Query Disambiguation Based on Clustering Techniques

Panagiota Kotoula, Christos Makris

Query Disambiguation based on Clustering Techniques

Panagiota Kotoula[1] and Christos Makris[1]

[1] Department of Computer Engineering and Informatics, University of Patras, Patra, Greece
Rio 26500 Patras, Greece, e_mail:{kotoula, makri}@ceid.upatras.gr

Abstract. In this paper, we describe a novel framework for improving information retrieval results. At first, relevant documents are organized in clusters utilizing the containment metric along with language modeling tools. Then the final ranked list (ascending/ descending order) of the documents that will be returned to the user for the specific query, is produced. To achieve that, firstly we extract the scores between the clusters and the query representations and then we combine the internal rankings of the documents inside the clusters using these scores as weighting factor. The method employed is based in the exploitation of the inter-documents similarities (lexical and/or semantics) after a sophisticated preprocessing. The experimental evaluation demonstrates that the proposed algorithm has the potential to improve the quality of the retrieved results.

Keywords: Query Disambiguation, Information Retrieval, Query Reformulation, Clustering, Containment, Semantics

1 Introduction

One of the most popular research issues has always been the subject of improving the quality of ranking in Information Retrieval results. To this extent, information need is expressed through the form of queries submitted against a search engine or platform with the purpose of receiving any available information related to the query (Baeza-Yates and Ribeiro-Neto, 2011; Manning, Raghavan, and Schütze, 2008). The problem or the challenge in this process is the search machine's potential and capability to respond and deliver the fittest set of information for the specific query, if this information actually exists.

On the other hand, users that post their queries are not specialists but instead plain, every day users that are not usually aware of the best format to provide their input query either because they do not leverage the full potential of the search platform or because they cannot express their intention clearly. The search engine's greatest challenge is then, to understand this user's intention through this given input, the query, that is to disambiguate the terms that synthesize the query and attempt to satisfy the query request.

The disambiguation process in information retrieval concerns both the way the query is approached as well as the way that the retrieved documents are processed. Effective

retrieval functions and techniques have been mostly derived from the class of probabilistic models and several approaches have been successfully implemented towards this direction (Baeza-Yates and Ribeiro-Neto, 2011; Crofy et al. 2009).

Recently, the aforementioned models have been combined with the cluster-based retrieval approach. Clustering depends on the relationship between document and query processing. If documents' processing depends on the query, then clustering can be seen as gathering all documents related to this query under the same cluster. Otherwise, documents can be clustered according to the relation between their content information based on lexical or semantic similarities and independent to the query. The level of similarity decided will provide the different number of clusters for a given corpus and will utilize the overlapping and redundant information present in the documents. This kind of retrieval is based on the hypothesis that similar documents can satisfy the same query or queries (Jardine and van Rijsbergen, 1971; van Rijsbergen, 1979). Moving in this line of thought recently researches have depicted that if good clusters can be designed then the retrieval performance will be improved (Raiber and Kurland, 2014).

In this paper we propose a framework in order to improve query search results by leveraging specific types on information representation along with selecting the proper clustering algorithm to organize the documents. Our approach examines query -independent document processing and representation resulting to a lexical based inter-document similarity in order to form the clusters. The key of improvement in our results is the use of a generalized structural unit, namely *gloss*, derived from WordNet's thesaurus hierarchy (Fellbaum, 1988; Princeton WordNet Gloss Corpus, 1979) instead of words, namely *senses*. Usual problems like frequent irrelevant terms are faced effectively since the generalized approach we design elevates the terms that have the most similar sense(s) between each other throughout the document and when gathered they actually represent its content due to their semantics. Our final result is the list of documents returned for the query as an improved ranking.

The paper is organized as follows: in Section 2 we present a brief survey on similar techniques, while in Section 3 we describe an outline of our approach. In Section 4 we describe the lexical and semantics processing of the documents and in Section 5 we conclude the document processing presenting the clustering along with the techniques for evaluating internal scores of the contained documents. In Section 6 we process the query and explain the reformulation. Finally, in Section 7 we describe our ranking technique along with comparing our result with other techniques and platforms in Section 8 and conclude our research in Section 9.

2 Related Work

Many researches have been developed regarding document and query processing in terms of query disambiguation techniques. Most of them use additional databases that extend the analysis from the standard search engines. These databases are known as thesaurus or ontologies and are basically lexical databases (mainly for the English language) providing extra information and tools for analyzing content and query processing like sets of synonyms, antonyms, definitions and other data, all derived from a hierarchical structure.

In these researches the various processing rules and techniques are applied mostly in conjunction with the WordNet (Princeton University) (Fellbaum, 1988; Princeton WordNet Gloss Corpus, 1979) ontology, along with probability distribution models to smooth and arrange the results. Most of our techniques were inspired or derived from the work of (Giakoumi, Makris and Plegas, 2015) that raise the question of whether the texts and the query should be approached with dependency between each other or not. In any case, texts are being scored in respect to one global lexicon and clusters are being formed through KL-Divergence (Kullback and Leibler, 1951). Throughout the process, TF-IDF and SMLE (Smoothed Maximum Likelihood Estimate) models provide the necessary smoothing of the contained information.

In other articles like (Makris et al., 2014; Plegas and Stamou, 2012; Agrawal et al., 2009; Angel and Koudas, 2011) processing focuses around information redundancy. These techniques leverage greedy algorithms like MAXI-MAXU (Maximum k-Intersection, Maximum k-Union), to allocate the maximum intersection of similar information between pairs of documents. Upon locating the same context, a new document is being created that contains this context once along with any new information contained in the texts that participated in the intersection process over the specified threshold. The newly created content is being checked with the coherence metric that concludes whether the derived text is logical and valid.

In a series of additional papers, (Kanavos, et al., 2013; Makris et al., 2013; Makris and Plegas, 2013; Raviv, Kurland, and Carmel, 2016), a recent and interesting approach is being explored to provide further information in text and terms annotation. The process of Wikification leverages the structure of Wikipedia pages in order to assign additional weights in the scores assigned to terms through WordNet's disambiguation process. Further clustering of the texts is being performed in these papers too, using n-tuples. Finally, using the TAGME (Ferragina and Scaiella, 2010) technique in conjunction with WordNet results, the final score is assigned to the texts before being returned for a specific query.

Moreover (Levi et al., 2016; Raviv, Kurland, and Carmel, 2016) explored language models, clustering and Wikification techniques in order to improve the retrieval performance. Our work can be considered to act complementarily to these attempts since it explores the use of WordNet and its glosses as an extra refined mechanism.

3 Outline of the approach

The disambiguation process in information retrieval concerns both the way the query is approached as well as the way that the retrieved documents/ pages are handled from the (any) search engine. In order to decide how this process can be better applied, in terms of a source corpus or a web search in respect to a specific query, we define the way to modify and represent these documents appropriately while reformulating the query at the same time.

At first, we process the documents and we consider this step as completely independent from the query. The reason for this decision is an intuitive thought which dictates that *given a specific query, the chances to find an answer for it and this answer to be ideal are very low*; on the other hand, given a document we can extract one or more queries that can find their ideal answer(s) inside the document they represent.

For text processing we use the *WordNet 2.1* ontology by choosing to extract the *glosses* from the hierarchy. The *glosses* are sentences that describe the various *senses* also included in this model. That way we insert the idea of generality in our technique; in other words we define a new generalized structural unit to represent our texts. We examine and evaluate the use of *glosses* due to their structural position inside the *WordNet* model. As such instead of limited and isolated *senses*, we elevate the use of *glosses* with the purpose of adding or **discovering** additional (important per case) information for the terms that are contained in the text.

A significant step in our approach is *document clustering* based on the *containment* metric. Through *clustering* we target to achieve the maximum possible concentration of the same or similar content of all documents under the same interface (representation). This interface/ representation is the lexicon of the *cluster*, based on the documents it contains. This lexicon is being forwarded later in the process in order to be compared with the various representations of the query. At the same time, the documents inside these clusters also receive a score independently of the query.

The query is being processed in a similar way – like the documents – through *WordNet 2.1* ontology information extraction. It is then reformulated and through the *glosses*, extracted from its terms, various representations are being produced (all possible combinations between *glosses* of each term).

In the final stage of our processing, we produce the final ranked list (ascending/ descending order) applied on the documents that will be returned to the user for the specific query. To achieve that, firstly we extract the scores between the clusters and the query representations and then we combine them through the *Borda Count* Method (Kozorovitzky, and Kurland, 2011) providing the final ranking order of the documents to return.

For the implementation of this research's algorithm we created and used tools in *JAVA* and *Python* in an end-to-end application so that the largest part of the process to be automated and user's interference to be minimum.

4 Language model processing of documents

4.1 Document Representation

We begin our document representation technique using natural language processing based on NLTK tools (Bird, Klein, & Loper, 2016) and OpenNLP (OpenNLP, 2016) trained models to extract the terms from the documents and lemmatize them. The output of these steps are being forwarded to WordNet 2.1 interface, where for each document, the contained words one by one are set as input in the thesaurus.

To select the appropriate sense or senses from the WordNet schema we utilize the Wu & Palmer similarity measure (Wu and Palmer, 1994) which is depicted below:

$$similarity\ (c_i, c_j) = \frac{2\ depth\ \left(LCA(c_i, c_j)\right)}{depth(c_i) + \ depth(c_j)}$$

, where we define $depth()$ as the depth of the node inside the ontology and as $LCA()$ the lowest common ancestor of the two senses c_i, c_j in the hierarchy, for the

terms t_i, t_j. We use the above algorithm because we do not know a priori the fittest sense for each incoming term and Wu & Palmer provide us the ones for which the similarity is maximized. The similarity is calculated over the senses but the algorithm moves one step further and retrieves from the WordNet ontology the respective gloss that the sense belongs to.

To this point, we have extracted all the necessary information from the WordNet schema and assigned these sets of glosses (sentences) to each one of the retrieved documents. Each document is now being represented by a set of glosses retrieved for its terms instead of the more common methods that stop to the retrieval of senses only. The main advantage of our approach is that there is no need to perform extensive smoothing techniques like TF-IDF. That is because by extracting a more generalized description for each term, the words inside the document that concern the actual concept will finally be expressed by more similar or the same glosses and as such collect the majority of the content. Instead, words that do not contribute to the overall conceptual representation will eventually be limited to minimum frequency in the following steps.

4.2 Language Model for Internal Scoring of documents in a cluster

In the next step, we represent the information contained inside the documents with a language modelling approach that aims to conclude scores for each document given a larger set of glosses contained in a fixed vocabulary V.

Given a constructed fixed vocabulary V, the idea is to calculate a probability measure over strings that belong to it (Manning et al., 2008). In this work, we construct these vocabularies V containing a fixed set of sentences and calculate the probability of those sentences to exist inside the document. Since our unit for representation is no longer one word but instead a set of words, we apply the same rules for calculating the probability metric over sentences (sets of strings).

For each document and for each contained sentence we calculate these probabilities using the SMLE model (Smoothed Maximum Likelihood Estimate). SMLE model adjusts the representation of the document in respect to the frequency of its contained sentences, in other words based on the number of occurrences of each sentence s in the fixed vocabulary V:

$$M_d^{SMLE}(s) = \begin{cases} \frac{f_{s,d}}{l_d} - c, & if\ s \in d \\ eps, & if\ s \ni d \end{cases} \quad , \quad \forall s \in V$$

, where $f_{s,d}$ is the number of occurrences of sentence s in document d and l_d is the number of terms contained in both d and V; also, eps is a very small quantity of the order of 10^{-10} and c is estimated as:

$$c = eps * \frac{|V| - n_d}{|V|}$$

, where $|V|$ is the number of sentences in vocabulary V and n_d the number of sentences both in the vocabulary V and the document d.

The Smoothed version of MLE containing quantities eps and c is performed in order to avoid the zero probability problem of MLE, that is:

$$M_d^{MLE}(s) = 0, \forall s \in V\ \&\ \forall s \ni d$$

Using the SMLEs produced we can then calculate the document's score against the fixed vocabulary V by adding these sub-SMLEs for all contained sentences inside the document representation.

5 Clustering

5.1 Containment

Documents are now represented by probability distributions and we can focus on how to leverage the similar ones in order to proceed with further clustering them. For this step we chose to use a simple but efficient metric known as *Containment* (Broder et al., 1997) accompanied with the K. To examine whether a document d_1 is contained inside another document d_2 we use the following formula:

$$Containment(d_1, d_2) = \frac{|S(d_1) \cap S(d_2)|}{|S(d_1)|}$$

, where $S(d_i)$ is the set of glosses that the document d_i contains. Containment is being calculated from both directions and we keep the one that produces the higher score. The maximum value for the containment is 1.0 which means that document d_1 is fully contained inside document d_2. The result in each comparison is rounded up to one decimal and the measure is calculated for all possible pairs of documents – removing from the final set those with containment equal to zero 0.0.

5.2 Containment Based Clustering

Having extracted the containment scores between all pairs of documents and keeping only the directions that produce the largest scores we now proceed to the clusters. Specifically, each cluster will be formed by those documents that the containment score between them belongs to a specific pre-defined set of values. One document may be present in more than one clusters.

For the final clustering of documents we have three tools: the content (containment), the value of the SMLE and the mapping between senses and glosses. We choose to use the simplest approach, namely the containment, to divide the texts into clusters. Specifically, having each time one hundred documents for processing we can put clusters of ten/fifteen documents setting containments limits in the range intervals of 0.1 units to meet separately all possible spaces between 0.1 and 0.9. The number of fifteen documents per cluster emerged after experiments. As for the other two metrics, the SMLE will be used for the internal ranking of documents within the cluster so that to get the final ranking of the representations of the query, in the last step of our processing. Finally, the mapping between concepts and glosses on the subject, can be used for further analysis but we stop at this point because we aim to end use only glosses. Nevertheless, one can combine senses and glosses and produce information more well defined to the nature of documents. Such information can produce more glosses for the representation of documents, to combine concepts from each other so that the glosses to be more targeted. We leave this as a matter of future research.

Returning to the clustering we develop the algorithm to extract clusters according to the analysis we did earlier. First, we define the first cluster with documents having containments between the 0.8 and 0.9 interval. Then we repeat this step every 0.1 units: 0.6 – 0.7, 0.4 – 0.5, 0.2 – 0.3 and 0.1.

In particular we run the list of document pairs and using their containment metric we pick up initially all pairs of values in the content space 0.8–0.9. Then we collect all the documents whose containment has values between 0.6–0.7. There is always the case of a document from the previous calculated interval, to appear and we permit this as case, considering that this raises the chances of finding more similarity between query and representation of a cluster because of the extra vocabulary the document adds to that cluster. We repeat the same procedure for the remaining intervals. When we completed the separation in the first cluster, we can reduce even more the documents per cluster and display different documents for the same intervals.

The distribution of the documents inside the clusters resembles a normal one; clusters produced for scores between 0.3 and 0.7 contain larger numbers of documents than clusters near the edge with scores 0.2 to 0.1 or 0.8 to 1.0 as shown in Figure 5.1:

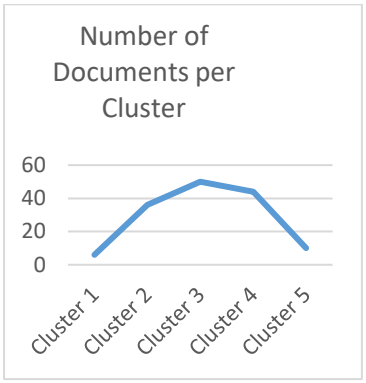

Figure 5-1 Normal Distribution of contained documents

Finally, based on the SMLE language model d, the score for each document inside the cluster is being produced. The fixed vocabulary V for each cluster which is used for this score is derived from the separate vocabularies of all documents contained in this cluster. The glosses occur one time at most since the global cluster vocabulary must not contain redundant information. The score for each document varies per cluster it belongs to.

It is important to mention two key characteristics for the above process:

1. The score for each document is produced based on the fixed vocabulary of each cluster it belongs to as such one document may have more than one scores based on the clusters it belongs to.
2. The option to create sub clusters is also utilized in our approach and the technique aims to create a more uniform representation inside the clusters in respect to the number of contained documents. It proves that the aforementioned technique performs worse than the normal distribution.

6 Query processing

To process the query we follow the same lexical and language model probability approaches, where we tokenize the input, lemmatize the separate terms and provide them as input to WordNet 2.1 in order to retrieve the respective glosses. In the document processing we chose to keep those glosses, whose senses produced the highest scores for the terms. For the query we choose to keep all glosses produced by the WordNet 2.1 schema. Our intuition behind this decision is that the user who posed the query might have had in mind any specific intention out of various possible ones; so, if a query can mean a lot of different things we have to keep all those meanings instead of just some.

In the final step of this process we reformulate the query by producing all possible glosses for each of its terms and we check all the possible combinations between them as shown if Figure 6.1. The final output will be a set of different query representations containing glosses.

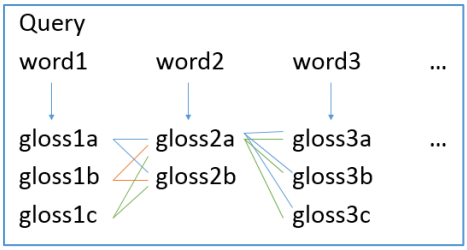

Figure 6-1 Normal Distribution of contained documents

7 Ranking

To produce the final list of documents that will be returned to the user, our algorithm calculates a similarity score between the global fixed vocabulary V of each cluster and each of the query representations. We annotate the query representations as: q_i ,$1 < q < Q$, where Q is the total number of query representations. Each cluster's vocabulary is annotated as: c_i, $1 < i < C$, where C is the total number of clusters. At the first step we produce the similarity measurement score between the two quantities for each query representation and each cluster's vocabulary.

A lot of techniques have been developed for lexical similarities over the years basically for application over characters between words. In our framework we adjusted a few of them to be performed over sentences – words instead of characters and sentences instead of words. We concluded in the use of two methods: *Positional N-Grams* similarity and *Levenshtein Distance*. The *N-Gram* similarity was used with position $n = 4$. The advantage of the use of the aforementioned similarity techniques is that they offer the possibility of two sentences to be partially similar and not totally the same. Between the two techniques, N-Gram executes in lesser time than *Levenshtein Distance* and produces better similarity scores. Based on comparing every representation q_i with every cluster C_j, there will be produced *Similarities*: S_{ij}:

Query representation	**Cluster**	**Similarity between query and cluster**
q_1	Cluster C_1	S_{11}
	Cluster C_2	S_{12}
	…	……
q_2	Cluster C_1	S_{21}
	Cluster C_2	S_{22}
	…	……

Then we adjust the ranking based on the documents of every cluster. As we already described the internal ranking of documents in each cluster employs SMLE and the containment metric.

Ranking in each cluster	**Query + Cluster: S_{ij} combination**
doc_1	S_{12}
doc_2	
doc_3	
doc_7	S_{31}
doc_8	
doc_9	
doc_1	
doc_2	S_{13}
…	…

The final score of each document is produced through Borda Count that combines the similarity between clusters and query representations with the internal document score as produced in each cluster. In particular the combined lists correspond to every distinct combination of cluster and query representation, the lists are ranked according to the internal scores of the document in each cluster and during the combination we use as extra weight the similarity between the query and the clusters.

8 Experimental evaluation

To perform our evaluation we explored 20 web queries from TREC 2012 WebTracks. We selected queries with the best characteristics against WordNet 2.1 and in respect to the quality of their corpus, in order to perform the experiments. All tracks use the 1 billion page ClueWeb09 (http://lemurproject.org/clueweb09/) dataset and contain a diversity task that contains a ranked list of documents that covers the query topic avoiding information redundancy. TREC assessors have assigned appropriate relevance judgement scores for the documents related to each of the queries. For our results we have employed Google and AOL search engines and for each we have retained the top 100 pages returned. Common identifier of the quality of the results (both for our algorithm as well as for the Google and AOL engines results) is the N-DCG metric (Baeza Yates & Neto, 2011; Manning et al., 2008) .

Each query is being examined against WordNet and confirmed that all of its terms extract the necessary information – otherwise the query is dismissed. We retrieved up to 200 documents from the ClueWeb09 corpus related to each query. For the clustering of the documents we used the containment metric and created the clusters based on the following sets of values: 0.8 -1.0, 0.6-0.7, 0.4-0.5, 0.2-0.3 and 0-0.1. To measure the efficiency of our method we used (a) the relevance judgement scores from TREC 2012 specialists and (b) the N-DGC, Normalized Discounted Cumulative Gain metric.

When comparing our algorithm with the ideal list of documents returned for each query we have a success of 75% in a very well controlled environment where: i) the stop words list is extended to contain additional words based on the previous results, ii) clustering is preserved in the values presented above, iii) queries selected have documents with strict normal distribution of scores, iv) we keep all glosses for the query terms and v) we change the threshold in our ranking techniques per case. The following figure 8-1 represents the most representative samples of our results with scores [0, 4]. In the vertical axis 0 indicates no relativity at all between the document and the query and 4 is the highest relativity between them. In the horizontal axis, reside the documents evaluated for the query.

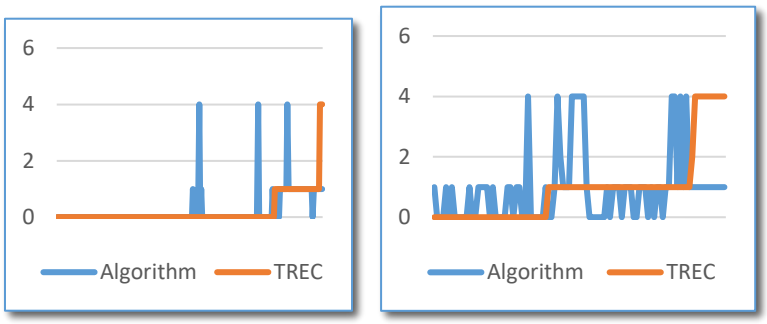

Figure 8-1 Algorithm evaluation against TREC assessment.

Finally, our algorithm was assessed against Google and AOL results for the same queries. The vertical axis in the following figure 8-2, shows the N-DCG metric and the horizontal contains the 20 queries. From the results, our algorithm performs much better than Google (69%) and better than AOL (71%).

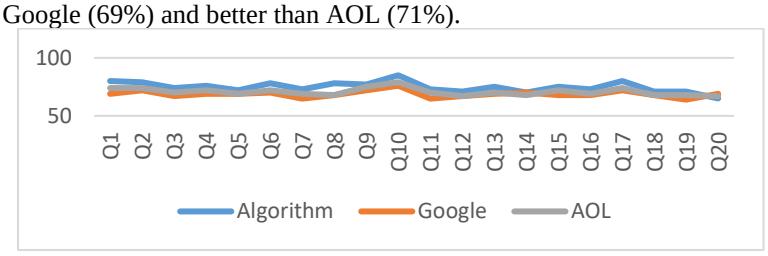

Figure 8-2 Algorithm evaluation against Google and AOL.

Google decides the returned list through Page Rank and focuses on user "clicks" as well as the incoming and outgoing links between pages. As such pages with lesser interactions between each other are hardly going to be presented in the list. On the other hand, the close values between our algorithm (75%) and AOL – which are both higher

than Google's - could be explained through the use of a common technique between the two approaches which is the clustering of the pages before they are returned for a specific query. Finally, we perform a soft comparison of our algorithm with the techniques and approaches presented in the related researches (Giakoumi, Makris, and Plegas, 2015; Plegas and Stamou, 2012) to further evaluate our algorithm. Our test range was much more limited than the ones used in the aforementioned paper as such the results are inconclusive. Specifically, our algorithm seems to perform 5% - 10% better but the queries used are not the same nor in the same extensive number. At the same time, the range of the corpus used in our experiments is significantly smaller mainly due to the limited capacity and related resources used for our research as such further experiments are necessary to conclude.

9 Conclusions

In this paper we presented a framework in an attempt to improve the quality in the retrieval performance of search engines. The novel approach of our research is the use of a more generalized structure unit, namely gloss, in order to represent our texts throughout language and probabilistic models in conjunction with a lately very effective technique – clustering. Our main goal in this approach is to introduce a more generalized way of modeling the input in a conceptual way in order to enforce the inclusion of independent modules, re-usable and easily attached to different methods and techniques. This way a model can be flexible enough to be attached in a variety of approaches. At the same time, our algorithm needs further improvement. In our approach the most significant one concerns the use of WordNet where the interface for the input has limited options; a different lemmatizer of stemmer could be developed to provide the words to WordNet in a better comprehensive and pos-tagged way without losing information. To this extend someone can further enhance the scores by applying the Wikification techniques in the final document ranking.

In a more complex approach, an entirely different thesaurus could also be used, like YAGO which offers a much better framework for conceptual analysis but so far lacks the tools to provide massive input and assess its results.

References

1. Agrawal, R., Collapudi, S., Halverson, A., Ieong S. (2009) Diversifying Search Results. In the 2nd International Conference on Web Search and Data Mining, pp. 5-14, 2009.
2. Angel, A., Koudas, N. (2011) Efficient Diversity-Aware Search. In Proceedings of the SIGMOD Conference, pp.781-792, 2011.
3. Baeza-Yates R.A, Ribeiro-Neto B.A. (2011): Modern Information Retrieval - the concepts and technology behind search, Second edition. Pearson Education Ltd., Harlow, England 2011, ISBN 978-0-321-41691-9
4. Bird, S., Klein, E., and Loper E. (2016) Natural Language Toolkit, WordNet Interface, Retrieved on: June 2016 from: http://www.nltk.org/howto/wordnet.html.
5. Broder, A.Z., Glassman, S.C., Manasse, M.S., Zweig, G. (1997): Syntactic Clustering of the Web, Computer Networks and ISDN Systems, Volume 29 Issue 8-13, Pages 1157-1166, Sep. 1997.
6. Croft, W. B., Metzler, D., Strohman, T. (2009): Search Engines - Information Retrieval in Practice. Pearson Education 2009, ISBN 978-0-13-136489-9, pp. I-XXV, 1-524

7. Fellbaum, C. (Ed.). (1998). WordNet, an electronic lexical database. Cambridge, MA: MIT Press.
8. Ferragina, P., Scaiella, U.: TAGME: on-the-fly annotation of short text fragments (by Wikipedia entities). In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM 2010), pp. 1625–1628. ACM, New York (2010)
9. Giakoumi, I., Makris, C., and Plegas, P.(2015): Language Model and Clustering based Information Retrieval, 11th International Conference on Web Information Systems and Technologies, WEBIST 2015: 479 - 486.
10. Jardine, N. and van Rijsbergen, C.J. (1971). The use of hierarchical clustering in information retrieval. Information Storage and Retrieval, 7:217-240
11. Kanavos, A., Makris, C., Plegas, Y. and Theodoridis, E. (2003): Extracting Knowledge from Web Search Engine using Wikipedia, 14th International Conference, EANN 2013, Proceedings, Part II: pp 100-109, Halkidiki, Greece, September, 2013.
12. Kozorovitzky, A.K., Kurland, O.(2011): Cluster-based fusion of retrieved lists. SIGIR 2011: 893-902
13. Kullback, S.; Leibler, R.A. (1951). On information and sufficiency. Annals of Mathematical Statistics. 22 (1): 79–86.
14. Levi, O., Raiber, F., Kurland, O., Guy, I. (2016): Selective Cluster-Based Document Retrieval. CIKM 2016: 1473-148.
15. Makris, C., Plegas, Y., Stamatiou Y.C., Stavropoulos, E.C., and Tsakalidis, A.K.(2014): Reducing Redundant Information in Search Results Employing Approximation Algorithms, 25th International Conference, DEXA 2014, Part II, LNCS 8645, pp. 240–247, Springer International Publishing Switzerland 2014.
16. Makris, C., Plegas Y., and Theodoridis, E.(2013): Improved Text Annotation with Wikipedia Entities, SAC '13 Proceedings of the 28th Annual ACM Symposium on Applied Computing, Pages 288-295.
17. Makris, C., Plegas, Y. (2013): Algorithms and Techniques of Personalized Search in Internet Environments Using Underlying Semantics, University of Patras, Department of Computer Engineering and Informatics, July 2013.
18. Manning, C.D., Raghavan, P., Schütze, H. (2008): Introduction to information retrieval. Cambridge University Press 2008, ISBN 978-0-521-86571-5, pp. I-XXI, 1-482
19. OpenNLP (2016) 1.5: Pre-Trained Models for 1.5 Series, Retrieved on: June 2016 from: http://opennlp.sourceforge.net/models-1.5/.
20. Plegas,Y., Stamou, S. (2012): Reducing Information Redundancy in Search Results, Journal of the American Society for Information Science and Technology — August 2012, DOI: 10.1002/asi, 63(8): 1581–1592.
21. Princeton WordNet Gloss Corpus (January 2008): WordNet Gloss Disambiguation Project, Sponsored by ARDA/DTO, SRI's AQUAINT Project, Retrieved on: July 2016 from: http://wordnet.princeton.edu/glosstag.shtml.
22. Raiber, F., Kurland, O. (2014), The Correlation between Cluster Hypothesis Tests and the Effectiveness of Cluster-Based Retrieval, ACM SIGIR, 1155-1158, 2014.
23. Raviv, H., Kurland, O., Carmel, D. (2016): Document Retrieval Using Entity-Based Language Models. SIGIR 2016: 65-74
24. Rebele, T., Suchanek, F.M., Hoffart, J., Biega, J., Kuzey, E., Weikum G.: YAGO: A Multilingual Knowledge Base from Wikipedia, Wordnet, and Geonames. International Semantic Web
25. van Rijsbergen. C.J. (1979): Information Retrieval, London: Butterworths; 2nd ed., 1979.
26. Wu, Z., Palmer, M. (1994) Verbs semantics and lexical selection. Proceedings of the 32nd Meeting of Association of Computational Linguistics; 1994; pp. 33–138.