, pp. 31–53. Springer, Heidelberg (2014). doi:[10.1007/978-3-642-54798-0_2](https://doi.org/10.1007/978-3-642-54798-0_2)
5. Berlanga, R., Nebot, V., Pérez, M.: Tailored semantic annotation for semantic search. Web Seman. Sci. Serv. Agents World Wide Web, 69–81 (2015). doi:<http://dx.doi.org/10.1016/j.websem.2014.07.007>
6. Alahmari, F., Magee, L.: Linked data and entity search: a brief history and some ways ahead. In: Proceedings of the 3rd Australasian Web Conference (2015). <http://crpit.com/confpapers/CRPITV166Alahmari.pdf>
7. Maynard, D., Greenwood, M.A.: Large scale semantic annotation, indexing and search at the national archives. In: LREC, pp. 3487–3494 (2012). <https://gate.ac.uk/sale/lrec2012/tna/tna.pdf>
8. Zakharova, I.V.: Speaking of a way to realize semantic document search on digital libraries. Ufa State Aviation Technical University Bulletin (2009). <http://cyberleninka.ru/article/n/ob-odnom-podhode-k-realizatsii-semanticheskogo-poiska-dokumentov-v-elektronnyh-bibliotekah>
9. Voskresensky, A.L., Khakhlin, G.K.: Semantic search tools. In: The ‘Dialog’ International Conference on Computer Linguistics and Intellectual Technologies (2006). <http://www.nkj.ru/prtnews/29136/>, <http://www.dialog-21.ru/digests/dialog2006/materials/html/Voskresenskij.htm>
10. Lukashevich, N.V.: Thesauruses in the information search problems (2010). <http://www.nsu.ru/xmlui/handle/nsu/9086>
11. Hasan, K.S., Ng, V.: Automatic keyphrase extraction: a survey of the state of the art. ACL **1**, 1262–1273 (2014). doi:[10.3115/v1/P14-1119](https://doi.org/10.3115/v1/P14-1119)
12. Arthur, D., Vassilvitskii, S.: k-means++: The advantages of careful seeding. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1027–1035. Society for Industrial and Applied Mathematics (2007). doi:[10.1145/1283383.1283494](https://doi.org/10.1145/1283383.1283494)
13. Chanyshv, O.G.: Dominant association fields and text analysis. Sobolev Institute of Mathematics SB RAS (2011). doi:[10.1145/1238844.1238847](https://doi.org/10.1145/1238844.1238847)
14. Mayfield, J., McNamee, P.: Indexing using both n-grams and words. In: TREC, pp. 361–365 (1998). <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.61.899>

Data Analytics and Management in Data Intensive
Domains

XVIII International Conference, DAMDID/RCDL 2016,
Ershovo, Moscow, Russia, October 11 -14, 2016,
Revised Selected Papers

Kalinichenko, L.; Kuznetsov, S.O.; Manolopoulos, Y.
(Eds.)

2017, XII, 281 p. 64 illus., Softcover

ISBN: 978-3-319-57134-8