

# SerfSIN: Search Engines Results' Refinement using a Sense-driven Inference Network

Christos Makris<sup>1</sup>, Yannis Plegas<sup>1</sup>, Giannis Tzimas<sup>2</sup> and Emmanouil Viennas<sup>1</sup>

<sup>1</sup>*Department of Computer Engineering and Informatics, University of Patras, Patras, Greece*

<sup>2</sup>*Department of Applied Informatics in Management & Economy, Faculty of Management and Economics, Technological Educational Institute of Messolonghi, Messolonghi, Greece*

**Keywords:** Search Engines, Semantic-enhanced Web applications, Re-ranking Model, WordNet Senses, Inference Network.

**Abstract:** A novel framework is presented for performing re-ranking in the search results of a Web search engine, incorporating user judgments as registered in their selection of relevant documents. The proposed scheme combines smoothly techniques from the area of Inference Networks with text processing techniques exploiting semantic information, and is instantiated to a fully functional prototype at present leading to a re-ranking whose quality outperforms significantly the initial ranking. The innovative idea is the use of a probabilistic network based to the senses of the documents. When the user selects a document, the belief of the network to the senses of the selected document is raised up and the documents that contain these senses are ranked higher. Also we present an implemented prototype that supports three different Web search engines (and it can be extended to support many more), while extensive experiments in the ClueWeb09 dataset using the TREC's 2009, 2010 and 2011 Web Tracks' data depict the improvement in search performance that the proposed approach attains.

## 1 INTRODUCTION

The global availability of information provided by the World Wide Web in the past few decades has made people's lives easier in terms of time saving and accuracy in information seeking. Commercial search engines have provided the necessary tools to the average Internet user to search for information about any topic he/she might be interested in and their everyday use rises constantly.

However there are still circumstances where one finds himself/herself wondering around the information maze posed to him/her by the Web. For example a user might be interested in "rockets" and missiles, but get highly ranked results after performing a query in a search engine about the famous NBA basketball team, or interested in animals and specifically in "jaguars", but get results about cars. Polysemy can be clarified by the general context of one's speaking, but search engines do not provide the necessary functionality to address this problem. Moreover, it often happens for the first set of results returned by a search engine to contain irrelevant information.

To overcome these drawbacks, in this paper we propose a new technique aiming to provide the necessary tools for the refinement of search results, taking into account feedback provided by the user. Besides the keywords provided from the initial query and the set of choices the user makes, we also utilize the semantic information hidden inside the returned pages. We feed this enriched combination of information in a probabilistic model and re-rank the results, without having to gather or import any additional data, thus making the whole concept simple and efficient.

Overall in this work, we describe the SerfSIN system (Search Engines results ReFinement using a Sense-driven Inference Network), that uses a re-ranking model based on inference networks and enhances it, in order to form an effective system for efficient reorganization of search results based on user choices. Moreover, we utilize the WordNet knowledge base in order to clarify the various senses that the query terms might carry and thus enrich our model with semantic information. Our technique is not restricted to WordNet, but it can also be extended to support other knowledge bases, such as

YAGO (Suchanek et al., 2007) and BabelNet (Navigli and Ponzetto, 2010).

The main idea of the paper is to transfer the belief of the user to the selected documents through the constructed network to the other documents that contain the senses of the selected documents. The re-ranking of the results is based on a vector that contains a weight for each document that represents the probability of the document to be relative for the user. We construct a probabilistic network from the terms and the senses of the documents so when the users select a document, the weights of the documents that contain these senses are taking bigger values and so they are ranked higher. Detailed experiments depict the superiority of the proposed system in comparison to the initial ranking and to previously relevant proposed techniques.

The rest of the paper is organized as follows. Section 2 overviews related literature and section 3 highlights the overall architecture. Next, in section 4 we present the re-ranking process, whereas the proposed approach is analytically covered in section 5. Section 6 depicts the results of our experiments. Finally, section 7 concludes the paper and provides future steps.

## 2 LITERATURE OVERVIEW

The last decade has been characterized by tremendous efforts of the research community to overcome the problem of effective searching in the vast information dispersed in the World Wide Web (Baeza-Yates and Ribeiro-Neto, 2011). A standard approach to Web searching is to model documents as bags of words, and a handful of theoretical models such as the well-known vector space model, have been developed employing this representation. An interesting alternative is to model documents as probabilistic networks (graphs), whose vertices represent terms, documents and user queries and whose edges represent relations between the involved entities defined on the basis of any meaningful statistical or linguistic relationships. Many works explore the usefulness of such graph-based text representations for IR like (Blanco and Lioma, 2012), (Boccaletti et al. 2006) and Bayesian Networks are prominent in them.

Bayesian Networks (Niedermayer, 2008) are increasingly being used in a variety of application areas like searching (Teevan, 2011), (Acid et al., 2003), (Callan, 2009), Bioinformatics (Ahmed et al., 2012), and many others. An important subclass of Bayesian Networks is the Bayesian Inference

Networks (BIN) (Turtle, 1991) that have been employed in various applications (Teevan, 2011), (Ma et al., 2006), (Abdo et al., 2011). Moreover, BINs form a major component in the search engine Indri's retrieval model (Metzler et al., 2005).

In this work we introduce a semantically driven Bayesian Inference Network, incorporating semantic concepts in order to improve the ranking quality of search engines' results. Related approaches were presented in (Lee et al., 2011), (Abdo et al., 2011). The authors in (Lee et al., 2011) enrich the semantics of user-specific information and documents targeting at efficient implementation of personalized searching strategies. They adopt a Bayesian Belief Network (BBN) as a strategy for personalized search since they provide a clear formalism for embedding semantic concepts. Their approach is different to ours, since they use belief instead of inference networks and they employ the Open Directory Project Web directory instead of WordNet. In (Abdo et al., 2011) the authors enhance the BINs using relevance feedback information and multiple reference structures and they apply their technique to similarity-based virtual screening, employing two distinct methods for carrying out BIN searching: reweighting the fragments in the reference structures and a group fusion algorithm. Our approach aims at a different application and employs semantic information, as a distinct layer in the applied inference network.

On the other hand and concerning search result's re-ranking, an interesting approach is to exploit information from past user queries and preferences. The relevant techniques range from simple systems implementing strategies that match users' queries to collections results (Meng et al., 2002), (Howe and Dreilinger, 1997) to the employment of the machine learning machinery exploiting the outcomes of stored queries, in order to permit more accurate rankings, the so called "learning to rank" techniques (Liu, 2011). There is also related but different to our focus work (Brandt et al., 2011) combining diversified and interactive retrieval under the umbrella of dynamic ranked retrieval.

The main novelty of our work centers in the transparent embedding of semantic knowledge bases to improve search engine results re-ranking. Also in order to achieve our purpose we create a new probabilistic model which takes as input different semantic knowledge bases. The most relevant to our work is the system presented in (Antoniou et al., 2012) embedding instead of Bayesian inference network, techniques based on the exploitation of semantic relations and text coverage between results.

Our work outperforms (Antoniou et al., 2012) improving its search performance, while it is based on a solid theoretical framework.

### 3 THE PROPOSED SYSTEM

In Figure 1 a high level overview of the overall system architecture is depicted. As shown the system is decomposed into the following core subsystems:

1. The **SerfSIN Web Interface** that interacts with the end users in order to provide searching and search refinement services.
2. The **Search API Modules** which are responsible for the communication with various search engines' APIs, in order to retrieve the relevant results. At present three search engines are supported, namely:
  - Google Search API, with two options provided to the user for the first set of retrieved results; the use of the Google deprecated API and the option of parsing the pages of the Google results.
  - Bing Search API.
  - Indri Search Machine (Strohman et al., 2005) over the ClueWeb09 Dataset (Callan, 2009).
3. The **Page Crawler** that fetches the content of the search engine results after the end user poses his first query.
4. The **HTML Decomposer** that parses the HTML code of a page and exports useful data, such as the title, keywords, metatags, highlighted text etc.
5. The **Sense Interference Network (SIN) Constructor** that creates the Network from the previous steps and reorganizes the search engines results.

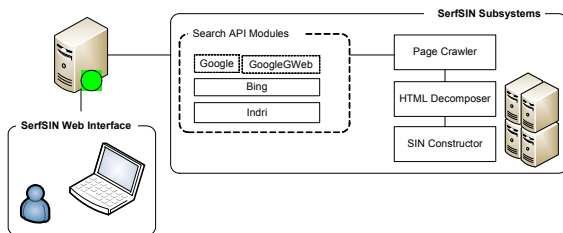


Figure 1: SerfSIN High Level System Architecture.

The subsystems work in a sequential manner and the first three interact with the users and the various Web sources.

In order for a user to utilize the services provided by SerfSIN, he/she simply makes a query after

setting the basic search parameters (explained in section 6) in the SerfSIN Web Interface. Every time the user clicks on a result, the core module is utilized producing the new improved ranking. In particular, the core subsystems work real-time and in parallel, rearranging the initial order and the algorithms utilized are presented in detail in the following sections.

### 4 THE RE-RANKING PROCESS

The whole process is initialized by the initial query performed by the user through the SerfSIN Web Interface. The query keywords are imported to the search engine selected (Google, Bing or Indri) through the relevant API and the results are collected by the Page Crawler. After this step, the initial results' ranking is the same as the one given by the search engine selected.

The next step improves the initial ranking based on the user's selections; the SIN Constructor is utilized accompanied by the input produced by the HTML Decomposer. This step runs iteratively every time the user makes a selection. In particular the proposed network is utilized, either as a standalone re-ranking algorithm, or in combination with the initial ranking returned by the search engine, or with the previous ranking of the results in the re-ranking process (if we have a series of re-rankings). In all cases we resolve ties, by following the previous ranking. The above options are expressed by the following two equations, for the ranking score of document  $d_i$ :

$$\text{New Ranking Score}_i = R_i \quad (1)$$

$$(n - \text{previous\_rank}(i) + 1) * (1 + \beta * R_i) \quad (2)$$

where  $R_i$  denotes the re-ranking weight provided by the network for  $d_i$  (its computation is described in the next section),  $\text{previous\_rank}(i)$  stands for the previous rank position of  $d_i$ ,  $n$  is the number of results retrieved and  $\beta$  is a user defined weight factor. Intuitively, when the factor  $\beta$  is raised the re-ranking process results in major rank changes. The equation  $(n - \text{previous\_rank}(i) + 1)$  introduces a factor based on the previous ranking of a result in the list. Equation (1) is used in the case of the network use as a standalone re-ranking system, while equation (2) is used for the composite case where the new ranking system is composed with the previous ranking of search results. In the new

ranking produced, the results are ranked according to the above calculated score. When the user selects further results, the same procedure is followed with the difference that the ranking produced by the previous phase is used as input for the next reordering.

Even though we assume that most results selected by a user are relevant, our scheme incorporates smoothly the previous ranking, hence it is robust to user misselections. A misselection of a result leads to the inclusion of its relevant information to the ranking process, but still can be made to not affect significantly the produced ranking.

### 5 RE-RANKING WEIGHT CALCULATION

Our extension of inference network, the Sense Inference Network (SIN), as depicted in Figure 2, consists of four component levels: the document level, the term level, the sense level, and a fourth level that represents the documents nodes and the value they take in order to re-rank the results; the fourth level can be considered to play the role of the query layer in the traditional inference network model and its presence signifies that we are not interested to model specific information needs, but re-rankings based on users' reaction. The SIN is built once for the retrieved documents of the query's results and its structure does not change during re-ranking. The document level contains a node ( $d_i$ 's) for each document of the query's results. For each term of the documents nodes texts, we add terms nodes ( $t_j$ 's) to the network and we interconnect the documents nodes with the terms nodes with arcs. The terms are induced by the retrieved pages by applying to them sequentially: (i) HTML stripping, (ii) removal of the stop words and finally, (iii) stemming using the Porter stemming algorithm (Porter, 1980). For each term node we also find its different senses using WordNet (Howe and Dreilinger, 1997) and we add them to the network. WordNet is a lexical database for the English language. It groups English words into sets of synonyms called synsets, provides short, general definitions, and records the various semantic relations between these synonym sets. The term nodes are connected through arcs to their different senses nodes ( $S_i$ 's). Finally, all the senses nodes which are contained in a document are connected to the respective document node at the last level. The formed network is a four level directed graph in

which the information flows from the documents nodes of the first level to the term nodes and then through the senses nodes to the documents nodes of the last level.

The innovative idea in our network is the existence of the level of senses (concepts) based on the WordNet knowledge base. The term nodes are connected to their different senses through directed arcs. The existence of a directed path between a document node and a sense node denotes that this sense is appearing in the respective document and more formally to the event that a sense has been observed in the documents collection. Similarly, a single sense node might be shared by more than one term. The dependence of a sense node upon the respective term node is represented directly in the network through an arc.

The final process for the construction of SIN is the creation of the arcs from the senses nodes to the document nodes at the last level. The last level's nodes are different from the nodes of the first layer; they represent the same entities (documents) but in different time instances (we depict this fact by drawing the arcs from the sense nodes to the nodes at the last layer as dashed, just to depict the fact that we are moving at a different time instances; this is a common practice that breaks cycles in Bayesian networks).

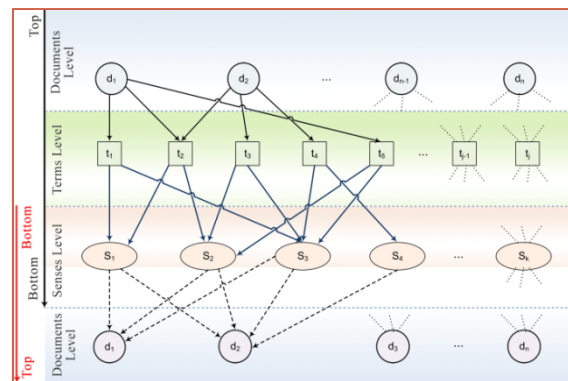


Figure 2: The Sense Inference Network.

The sense level models the hidden semantics and the belief of the network that the senses of a document are the senses that the user looks for. The senses nodes are connected to every node at the last level representing a document where this sense appears (this can be validated if there exists a path from the document node at the first level to the sense node). The document nodes at the last level have an accumulated belief probability that is used for re-ranking. The value of this belief is estimated based on the different senses of the document and denotes

the conceptual similarity between the document and the information need of the user.

### 5.1 Estimation of Probabilities and Rearrangement of the Results

Our inference network differs from that in (Turtle, 1991) since we employ it only as a weight propagation mechanism, using the (Turtle, 1991) machinery for computing the beliefs at the last level of the network, provided a set of prior probabilities for the first level; when a user selects a document its prior probability is raised and we compute the change at the beliefs at the last level. Our approach is different in comparison to (Turtle, 1991) and is adapted to the problem of the successful reorganization of search results. (Turtle, 1991) proposed an information retrieval model while our work proposes a re-ranking model.

In order to estimate the probabilities for the nodes of the constructed SIN, we first begin at the root (documents) nodes. Each document node has a prior probability that denotes the chance of the selection of that document from the user. For our collection and for each document  $d_i$  this prior probability will generally be set to be:

$$p(d_i) = \frac{1}{n}, \quad i \in [1..n] \quad (3)$$

where  $n$  is the number of the query results.

This probability will change into 1, when a result is selected in order to denote that this document is relevant from the user's point of view. This belief is transferred through the network to the senses nodes and then to the final layer representing the document nodes and changes the values of the classification weights, re-ranking the results.

For all non-root nodes in the SIN we must estimate the probability that a node takes, given any set of values for its parent nodes. We begin with the term nodes. Each term node contains a specification of the conditional probability associated with the node, given its set of parent document nodes. If a term node  $t$  has a set of parents  $pars_t = \{d_1, \dots, d_k\}$ , we must estimate the probability  $P(t|d_1, \dots, d_k)$ . Here we follow the inference network machinery described in (Turtle, 1991) employing the tf-idf weights for the term nodes and setting:

$$p(t|d_i) = 0,5 + 0,5 * ntf(t, d_i) * nidf(t) \quad (4)$$

where  $ntf()$  and  $nidf()$  are the normalized term

frequency and normalized inverse document frequency components for term  $t$ . In particular, if  $tf(t, d_i)$  denotes the number of times term  $t$  appears in  $d_i$ ,  $f_t$  the number of its occurrences in the collection and  $max\_tf(d_i)$  the maximum number of a term's occurrences in  $d_i$  then:

$$ntf(t, d_i) = \frac{tf(t, d_i)}{\max\_tf(d_i)}, \quad nidf(t) = \frac{\log(\frac{n}{f_t})}{\log n} \quad (5)$$

Based on these weights we can collect the belief for a term node from its father set by employing the formed weighted *sum* link matrix (Turtle, 1991).

The next step is the estimation of the probabilities for the sense nodes. If a sense node  $S$  has a set of parents  $pars_S = \{t_1, \dots, t_k\}$ , we must estimate the probability  $P(S|t_1, \dots, t_k)$ . The probability of a sense node denotes the importance of the sense for each term. Initially, the different meanings for each term are calculated using WordNet. The probability of a sense to be the unique sense of a term is equally likely among the different senses and is defined to  $1/(\text{number of different senses})$  for each term, hence we set.

$$p(S|t_i) = \frac{1}{m_i}, \quad (6)$$

where  $S$  is a sense of  $t_i$  and  $m_i$  is the number of the different senses of the term  $t_i$ . Based on these weights we collect the belief for the sense node from its father set, by employing a weighted *sum* link matrix.

Finally, we estimate the probability for a document node at the last level to be relative to the user's interests, based on the network's structure. The parents of the documents nodes at the last level of the network are the senses nodes. Therefore, the selection from the user of a result gives the network the ability to distinguish the senses, which interest the user and the document nodes at the last level get new weights signifying this knowledge. The senses nodes are affecting the document nodes according to their semantics as these are represented through arcs. We do not give weights but instead use a simple *sum* link matrix and the probability/belief for a document  $d$  at the last layer with  $q$  father senses is simply the sum of the beliefs of its father senses divided by  $q$ .

The Senses nodes are connected to the documents nodes for which there exist paths between the documents and the senses (through the term nodes). Final step is the computation of the weights at last level that entail the belief of the

relatedness of the documents to the selected document.

The values of the beliefs at the last level of document nodes  $W_i = bel(d_i)$  compose a vector  $(W_1, W_2, \dots, W_n)$ . Let  $(W_1^0, W_2^0, \dots, W_n^0)$  be the vector of beliefs at the last level before the selection of any results from the user (initial probabilities at the first level are all equal). For each result that the user selects, a new vector is estimated. In particular when the user selects a single result  $d$ , its probability at the first level is raised to 1, and a new belief for every document in the last level is computed. After this weight propagation, the re-ranking weight  $R_i$  for document  $d_i$  is defined as  $R_i = W_i - W_i^0$ . Similarly as the user selects further results, the probabilities of the respective nodes at the first level are raised to 1 and the beliefs are recalculated. By repeating this process and computing the re-ranking weights using formulae (1) and (2) (note that we always subtract from the initial weights) we reorganize the results accordingly.

The innovative idea is the use of the senses (not the terms) in order to re-rank the results. The importance of the terms modulates the importance of the senses. When the user selects a result, the terms of this document get higher values, thus the respective senses get higher values. Consequently, all the documents that have these senses get higher values in the final ranking process. The more senses (from the selected document) a document has, the higher ranking value it takes. Thus, the documents which contain the more senses (and the more times) from the selected document, are ranked higher.

## 6 EVALUATION OF SERFSIN'S PERFORMANCE

To carry out our evaluation, we explored 150 web queries; 50 queries from the TREC WebTrack 2009 (Clarke et al., 2009), 50 queries from the TREC WebTrack 2010 (Clarke et al., 2010) and 50 queries from the TREC WebTrack 2011 (Clarke et al., 2011) datasets respectively. All tracks employ the 1 billion page ClueWeb09 collection (<http://lemurproject.org/clueweb09/>). These ranked lists of results contain, for every page, relevance judgments made by human assessors.

We assessed our network performance by comparing the rankings it delivered to the rankings search machines returned for the same set of queries and results. For our comparisons, we relied on: (i) the available relevance judgments and (ii) the

*Normalized Discounted Cumulative Gain (nDCG)* measure (Jarvelin and Kekalainen, 2000), which quantifies the usefulness, or gain, of a document based on its position in the result list. The assumption in *nDCG* is that the lower the ranked position of a relevant result, the less valuable it is for the user; because it is not likely that it will be examined by the user. Formally, the *nDCG* accumulated at a particular ranking position  $p$  is given by:

$$DCG_p = rel_1 + \sum_{(i=2)}^p \frac{rel_i}{\log_2 i}, \quad nDCG_p = \frac{DCG_p}{IDCG_p} \quad (7)$$

where  $rel_i$  are the document relevance scores and  $IDCG_p$  is the Ideal DCG, i.e. the DCG values when sorting the documents by their relevance.

We set the re-ranking experiment by selecting randomly a relevant document, after posing each query and then performing re-ranking; we used *nDCG* before and after re-ranking to estimate the ranking performance.

The experiments carried out can be distinguished according to three basic parameters (all of the parameters can be accessed through the SerfSIN Web Interface at <http://150.140.142.5/research/SerfSIN/>):

- The search engine selected to perform the initial query.
- The percentage of the resulting pages text used to identify the terms that feed the second level of the constructed network (SIN).
- The participation of the initial ranking returned by the search engine, or the previous ranking of the results after a user's selection, in the re-ranking process, that is the significance of the proposed network compared to the initial ranking.

Our experiments were carried out using the Indri search engine over the ClueWeb09 Category B Dataset. We also utilized two different general purpose search engines, namely Google and Bing. We used 50 results in the case of Indri and Bing, while in the case of Google we used 20 in order to test if the chosen result set size has any effect on the attained performance. The queries used, as well as the relevant judgments rely on the ClueWeb09 Dataset and this is the reason why we selected Indri as our main experimental search engine. In the case of the general purpose search engines, there are no relevant judgments for all the returned pages; hence we ignored these pages in our measurements.

In relation to the second parameter, that is the text percentage of the resulting pages used to extract the network terms, we employ two distinct

approaches: we either use the whole text stemming from the HTML pages, or we use the most important text contained in the pages (e.g. keywords, meta description, Google description, h1, h2 tags, strong words etc.).

Table 1: Parameters' configuration for the twelve experiments (Most Important Parts: MIP, Full Text: FT).

| Exp. | Search Engine | Text Coverage | Use of Initial Ranking | Web Track Queries |
|------|---------------|---------------|------------------------|-------------------|
| 1    | Indri         | MIP           | Yes ( $\beta=1$ )      | 2009              |
| 2    | Indri         | MIP           | Yes ( $\beta=1$ )      | 2010              |
| 3    | Indri         | MIP           | Yes ( $\beta=1$ )      | 2011              |
| 4    | Indri         | MIP           | Yes ( $\beta=2$ )      | 2011              |
| 5    | Indri         | FT            | Yes ( $\beta=1$ )      | 2010              |
| 6    | Indri         | FT            | Yes ( $\beta=1$ )      | 2011              |
| 7    | Google        | FT            | No                     | 2010              |
| 8    | Google        | MIP           | No                     | 2009              |
| 9    | Google        | MIP           | Yes ( $\beta=1$ )      | 2010              |
| 10   | Bing          | MIP           | No                     | 2011              |
| 11   | Bing          | MIP           | Yes ( $\beta=1$ )      | 2011              |
| 12   | Bing          | FT            | No                     | 2011              |

Table 2: nDCG Averages for every experiment.

| Exp. | nDCG Aver. (Before) | nDCG Aver. (After) | Difference  |
|------|---------------------|--------------------|-------------|
| 1    | 0.259444438         | 0.309425945        | 0.049981506 |
| 2    | 0.236960647         | 0.285369305        | 0.048408658 |
| 3    | -0.09917842         | 0.055169266        | 0.154347686 |
| 4    | 0.100722072         | 0.011697012        | 0.112419084 |
| 5    | 0.062907506         | 0.120053429        | 0.057145923 |
| 6    | 0.103749409         | -0.03013304        | 0.073616369 |
| 7    | 0.538630812         | 0.591288965        | 0.052658153 |
| 8    | 0.525453777         | 0.593815908        | 0.068362132 |
| 9    | 0.549054511         | 0.575225518        | 0.026171007 |
| 10   | 0.43777936          | 0.486484735        | 0.048705375 |
| 11   | 0.434751776         | 0.468521246        | 0.033769469 |
| 12   | 0.43111441          | 0.449078004        | 0.017963594 |
| Avg: | 0.264370611         | 0.326333024        | 0.061962413 |

In the following, we depict the most representative experimental results from the vast material that we collected. We summarize all the depicted

configurations (Table 1) and then we particularize presenting the average nDCG values for every case (Table 2). We note that the proposed network has significantly better overall performance by 0.061962413, since the average nDCG values of the search engines and the proposed network are 0.264370611 and 0.326333024 respectively.

We present twelve experiments with alternative configurations based on the parameters analyzed in the previous paragraphs (Table 1). For every experiment we calculated the total average of nDCG values, for every rank position of the results. We also calculated their difference, in order to depict the corresponding improvement that is evident in all cases.

In the next figures the experiment's graphs are depicted, where for every rank position of the results, the average nDCG values of the search engine are compared to the average nDCG values of the proposed network. We started our analysis with the Indri search engine, where the experiments took place in a "controlled" environment employing the ClueWeb09 Dataset and its relevant judgments and we continued by studying the proposed network's performance for the cases of general purpose search engines.

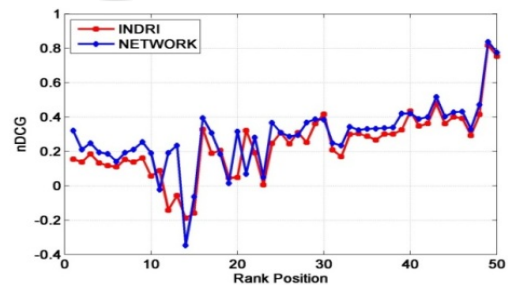


Figure 3: Experiment 1.

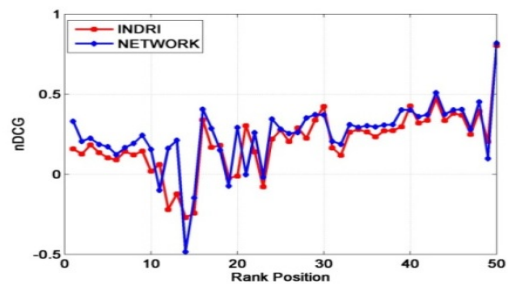


Figure 4: Experiment 2.

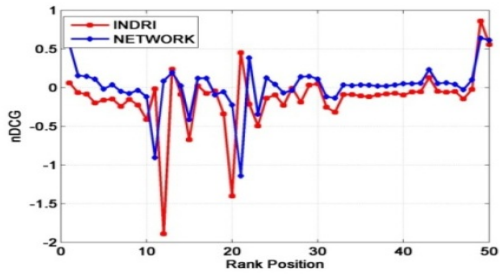


Figure 5: Experiment 3.

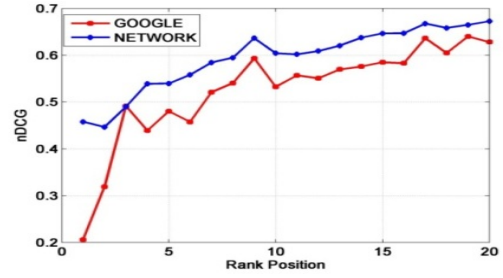


Figure 10: Experiment 8.

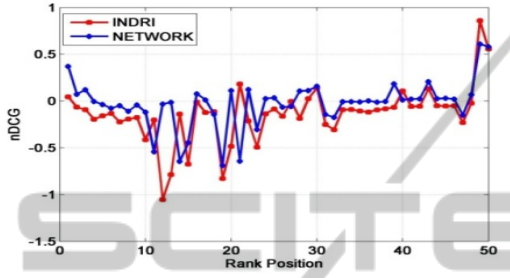


Figure 6: Experiment 4.

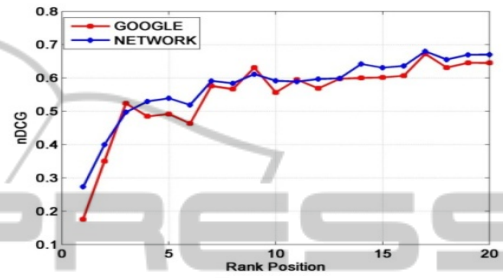


Figure 11: Experiment 9.

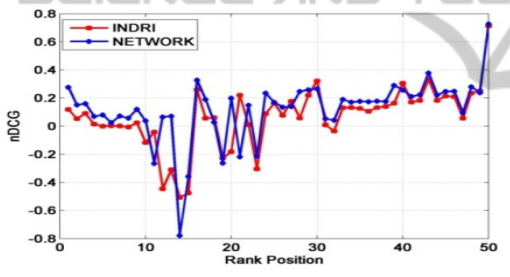


Figure 7: Experiment 5.

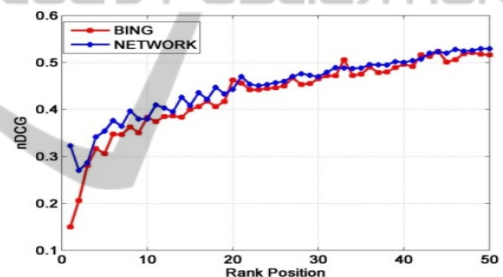


Figure 12: Experiment 10.

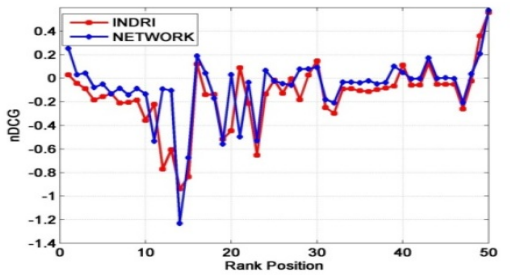


Figure 8: Experiment 6.

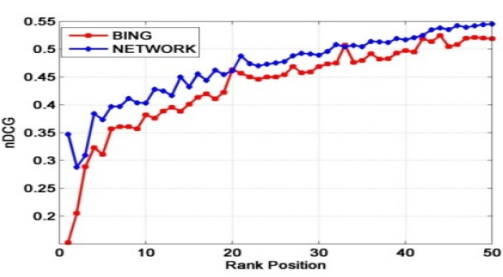


Figure 13: Experiment 11.

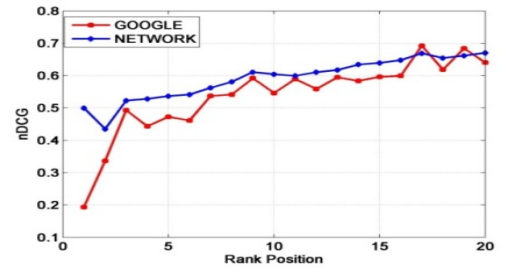


Figure 9: Experiment 7.

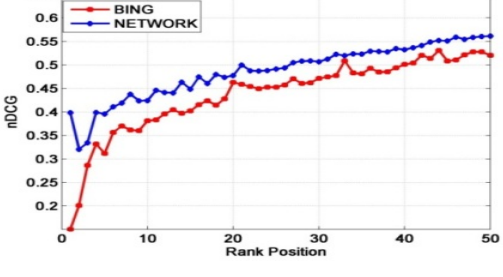


Figure 14: Experiment 12.



In the performed experiments we measured the proposed network's behaviour, as a standalone re-ranking system as well as for the composite case, where the previous ranking of the results is counted in the re-ranking process.

As the figures depict, the proposed network performs well in both cases: when used as a standalone re-ranking system (experiments 7, 8, 10 and 12), or in the case where the previous ranking is taken into account. Especially in the experiments 8, 10 where the network was used as the primary re-ranking engine in contrast to the general purpose search engines Google and Bing, the performance was excellent.

It is worth noticing that when the network takes into account the previous ranking (experiments 1-6, 9 and 11), it behaves better than the search engines and generally smoother (less rank changes when reordering) than the case where it acts as a standalone re-ranking system.

Another interesting observation is that the overall network performance did not deescalate when the most important text parts were used for the terms extraction process, instead of utilizing the full text. Moreover, as shown in experiment couples (1, 2), (2, 3) and (7, 8) the query sets employed did not play an important role in the quality of the results, as the network behaves equivalently in all configurations.

Finally, in our experiments (we indicatively depict experiments 3 and 4) we notice that when the  $\beta$  factor, increases from 1 to 2 the results remain the same, so in most of the cases the experiments were performed with  $\beta$  equal to 1; moreover our technique leads to better performance irrespectively of the result set size.

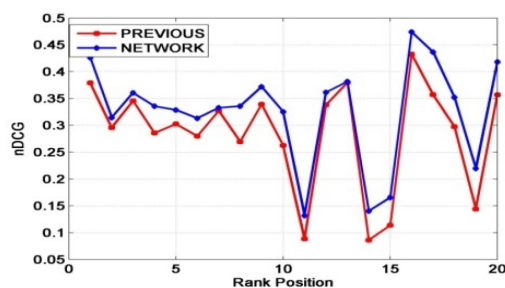


Figure 15: A comparison between the two approaches.

We have employed in our experiments the system described in (Antoniou et al., 2012) and in Figure 15 a diagram comparing their technique with ours is depicted, assessing the total average nDCG values of all the above experiments for the first 20 ranking positions. It is clear that the new approach performs

better in all positions by 0.033364065, since the average nDCG values of the previous approach and the proposed network are 0.284028022 and 0.326333024 respectively.

Moreover, the approach presented in this paper differs in the way the relevant information is retrieved, as well as the re-ranking process. The techniques employed in (Antoniou et al., 2012) are based on the exploitation of semantic relations and text coverage between results, while we employ a variant of a Bayesian inference network. The main advantages/features of our technique can be summarized as follows:

- Employment of a solid theoretical framework that can be tuned and be enhanced and not just heuristic use of terms and senses and their overlap.
- Ability to transparently enhance the inference network with other ontologies.
- Due to the inference network employed and our weight parameter, our technique is more robust to user errors, and a misselection of a result cannot affect significantly the final ranking.

It should be finally noted that we employed the paired t test, and the computed one tailed p-values were found to be very small (in all cases less than 0.01), proving that the results have significant differences and thus signifying the superiority of our technique in a strict statistical sense, and for a significance level of  $\alpha=0.05$ .

## 7 CONCLUSIONS AND FUTURE WORK

In this work we have presented a novel framework for re-ranking the search results of user queries employing a sense inference network in combination with semantic based techniques exploiting WordNet. The framework was applied in three search engines, and the attained results are quite encouraging depicting the ability of the proposed technique to capture the user preferences and produce a preferable ranking. A minor disadvantage seems to be the re-ranking process execution time; we plan to reduce the execution time by carefully tuning our code, without affecting the search quality. Moreover we aim to incorporate in our technique other knowledge bases besides WordNet, such as YAGO (Suchanek et al., 2007), and BabelNet (Navigli and Ponzetto, 2010) and to further enhance our inference network by embedding in it relationships between synsets present in WordNet.

Our approach comes as an improvement to the technique that was proposed in (Antoniou et al.,

2012), and embeds semantic information, but without exploiting the machinery of the inference networks. It could be motivating to further explore the connection between the two approaches, and propose a unified scheme incorporating both of them.

## ACKNOWLEDGEMENTS

This research has been co-financed by the European Union (European Social Fund-ESF) and Greek national funds through the Operational Program “Education and Lifelong Learning” of the National Strategic Reference Framework (NSRF)-Research Funding Program: Heracleitus II. Investing in knowledge society through the European Social Fund.

This research has been co-financed by the European Union (European Social Fund-ESF) and Greek national funds through the Operational Program “Education and Lifelong Learning” of the National Strategic Reference Framework (NSRF)-Research Funding Program: Thales. Investing in knowledge society through the European Social Fund.

Finally, authors also would like to thank the reviewers for their valuable comments and suggestions.

## REFERENCES

- Abdo, A., Salim, N., and Ahmed, A., 2011. Implementing relevance feedback in ligand-based virtual screening using Bayesian inference network. *Journal of Biomolecular Screening*, 16, 1081–1088.
- Acid, S., de Campos, L. M., Fernandez, J. M., and Huete, J. F., 2003. An information retrieval model based on simple Bayesian networks. *International Journal of Intelligent Systems*, 18, 251–265.
- Ahmed, A., Abdo, A., and Salim, N., 2012. Ligand-Based Virtual Screening Using Bayesian Inference Network and Reweighted Fragments. *The Scientific World Journal*.
- Antoniou, D., Plegas, Y., Tsakalidis, A., Tzimas, G., and Viennas, E., 2012. Dynamic Refinement of Search Engines Results Utilizing the User Intervention. *Journal of Systems and Software*, 85, 7, 1577–1587.
- Baeza-Yates, R. and Ribeiro-Neto, B., 2011. *Modern Information Retrieval: the concepts and technology behind search*. Addison Wesley, Essex.
- Blanco, R., and Lioma, C., 2012. Graph-based term weighting for information retrieval. *Information Retrieval*, 15, 1 (2012), 54-92.
- Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., and Hwang, D. U., 2006. Complex networks: structure and dynamics. *Physics Reports*, 424, 175–308.
- Brandt, C., Joachims, T., Yue, Y., and Bank, J., 2011. Dynamic Ranked Retrieval. *In WSDM '11*.
- Callan, J., 2009. The ClueWeb09 Dataset. available at <http://boston.lti.cs.cmu.edu/clueweb09> (accessed 1st August 2012).
- Chapelle, O., and Zhang, Y., 2009. A dynamic Bayesian network click model for web search ranking. *In Proceedings of the 18th International Conference on WWW, ACM, New York, USA, pp. 1–10*.
- Clarke, C. L. A., Craswell, N., and Soboroff, I., 2009. Overview of the TREC 2009 Web Track. *In Proceedings of the 18th TREC Conference*.
- Clarke, C.L.A., Craswell, N., Soboroff, I., and Cormack, G., 2010. Overview of the TREC 2010 Web Track. *In Proceedings of the 19th TREC Conference*.
- Clarke, C.L.A., Craswell, N., Soboroff, I., and Voorhees, E. M., 2011. Overview of the TREC 2011 Web Track. *In Proceedings of the 20th TREC Conference*.
- Fellbaum, C., 1998. WordNet, an electronic lexical database. The MIT Press.
- Howe, A. E., and Dreilinger, D., 1997. SavvySearch: a meta-search engine that learns which search engines to query. *AI Magazine*, 18, 2, 19–25.
- Jarvelin, K., and Kekalainen, J., 2000. IR Evaluation Methods for Retrieving Highly Relevant Documents. *In Proceedings of the 23rd International ACM SIGIR Conference*, 41-48.
- Lee, J., Kim, H., and Lee, S., 2011. Exploiting Taxonomic Knowledge for Personalized Search: A Bayesian Belief Network-based Approach. *Journal of Information Science and Engineering*, 27, 1413-1433.
- Liu, T-Y., 2011. *Learning to rank for Information Retrieval*. Springer.
- Ma, W. J., Beck, J. M., Latham, P. E., and Pouget, A., 2006. Bayesian inference with probabilistic population codes. *Nat. Neurosci.*, 9, 1432–1438.
- Meng, W., Yu, C., and Liu, K., 2002. Building efficient and effective metasearch engines. *ACM Computing Surveys*, 34, 1, 48–89.
- Metzler, D., Turtle, H., and Croft, W. B., 2005. Indri: A language-model based search engine for complex queries (extended version). *IR 407, University of Massachusetts*.
- Navigli, R., and Ponzetto, S. P., 2010. BabelNet: building a very large multilingual semantic network. *In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, 216-225.
- Niedermayer, D., 2008. *Innovations in Bayesian Networks*. Springer.
- Porter, M.F., 1980. An algorithm for suffix stripping. *Program*, 14, 3, 130–137.
- Strohman, T., Metzler, D., Turtle, H., and Croft, B., 2005. Indri: A Language Model-based Search Engine for Complex Queries. *In Proceedings of the International Conference on Intelligence Analysis (May 2-6, 2005)*, McLean, VA.

- Suchanek, F. M., Kasneci, G., and Weikum, G., 2007. YAGO: A core of semantic knowledge. *In Proceedings of the 16th International Conference on WWW*, 697-706.
- Tebaldi, C., and West, M., 1998. Bayesian Inference of Network Traffic Using Link Data. *In the Journal of the American Statistical Association*, 557-573.
- Teevan, J. B., 2011. Improving Information Retrieval with Textual Analysis: Bayesian Models and Beyond. Master's Thesis, Department of Electrical Engineering, MIT Press.
- Turtle, H. R., 1991. Inference Networks for Document Retrieval. Ph.D. Thesis.

