Information Sciences 180 (2010) 1908–1923

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Combining information from multiple search engines—Preliminary comparison

Maria Ganzha [a,b,*], Marcin Paprzycki [a,c], Jakub Stadnik [d]

[a] Systems Research Institute Polish Academy of Science, ul. Newelska 6, Warsaw, Poland
[b] University of Gdansk, ul. Wita Stwosza 57, Gdansk, Poland
[c] Warsaw Management Academy, ul. Kaweczynska 36, Warsaw, Poland
[d] Warsaw University of Technology, Warsaw, Poland

ARTICLE INFO

Article history:
Received 12 December 2008
Received in revised form 5 January 2010
Accepted 6 January 2010

Keywords:
Metasearch
Game theory
Consensus method
Auction method
Agent-based system
Decision support system
Internet search

ABSTRACT

The total number of popular search engines has decreased over time from its peak of the late 1990s. However, when combining the three remaining major ones (Yahoo!, Google, MS Live) with large repositories of information (e.g. BBC.com, NYT.com, Gazeta.pl, etc.), the total number of important information sources can be seen as slowly increasing. Focusing on utilization of search engines only, it is easy to observe that the same query issued to each one of them results in different "suggestions." The question thus arrives, is it possible (and worthy) to combine responses obtained from each one of them into a single answer set. In this paper, we look into three approaches of achieving this goal, which are based on: *Game theory*, *Auction* and *Consensus* methods, while our focus is to study (and compare) their "performance."

1. Introduction

By a simple experiment it is easy to establish that, for exactly the same query, different Internet search engines produce different results. Furthermore, this effect is particularly visible when more complicated queries are posed. Thus, we can say that each of the search engines "conceptualizes the world" in a different way. Let us now observe that, based on statistics provided by Internet monitoring entities, different users have different preferences as to which search engine to use (e.g. at the moment of writing of this paper, early in 2009, about 44% utilize Google as their primary source of information, while 29% use Yahoo! and 13% utilize MS Live [5]). Obviously, when considering possible sources of information on the Internet, one could also add large information repositories, such as news organizations: NYT.com, BBC.com, Gazeta.pl, or general purpose portals, e.g. wp.pl, onet.pl, etc. However, for the purpose of this paper we focus our attention on search engines only, and pose a question: is it possible to combine answers from different sources in such a way that the obtained answer is going to be "better" than when using only a single source?

Combining "advice" from multiple sources into a single response is grounded in a popular scenario, when a panel of experts is used to address a problem. In such case, each expert may have a different view of the situation, but in most cases their "joint wisdom" is considered to be better than when relying on their individual ideas. In the context of combining information from multiple Internet sources, we have decided to utilize three different methods: (1) *Consensus method* [27,24,26,29], (2) *Game theory* and (3) *Auctions* [31,30]. All three of these approaches have been originally posed as applica-

* Corresponding author. Address: Systems Research Institute Polish Academy of Science, ul. Newelska 6, Warsaw, Poland.
E-mail address: maria.ganzha@ibspan.waw.pl (M. Ganzha).

0020-0255/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2010.01.010

tions of (and implemented using) software agents. While utilization of agent technology is not necessary (it is clear that all needed functionalities can be achieved without agents), we follow our predecessors and use the JADE agent platform to implement the three methods. However, *none* of the results presented here depends on this particular approach to implementation. Finally, we briefly consider the question of utilization of the MySpiders client ([6]) that was designed to be used to combine responses from multiple Internet sources, and compare its results with the three approaches utilized in our work.

We proceed as follows. In the next section, we outline most important existing results that are pertinent to our work. Next, introduce the three approaches to information combining. We follow with experimental results and their analysis. Work presented here extends initial results presented in [34].

2. Related work

While there is a large number of research papers related in some way to the issue of advanced search, they are mostly focused on its aspects that are different from our work. The closest to what we do are metasearch engines ([4,7]). An extensive list of metasearch projects that existed in 2005 can be found in [32]. Unfortunately, most of these search engines, being commercial projects, do not provide information as to how they aggregate the results. Here, let us mention three of them. First, mamma.com, which is one of the oldest—founded in 1996—metasearch engines. Currently, its technology has been commercialized by Copernic Inc. and focused on searching and organizing information for individual PC's ([2]). Second, dogpile.com, which originally displayed search results separately for each individual search engine. Currently, it aggregates them somehow (dogpile technology is a property of Infospace Inc.), and for each result informs which search engines this result (link) was found in. Analyzing results for simple queries it can be conjectured that one of the key criteria, used in ordering combined results, is the number of search engines that a given URL was found in. Finally, the MySpiders project ([6]); an academic research project (summarized in papers found at [1]), which we have selected as a sample metasearch engine to compare results of our work with (see Section 6). This selection was based on the fact that approach used in this project was thoroughly described in research papers and thus is open to further scrutiny.

In the context of metasearch engines it is also worthy to note the Proteome Software project, which combines multiple searches for peptide identification. This proprietary project is an example of combining results in a narrowly defined field of knowledge (computational biology, in this case). However, it is unclear if approach used within it, if known, would generalize to Internet searches. Finally, it is worthy mentioning that, within the site that published the above mentioned report on metasearch engines ([32]), content published currently, extensively focuses on link positioning/search optimization for individual search engines, without much attention paid to the fact that different search engines produce different results.

Separately, work is done in the area of context-based information retrieval ([15,13,14]). In this case search involves not only search engines, but also knowledge about various forms of context (e.g. who is searching, where is she searching work/home etc.). While this research is definitely of great value, it goes a step forward to what we are interested in. Specifically, there are two possible approaches to utilization of context and multiple search engines. First, it is possible to "contextualize" individual results and then to combine them. Second, it is possible to combine results and then apply context to find the most relevant ones. As can be seen, in either case application of context is independent from combining results from multiple sources.

Next, a large body of research is devoted to ontology-based approaches. A typical example of an approach representing semantically-oriented techniques is Latent Semantic Indexing [12,21]. LSI estimates the semantic content of documents, and uses that estimate to rank them according to the relevance to user's query. Note that here, event though data processing is semantically-based, the proposed method does not invoke explicit ontologies. Furthermore, responding to the user-query involves analysis of the content of documents, which (if possible in the first place-in most cases multimedia content is demarcated using tags only) can be extremely resource consuming. Therefore, LSI can be applied to well defined collections of documents stored in a limited number repositories, but is not likely to succeed when applied to the Internet as a whole.

The next group of approaches can be seen as related to the foundations of the Semantic Web [8]. Specifically, methods considered here starts from the fundamental assumption that data/resources is/are ontologically demarcated. A typical example of work proceeding in this direction is [10]. Here, ontology is utilized as a generic solution to the problem of integrating information/resources residing in separate/independent repositories. As soon as it is assumed that there exists ontology that can be utilized in the system (and, hopefully there is only one such ontology, so one does not have to deal with ontology matching) documents can be annotated and classified according to such ontology. In this case, the semantics-based mining can extract fine-grained metadata from articles contained in digital repositories and annotate them. Moreover, current semantic reasoning techniques should allow to answer structured queries to access appropriate resources/documents. Similar approach, applied to a reduced scope application, has been discussed in [16]. Here, the semantic search was focused on a specific data set, the DBLP [3]; and attempt was made at addressing the question, how to find the most appropriate items within the DBLP. However, even in this case, the first step has to be to create an ontology/extract metadata from publications. Now, a crucial question arises: what would the success within the DBLP (CS-related publications only) prove, as far as usability of this approach on a large scale? Here, worthy noting is also "ontology-based focused crawling," described in [40]. In their work, Zheng and colleagues propose utilization of a neural network, which is applied in the context of a domain-specific ontology and utilized in classification of web pages. Their experimental results show that such approach outperforms the breadth-first search crawling, the simple keyword-based crawling approach, and crawling utilizing only ontologies.

The key observation to be made is that approaches similar to these three, work *only* under the assumption that there exists an agreed-on, explicitly defined ontology, and that documents in (various) repository(ies) are appropriately annotated. While vision behind this assumption is very appealing, today there are only very few repositories with ontologically annotated data. Furthermore, the issue of successful search across multiple ontologies (within the Semantic Web) is wide open. In the meantime users need methods and techniques that can be used today, instead of waiting for the idea of the Semantic Web to materialize. In this context it is also worthy noting a recent publication [19]. While it concerns primarily Semantic Web Services, observation contained in it apply also directly to the Semantic Web in general. Therefore, we believe that these facts combined, support relevance of research reported in this paper; where we try to address an existing problem without invoking future technologies that may or may not be successful in the first place.

Since plain semantic search is not likely to come to the mainstream anytime soon, considerable effort has been related to the so-called *keyword search*. Keyword search can be understood as searching inside documents for keywords that are a part of the query. Sometimes keyword search may involve also (instead) some form of metaserach, understood as searching within keywords that a resources are annotated with (not necessarily ontologically annotated; natural approach to deal with multimedia content). Note that in the latter case documents have to be pre-processed, which brings us back to what we have just discussed in the case of ontology-driven searches. In this area, one of the typical examples of research directions was reported in [20]. Here, a template-based approach to keyword searching was proposed. Separately, addressing an issue of capturing the meaning actually intended by users in their queries was discussed in [41]. Finally, in [38], an RDF-graph-based approach has been proposed, which explores connections between nodes that correspond to keywords in the query. This way, all interpretations of the query that can be derived from the underlying RDF graph, can be computed.

Unfortunately, keyword query searches not always provide needed answers. Therefore, sometimes, they are combined with semantic searches. An example of this approach was discussed in [9]. Here, an approach to query over textual data and semantic data in an integrated way is discussed. Hybrid searching, on the large scale (within the web environment), brings about its own challenges. Obviously, it requires semantic annotation of data, which cannot be assumed without serious reservations (see, above). Furthermore, it requires capability to store, index and integrate large amount of documents and semantic data, and thus questions of scalability come to play. To address this issue, in [38], authors present a unified framework to represent query both RDF data and documents in an integrated way, which is also capable of dealing with large amounts of RDF data and documents. A slightly different approach to address scalability issues was presented in [39], where the answer was sought in application of Grid computing.

Finally, it is worthy noting, that there exists an important research area that may provide one more interesting approach to combining responses from multiple sources. This area is known as *multi-criteria multi-expert decision making*. Here, the approach could follow that described, for instance, in [37,18]. However, this approach is deeply rooted in psychology (see, for instance references cited in [37]), and is most likely to be applicable in the case of "social networks-based searching," rather than for direct combining of results obtained from multiple search engines.

In summary, out of existing approaches, metasearch engines, which are most relevant to our work, are mostly closed commercial projects, quality of which can be assessed only through a large-scale comparative study involving human subjects (which is out of scope of our current research interests). Introduction of context to searches is not attempting at utilizing multiple search engines, but to "personalize" results from a specific repository. Semantic Web originating techniques deal with searching multiple repositories, however they assume existence of semantically demarcated data. This assumption cannot be made without reservations, in particular taking into account how few such repositories exist today. At the same time, research concerning keyword searches and hybrid searches involves mostly single repositories. Finally, we see no simple direct way of applying results obtained in the area of multi-criteria multi-expert decision making to combining and ranking results of queries issued to multiple search engines. Therefore, research discussed in this paper attempts to address in a very pragmatic and unique way an existing need of today.

3. Three main algorithms

Let us start our presentation by describing the three approaches to combining search results from multiple sources. In following sections we describe three core algorithms, while further details about their implementation, special situations that have to be dealt with, etc., can be found in Section 4, as well as in [34,33]. Note that we assume that the input for each method consists of a ranked set of results provided in a form that is "the same" for each data source (search engine). In this context, to illustrate all three methods in a uniform fashion, let us assume that sample data has been provided by the search

Table 1
Sample results to be used to illustrate work of the three methods.

Position	A_1		A_2		A_3	
	Link	Weight	Link	Weight	Link	Weight
1	L_1	35	L_2	30	L_3	30
2	L_3	20	L_3	25	L_1	25
3	L_2	10	L_1	20	L_2	15

engines and pre-processed (details about pre-processing can be found in what follows, as well as in [34,33]). Specifically, in Table 1, we present response-sets for three agents: $A_1,\ A_2$, and A_3. After pre-processing each one of them has in its response-set three links (URL's): $L_1,\ L_2$, and L_3. It has to be stressed, that response-sets found in Table 1 have been created purely artificially for the purpose of illustrating working of the three methods *only*.

Note that, in the general case, the weight (confidence value) of a link is calculated as follows: $|\texttt{Set of answers}| - i$, where $|\cdot|$ denotes cardinality of the set and i is a position of the URL in the answer set. However, the *Game theory* and the *Auction method* require that each result set contains the same URLs (in order that is specific to the given search engine). If this is not the case, these two algorithm break, since agent may "know nothing" about a certain URL and therefore comparison of ranks of this specific URL cannot be performed. Realizing this assumption is the part of the pre-processing phase (see, Section 4.1).

In the case of long-term use of the system, it is assumed that in addition to the ranks resulting from the order in the answer set, weights are going to assigned to each item in the result set. These weights would originate, primarily, from user feedback. Here, for instance, results reported in [35,17] could be utilized. However, in this paper we omit this issue as, among others, it would require running large number of experiments involving human subjects. While interesting in its own right, this was not a goal of our work.

3.1. Game theory

In our work, we have used a modified version of the *Game Theory*-based algorithm utilized in the *NeurAge* system [31,30,11]. Here, instead of voting for certain "classes of data," agents vote for URLs retrieved by individual search engines. Confidence values used in the original algorithm have been replaced by URL ranks (weights, obtained during the pre-processing phase of the algorithm; see Section 4.1). Furthermore, in the *NeurAge* system, as an answer, agents delivered a single "data class." However, when searching the Internet/data repositories, users expect to see multiple, ranked responses (ordered lists of links). Therefore, in the adapted approach, our algorithm iterates to find top ten links (obviously, this number can be modified). Specifically, each iteration utilizes the original "single-class selection" algorithm to pick a single URL (considered best at this stage). This URL is then included in the final answer set and removed from further considerations (deleted from answer sets proposed by each search engine). This process is then repeated (using reduced answer sets as the input). It should be stressed that this modification does not violate the main assumptions of the Game theory based algorithm [36].

In general, a "game" consists of set of players, set of moves (strategies) and specification of payoffs for each combination of moves. In a normal form the game is defined as follows:

There is a finite set P of players, which we label $\{1,2,\dots,m\}$. Each player k has finite number of pure strategies (moves) $S_k = \{1,2,\dots,n_k\}$ A pure strategy profile is an association of strategies to players, that is *m-tuple*

$$\sigma = (\sigma_1, \sigma_2, \dots, \sigma_m),$$

such that

$$\sigma_1 \in S_1, \sigma_2 \in S_2, \dots, \sigma_m \in S_m.$$

Let strategy profiles be denoted by Σ. A payoff function is a function

$$F : \Sigma \to \Re$$

which represents the reward given to a single player as the outcome of the game. Accordingly, within a game the payoff function has to be specified for each player in the player set $P = \{1,2,\dots,m\}$. Under these assumptions, a game in a normal form is a structure (P,S,F), where $P = \{1,2,\dots,m\}$ is a set of players, $S = (S_1,S_2,\dots,S_m)$ is a m-tuple of pure strategy sets, one of each player and $F = (F_1,F_2,\dots,F_m)$ is an m-tuple of payoffs.

In the game considered here, its components are as follows: (a) players are agents, (b) possible moves are: to change or to keep the URL, (c) payoffs for individual moves are defined as a 2×2 matrix. Each agent calculates two values: for keeping a given URL and for changing it. Those values may, or may not, change during each round of the game, depending on the outcome of the previous round.

At the beginning of the process, results obtained from individual search engines are filtered, ranked and updated (see below for more details). The URL ranking (weights assigned to links) resulting from the pre-processing stage of the algorithm represents "confidence in a specific URL." In each round of the game two agents with highest ranked URLs "compete against each other" and during a single competition they have two possibilities: to decide to keep their answer or to decided to change it. If the *keep action* has a higher value than the *change action*, the agent will keep its top URL for the next round. If, however, the agent decides to change its URL while the second agent decides to keep its URL, the latter is considered a *winner* of the round and the former is considered a *loser*. At this stage, the looser agent and its result set is discarded from further considerations. Afterwards, the next round starts (without the "looser agent" and its result set—as it was removed in the previous round). Process continues, until there is only one agent left and its top URL is the winner of this "big round." This URL is included in the final answer set (and removed from all result sets). Process is then repeated, without the URL that was selected in just completed "big" round. Note that, as a result of pre-processing, before the first "big round" starts, all result sets contain the same group of URL's. Taking into account that in each round a single (the same) URL is removed from all answer sets, this property is sustained during all stages of the game. Algorithm continues for 10 "big rounds" (until 10 "best" URLs are selected). The main algorithm for the game-theoretic approach is summarized in the following pseudo-code:

Input: URL rankings provided by each search engine (Search Agent)
Output: Ordered URL's (final answer set)
BEGIN

1. repeat until there are 10 URLs in answer list
2. repeat until one agent remains
3. find agent, which has the highest ranked URL, and note this URL—let those be the FA (first agent) and the FAU (first agent URL)
4. find agent which has the second ranked URL, and this URL—let those be the SA (second agent) and the SAU (second agent URL)
5. calculate keep and change payoff values as follows:

$$\begin{aligned} FA_{keep} &= rank(FA, FAU) - rank(FA, SAU), \\ SA_{keep} &= rank(SA, SAU) - rank(SA, FAU), \\ FA_{change} &= \frac{rank(FA, FAU) + rank(FA, SAU)}{2}, \\ SA_{change} &= \frac{rank(SA, SAU) + rank(SA, FAU)}{2}. \end{aligned}$$

6. determine agent actions by comparing their values—the action with higher value is the chosen action
7. determine round winner:
   - if action assigned to one of agents (FA or SA) is the keep action and action assigned to the other is the change action, the one that selected the keep action is marked as winner, the second one is marked as loser and is discarded from that "big round" of the game
   - if both agents are assigned the same action, their URL ranks are replaced by the values of the chosen action; if this situation occurs for the second time the following takes place:
     Depending on the initial ranks of URLs assigned to both agents, the one with the higher ranking is considered to be a winner of the Game, and the other is the loser.
8. add winning URL to the final answer set
9. remove winning URL from all agent answer sets
10. go to step 2 (start the next round)

END

Let us now see how this algorithm is going to work for the sample result sets found in Table 1. As it can be seen, the highest ranked URL is L_1 (with weight 35), while the next two have weight 30 (L_2 for agent A_2 and L_3 for agent A_3). Therefore, the first game of the first "big round" is between agents A_1 (for link L_1) and A_2 (for link L_2). In this round the following pay-out table is created (Table 2).

It means that A_1 is assigned the *keep* action, whereas A_2 is assigned the *change* action. Therefore, A_1 is winner in this round and A_2 is loser. Thus A_2 and its result set are removed from the further consideration until the game restarts for the next "big round."

After that, A_1 updates the rank of the URL it kept in the first round. That is URL L_1 is no longer ranked with 35 but with 25, which was the keep action payoff. This is followed with a game between agents A_1 and A_3 and ends with agent A_1 becoming a winner. First, both agents have the same "strategy"—"to change the URL". To yield a winner initial rankings (from the beginning of this "big round") of the URL under consideration are compared. Agent's 1 ranking of URL L_1 was equal to 35. Agent's 3 ranking of L_1 is equal to 25. As a result A_1 is winner in this round and A_3 is a loser. Since this is the last agent that A_1 could compete against, A_1 (and L_1) are winners. Next, L_1 is added to the final answer set and removed from answer sets of all agents and the process restarts. Obviously, Table 1 has to be modified; and is transformed into Table 3.

Based on values found in Table 3, we can see that in the next "big round" the first game will be between agents A_2 (for link L_2) and A_3 (for link L_3). The result of this game will be link L_3, while agent A_2 will be declared a looser and removed from

Table 2
Payout table of the first game of the first round of the sample *Game theory*-based approach.

"Strategy"	A_1	A_2
Keep	35 − 10 = 25	30 − 20 = 10
Change	(35 − 10)/2 = 22.5	(30 + 20)/2 = 25

Table 3
Result sets after the first "big round" of the sample *Game-Theory*-based approach.

Position	A_1		A_2		A_3	
	Link	Weight	Link	Weight	Link	Weight
1			L_2	30	L_3	30
2	L_3	20	L_3	25		
3	L_2	10			L_2	15

considerations. The second game will be between agents A_1 and A_3, resulting in link L_3 becoming the winning link. Therefore, the final outcome of the *Game theory*-based approach will be the answer set consisting of links ordered (L_1, L_3, L_2).

3.2. Auction-based approach

There exists a large body of research devoted to utilization of software agents in autonomous negotiations. Among them auction-based approaches play a prominent role. In the context of, broadly understood, "combining information from multiple sources" auctions have been used in the *NeurAge* system [30]. Since, similarly to the case of the *Game Theory*-based approach, in the *NeurAge* system a single category (answer) was returned, we have adapted it to return 10 distinct ordered URLs. To achieve this goal, we have modified the algorithm to proceed in rounds. In each "big round," each agent tries to "sell its product" (the URL). To do this it calculates "cost" of the URL. Costs are compared and agent with the highest cost is considered a loser. Afterward, the confidence values (weights) for selected URLs are updated by subtracting just calculated cost from their value. Henceforth, the next "small round" takes place. If the agent that was marked before as a loser, loses again, it and its result set are discarded from further auctions (in this "big round"). After removal of the twice-looser, the process enters the next "small round," and continues until a single agent remains with its selected answer. This process is repeated 10 times, while after each "big round" the URL that was selected to be included in the final answer set is removed from result sets of all agents. Note that, as in the case of the *Game theory*-based approach, during the pre-processing it is assured that all answer sets contain the same URL's (though in different orders). In this way it is assured that the approach will produce a "combined answer set." Here, we present an example of the flow of one "big round" of the *Auction* method:

Input: URL rankings provided by each search engine (Search Agent)
Output: Ordered URL's (final answer set)
BEGIN

1. repeat until there are 10 URLs in the final answer set
2. repeat until one agent remains
3. find highest ranked URLs for all agents and pair them $(A^{(i)}, U^{(i)})$
4. calculate costs for each agent:

$$cost(A^{(i)}) = \frac{\sum_{i=1, i \neq j}^{m} (rank(A^{(i)}, U^{(i)}) - rank(A^{(j)}, U^{(j)}))}{10}, \tag{1}$$

   where $U^{(i)}$ is the highest ranked URL for agent $A^{(i)}$ and $U^{(j)}$ the highest ranked URL for agent $A^{(j)}$.
5. find agent with highest cost—a loser; it may happen that all agents have the same costs—if it occurs twice the agent which is assigned the URL initially ranked as the lowest is considered a loser and thus removed from further negotiation, if it is so, go to 7.
6. if the agent is a loser twice in a row remove it from further auctions in this "small round"
7. update URL rankings for all agents as follows:
   $rank(A^{(i)}, U^{(i)}) = rank(A^{(i)}, U^{(i)}) - cost(A^{(i)}),$

   where the pair $(A^{(i)}, U^{(i)})$ is found at the beginning; at this point the winning URL can be changed
8. add URL to the final answer set
9. remove the URL from all answer sets

END

Let us now use data found in Table 1 to illustrate the *Auction*-based method. In the first step each agent calculates the "cost" of each of its links using formula (1). Results of such calculations are illustrated in Table 4. As a result, the A_1 agent is declared a looser (cost of its URL is maximal) and cost values recalculated by subtracting the cost value from them. Results of this operation are as follows:

$$(A_1, L_1) = 35 - 4 = 31,$$
$$(A_2, L_2) = 30 - 1.5 = 28.5,$$
$$(A_3, L_3) = 30 - 2 = 28.$$

Table 4
Calculation of the value of each link in the first round of a sample *Auction*-based approach; step 1.

	A_1	A_2	A_3
A_1	$\frac{(35-10)+(35-20)}{10} = 4$		
A_2		$\frac{(30-20)+(30-25)}{10} = 1.5$	
A_3			$\frac{(30-15)+(30-25)}{10} = 2$

These new values are included in the overall rankings for each agent. As a result of this process agents may need to change their favored answer (if the updated cost is smaller than that of the next one in ranking). However, this is not happening here, since all updated ranks are higher than the remaining ones. Then, agents enter the next round with their URLs ranked as presented in the Table 5.

Therefore, costs in the next step is presented in the Table 6. This means that agent A_1 is declared a looser again and removed from consideration. As it is easy to see, after the first "big round" of the *Auction*-based approach, link L_2 is the winning one and included in the final answer set; as well as removed from result sets of all agents. The resulting modified ranking of URL's remaining in result sets is summarized in Table 7.

Now, it can be easily verified that after the *Auction*-theory based process has been completed, the resulting list of ordered links (final answer set) is (L_2, L_3, L_1) (different than in the case of the *Game theory*-based approach).

3.3. Consensus method

The *Consensus* method was used previously in the *AGWI* system [24–28]. Its aim is to combine a set of answers into a final joint answer that is to represent a "consensus" between inputs. Differences between the *AGWI* system and the modified approach presented here are as follows. First, in the *AGWI* system the number of available search engines was larger than the number of *Search Agents* (and thus, based on a special algorithm, only some search engines were selected to be used). Here, as a default (and in experiments reported below) there are as many *Search Agents* as there are search engines (we use all of them). Second, the algorithm for measuring distances between result sets was adapted (to use the Levenshtein method [22]).

In the *Consensus method*, individual result sets provided by search engines are evaluated and a combined answer set (without repeating URLs) is created. Next, for each URL its average position in all result sets is calculated, and the combined answer set is sorted according to the average position of each URL. Then, the consistency of the resulting answer set is checked. Before performing the calculation, however, the result sets and the consensus are normalized; only a specific number of top URLs are incorporated into the answer. This number is of size of the smallest non-zero result set. To check

Table 5
URL rankings after the first "small round" of the sample *Auction*-based approach.

	A_1		A_2		A_3	
	Link	Cost	Link	Cost	Link	Cost
1	L_1	31	L_2	28.5	L_3	28
2	L_3	20	L_3	25	L_1	25
3	L_2	10	L_1	20	L_2	15

Table 6
Calculation of the value of each link in the second "small round" of the sample *Auction*-based approach; step 2.

	A_1	A_2	A_3
A_1	$\frac{(31-10)+(15-20)}{10} = 3.2$		
A_2		$\frac{(28{,}5-20)+(28{,}5-25)}{10} = 1.2$	
A_3			$\frac{(28-15)+(28-25)}{10} = 1.6$

Table 7
Result sets after the first "big round" of the sample *Auction*-based approach.

	A_1		A_2		A_3	
	Link	Cost	Link	Cost	Link	Cost
1	L_1	27.8	L_2	27.3	L_3	26.4
2	L_3	20	L_3	25	L_1	25
3	L_2	10	L_1	20	L_2	15

consistency of the consensus answer, average of distances between result sets and average of distances between each result set and the consensus (final) answer set are evaluated. If the average of distances is bigger than average of distances of result sets and the consensus answer, then the consensus answer is consistent; if not, the consensus answer is not consistent. The following listing presents the pseudo-code of algorithm for finding the consensus answer.

Input: URL rankings provided by each search engine (Search Agent)
Output: Ordered URL's (final answer set)
BEGIN

1. create set ***URLS*** consisting of all URLs from all individual result sets (without repetitions)
2. for each $U \in$ ***URLS***
   - create array $< t_1, t_2, \ldots, t_n >$, where t_i is position in which U appears in $r^{(i)}$;
   - if ***U*** does not appear in $r^{(i)}$ then set t_i as the value of the largest ranking increased by 1
   - divide each t_i by $weight(r^{(i)})$; if $weight(r^{(i)}) = 0$ divide by 0.01
   - calculate average $t(U)$ of values $(t_1, t_2, \ldots, t_n)$
3. consensus answer is obtained by ordering (sorting) elements of according to just obtained values

END

Having checked the consistency of the result set, the algorithm decides on the next step. When the consistency is low, the answer containing all results is returned and feedback is requested. If the consistency is high, 10 first URLs from the consensus answer are presented.

Let us now look into the calculation of the *Consistency*-theory-based answer for the input data presented in Table 1. Obviously, the answer set will consist of links (L_1, L_2, L_3). As it can be easily verified, the average position of each of the links is: $t(L_1) = 2,\ t(L_2) = 2\frac{1}{3},\ t(L_3) = 1\frac{2}{3}$. Thus the final answer set is: $(L_3,\ L_1,\ L_2)$ (again, different than in the case of the remaining two approaches).

To compute if the final answer is consistent we have to compute two distances:

1. $\hat{d} = \frac{\sum_{i,j=1,3;i\neq j} d(A_i,A_j)}{m(m+1)}$ and
2. $\hat{d} = \frac{\sum_{i=1,3} d(A_i,C)}{m}$.

When trying to establish consistency of the final answer set we find out that distances between result sets are: $d(A_1,A_2) = 2;\ d(A_2,A_3) = 2;\ d(A_1,A_3) = 2$ and thus their distance is $\hat{d} = \frac{1}{2}$. At the same time distances between result sets and the consensus answer are: $d(A_1,C) = 2,\ d(A_2,C) = 2,\ d(A_3,C) = 0$ and thus the average distance is $|\hat{d}_{min}| = \frac{4}{3}$. Comparing these results we can see that the obtained result set is not consistent. Interestingly, we have not found a consistent result set in any of our experiments.

4. System setup

Here, we only briefly describe main issues involved in implementation of the proposed system. Note that, as stated above, software agents were used because original systems that were modified, to be utilized with data generated by search engines, have been implemented using software agents. However, this method of implementation has no bearing on the results. It is thus reported here only for completeness. Separately, in all three approaches discussed here, user feedback may be utilized to rank obtained results and further improve the performance of information retrieval (as in [35,17]). However, we have decided that this issue is a complex subject on its own and requires performing multiple experiments involving human subjects, and as such falls outside of scope of current contribution. More details about implementation can be found in [34,33].

Developed system consists of two main parts: the *Client Module* (the interface) and the *Main Module*. The *Client Module* is responsible for interacting with the end-user. The *Main Module* is an application that processes user requests, which are received from the *Client Module*. Upon reception of a request, the *Main Module* performs search utilizing search engines and processes (combines) obtained results using one of the selected methods. The *Main Module* consists of two types of agents—a single *Manager Agent* (*MA*) and a set of *Search Agents* (*SA*).

The *Manager Agent* receives requests from the *Client Module* and manages agents used for information retrieval and for combining results. Note that the *Manager Agent* could also be accessed from any application (standalone/web-based) as its entry point is a separate thread in the application.

The *Manager Agent* awaits the query and the selection of the result combining algorithm (provided through the web interface). When the input is received, the *MA* creates as many *Search Agents* as there are available/selected search engines (user can specify how many/which search engines to include/exclude). Each *SA* is assigned a different search engine. *SA*s query the database using the content of the query and information about selected algorithm, to retrieve a set of weights, which are used during data processing. Weights are the ranks of search engines; computed for a given algorithm based on previous results. Their belong to the interval (0,1) and depend on how the algorithm evaluated result set of a particular search engine for a given query. If engine performed badly—results were not satisfactory; it is assigned a smaller weight than the engine results of which were considered as better ones (in the sense of the algorithm). Those weights are used during the ranking processes to "boost" URLs originating from engines, which contributed better results in previous runs for that query and algorithm. If this is the first time for a given query, all ranks are set to 1; and this is the assumption for all of experiments presented below. In other words, results presented below do not involve any additional (feedback-based) rankings except for URL orders in individual result sets.

Next, all *SA*s execute the query and return their results to the *MA*, which processes them according to the selected result-combining algorithm. When the processing is finished, the *MA* sends the final answer set to the web application to be displayed. Note that if the algorithm was able to find the best result, the result list is displayed and knowledge base is updated. The search engine which yielded the top result is ranked as the best and other engines are ranked according to how close they were to this engine. If, however, algorithm was not able to yield a "satisfactory" answer; application displays "an answer" and an prompts the user to provide feedback (subjective evaluation performed by the user). Feedback (if received) explicitly ranks search engines. In the knowledge base we store—for each query, search engine and method of answer processing, and the engine rank.

4.1. Common algorithms

There is a number of algorithms that are commonly used across the application. Here, we present only the algorithm for the initial URL ranking, which is performed before start of the *Game theory* and *Auction* methods (not the *Consensus* method), which is crucial for the operation of the system (the remaining common algorithms, as well as additional details, can be found in [34,33]). As stated above, the purpose of the initial processing and ranking algorithm is, first, to calculate weights (confidence values) that *Search Agents* associate with retrieved URLs. Furthermore, the *Game theory* and the *Auction*-based approach require that each result set contains the same URLs (in any order); or they break. Therefore, the second function of this algorithm is to extend individual result sets by adding "missing" URLs. It also determines if computational parts of these two approaches can be performed. The rule is as follows: if, for all pairs of result sets A and B, the $A \cap B = \emptyset$ then the main part of the *Game theory* and *Auction*-based method cannot start. Thus, if a result set has no common URL with any other result set it is removed from the process as being not suitable for the algorithms which require input URLs to be in every result set. The pseudo-code of the algorithm is as follows.

Input Result sets $\langle a^i, r^i \rangle$ provided by *Search Agents*—each in the form $r^i = \langle U_1^i, U_2^i, \ldots, U_n^i \rangle$, where $U_j^i, j = 1, \ldots, n$, are URLs.
Output Result sets with ranked and possibly extended sets of URLs
BEGIN

1. for each *Search Agent*:
   - check if result sets of other *Search Agents* contain any of "its"
   - construct matrix representing how many URLs of the agent are contained in the each result set of other *Search Agents*
2. check if each *Search Agent* has at least one common URL with another if not—remove it from further processing
3. if result set of every *Search Agent* are disjoint with result sets of every other *Search Agent*—stop algorithm
4. for each *Search Agent*:
   - for each URL in *Search Agent* result set:
     - rank the URL as follows:

       $$rank(U^i) = (|r| - i) \times weight(r),$$

       where i is a position of URL in r
     - find *Search Agent* which result set does not contain the URL, update their rankings:

       $$rank(U^i) = 1.0 \times weight(r)$$

       (weights calculation—listing found in [34])
5. return ranking

END

5. Experimental setup

We have performed a number of experiments using the proposed approaches. However, for the purpose of this text, we have selected a single "case study" example. Nevertheless, generalizations presented below are based on all experiments. The experimental setup was as follows:

1. In the case study, we have used three queries: *consensus decision making*, *consensus decision making for conflict solving*, and *is consensus decision making for conflict solving good enough or maybe game theory or auction is better*. The idea was to use

queries which are related to the same topic; however, one was to be simple, second complex, and third very complex, while retaining coherence.

2. We utilized five search engines. Four of them were English-language-based: *Google, Ask.com, Live, Yahoo!* and one of Polish origin—*Interia*. Note that the latter one is a "Google-based" engine. However, it very often produces results which differ from its "parent engine."
3. System was setup to return 20 unique initial results (URLs) for each query. This allowed for fast data processing, while keeping results valid for generalizations.
4. When comparing results presented by search engines and the three above described algorithms, we have first, compared top results in obtained answer sets; second, we looked into the *Set Coverage* and the *URL-to-URL* coverage. *Set Coverage* measures how many URLs from the final result set produced by an algorithm are contained in the result set returned by the search engine (regardless of the position of the URL). *URL-to-URL* measures how many URLs were at the same position in two result sets. However, these two measures were taken only for the top 10 results returned by each search engine and the algorithms.
5. Recall that in our experiments we did not utilize the user feedback—each presented result is an effect of a "fresh" query. Therefore, we have set all weights to 1 (each query was treated as a new one).
6. Furthermore, we did not take into account that for most search engines order of key words used in the query matters (here, we would like to thank Mihai Mocanu from the University of Craiova for his important remarks made during the IDC Conference). In other words, the query "*consensus decision making*," can produce different results than a query "*making consensus decision*," or the "*consensus making decision*" query. The reason for this "limitation" of our experiments was that we believe that it is rather unlikely that an individual user would know how to adjust order of words in the query to achieve "better results" (see, also [23]). Furthermore, topic of optimal use of search engines, while very interesting and considered very important (see also comments about [32] site in Section 2), is out of scope of this paper.

6. Experimental results

6.1. Comparing result sets directly

In Tables 8–10 we summarize results obtained for the three queries. Due to the lack of space, we present only five top URLs from each result set. However, the discussion that follows is based on complete result sets (and other performed experiments; see [33]).

6.2. Auction method

We can observe that, in the case of the simple query (Table 8), the Auction method as the two top URLs returned those which are also the two top-most in Google, Interia and Live. The third URL in the result set is also the third URL in the Yahoo! engine (5th in Interia and 6th in the Ask.com). However, it does not appear among top 10 results in Google and Live.

For the complex query the Auction method returned as its top-most result the URL which is present in only two input result sets (2nd in Interia and same rank in Google). However, the second returned URL was included by every search engine. Note that most of URLs included in the result set were a part of the result set produced by the Interia search engine. This shows that the Auction-based approach typically includes URLs which were not eliminated at the end of processing, rather than keeping URLs from the beginning of the process.

Finally, for the very complex query, the Auction method follows the lead of the Google search engine. It should be also noted that for all three queries some URLs present in results returned by all search engines were not selected to appear in the final answer set. However, individual rankings of these results were "spread all over" result sets of delivered by search engines. Looking at the Auction method algorithm one can realize that, overall this method favors results that appear in similar positions across many result sets. In other words, it is better to be ranked close to each other, even in a smaller number of result sets, than to be ranked "top" in one and "bottom" in another. This observation was supported by the *URL-to-URL* comparisons, where the Auction method had somewhat better results than the remaining methods.

6.3. Game theory

For the simple query, the Game theory method, as the top-most URL, returned the URL which was present in results produced by only three out of five search engines. This happened because during the game, other higher ranked URLs were discarded (keep payoffs for them were smaller than the change ones). However, the 2nd returned URL was ranked 4th in the worst case among result sets. This means that the keep payoff for this URL was high enough and it was not eliminated during the second "big round." Game theory method returned the top-most URLs from all search engines. However, some of those URLs were not ranked very high in the final result (e.g. top result from Google, Interia, Live and Yahoo! was returned only at the 7th position).

For the complex query, the Game theory method returned as its top choice a URL which was found in two out of five search engines. Similarly to the simple query, some high ranked answers were discarded at the beginning of the game

Table 8
Top five results from search engines and all three algorithms for the simple query.

	Interia	*Google*
1	http://en.wikipedia.org/wiki/Consensus_decision-making	http://en.wikipedia.org/wiki/Consensus_decision-making
2	http://en.wikipedia.org/wiki/Consensus	http://en.wikipedia.org/wiki/Consensus
3	http://www.actupny.org/documents/CDdocuments/Consensus.html	http://www.actupny.org/documents/CDdocuments/Consensus.html
4	http://www.npd-solutions.com/consensus.html	http://www.npd-solutions.com/consensus.html
5	http://www.zmag.org/forums/consenthread.htm	http://www.seedsforchange.org.uk/free/consflow.pdf
	Ask.com	*Live*
1	http://www.actupny.org/documents/CDdocuments/Consensus.html	http://en.wikipedia.org/wiki/Consensus_decision-making
2	http://www.casagordita.com/consensus.htm	http://en.wikipedia.org/wiki/Wikipedia:CON
3	http://www.npd-solutions.com/consensus.html	http://www.npd-solutions.com/consensus.html
4	http://www.welcomehome.org/rainbow/focalizers/consenseus.html	http://www.actupny.org/documents/CDdocuments/Consensus.html
5	http://www.ballfoundation.org/ei/tools/consensus.html	http://www.consensus.net/
	Yahoo	*Auction*
1	http://en.wikipedia.org/wiki/Consensus_decision-making	http://en.wikipedia.org/wiki/Consensus_decision-making
2	http://www.actupny.org/documents/CDdocuments/Consensus.html	http://en.wikipedia.org/wiki/Consensus
3	http://www.zmag.org/forums/consenthread.htm	http://www.zmag.org/forums/consenthread.htm
4	http://en.wikipedia.org/wiki/Consensus	http://www.casagordita.com/consensus.htm
5	http://www.casagordita.com/consensus.htm	http://www.welcomehome.org/rainbow/focalizers/consenseus.html
	Game theory	*Consensus*
1	http://en.wikipedia.org/wiki/Consensus	http://en.wikipedia.org/wiki/Consensus_decision-making
2	http://www.actupny.org/documents/CDdocuments/Consensus.html	http://www.actupny.org/documents/CDdocuments/Consensus.html
3	http://www.npd-solutions.com/consensus.html	http://www.npd-solutions.com/consensus.html
4	http://www.casagordita.com/consensus.htm	http://www.casagordita.com/consensus.htm
5	http://www.ballfoundation.org/ei/tools/consensus.html	http://www.ballfoundation.org/ei/tools/consensus.html

Table 9
Top five results from search engines and all three algorithms for the complex query.

	Interia	*Google*
1	http://www.npd-solutions.com/consensus.html	http://www.npd-solutions.com/consensus.html
2	http://www.actupny.org/documents/CDdocuments/Consensus.html	http://www.actupny.org/documents/CDdocuments/Consensus.html
3	http://www.managingwholes.com/–consensus.htm	http://www.managingwholes.com/–consensus.htm
4	http://www.crcvt.org/mission.html	http://www.wiley.com/WileyCDA/WileyTitle/productCd-0893842567.html
5	http://www.exedes.com/	http://www.exedes.com/
	Ask.com	*Live*
1	http://www.exedes.com/	http://www.exedes.com/main.htm
2	http://www.exedes.com/main.htm	http://www.exedes.com/
3	http://allentech.net/techstore/related_1560521996.html	http://www.hrdq.com/products/40decisionativitiesSB.htm
4	http://www.urbanministry.org/esa/maintaining-unity-decision-making-problem-solving	http://www.hrdq.com/products/25problemsolving.htm
5	http://www.sasked.gov.sk.ca/docs/native30/nt30app.html	http://www.teleometrics.com/programs/decision_making_and_consensus_building.html
	Yahoo	*Auction*
1	http://policy.rutgers.edu/CNCR/pdmcm.html	http://www.actupny.org/documents/CDdocuments/Consensus.html
2	http://www.au.af.mil/au/awc/awcgate/ndu/strat-ldr-dm/pt3ch11.html	http://www.exedes.com/
3	http://www.exedes.com/main.htm	http://www.crcvt.org/mission.html
4	http://www.exedes.com/	http://www.managingwholes.com/–consensus.htm
5	http://www.hrdq.com/products/40decisionactivitiesSB.htm	http://www.ic.org/pnp/ocac/
	Game theory	*Consensus*
1	http://www.actupny.org/documents/CDdocuments/Consensus.html	http://www.exedes.com/main.htm
2	http://www.npd-solutions.com/consensus.html	http://www.exedes.com/
3	http://allentech.net/techstore/related_1560521996.html	http://www.colorado.edu/conflict/peace/glossary.htm
4	http://www.exedes.com/main.htm	http://www.policy.rutgers.edu/CNCR/pdmcm.html
5	http://www.exedes.com/	http://www.npd-solutions.com/consensus.html

due to the keep payoff not high enough. Then, after the top-most URLs were already selected, situation has changed—keep payoffs were high enough—and some highly ranked URLs were added to the final answer at lower positions. and the same scenario repeated for the very complex query.

Table 10
Top five results from search engines and all three algorithms for the very complex query.

	Interia	*Google*
1	http://plato.stanford.edu/entries/Gametheory/	http://scholar.google.com/scholar?num=20&hl=en&ie=UTF-8&q=author:"Raiffa"intitle:"NegotiationAnalysis:TheScienceandArtof..."&um=1&oi=scholar
2	http://www.ejournal.unam.mx/cys/vol03-04/CYS03407.pdf	http://scholar.google.com/scholar?num=20&hl=en&ie=UTF-8&q=author:"Martimort"intitle:"DelegatedCommonAgencyunderMoralHazardandthe..."&um=1&oi=scholar
3	http://updatecenter.britannica.com/eb/article?articleId=109420&pid=ursd07	http://scholar.google.com/scholar?num=20&hl=en&ie=UTF-8&q=author:"Day"intitle:"EXPRESSINGPREFERENCESWITHPRICE-VECTORAGENTSIN..."&um=1&oi=scholar
4	http://www.people.hbs.edu/mbazerman/curriculum_vitae.html	http://plato.stanford.edu/entries/Gametheory/
5	http://doi.ieeecomputersociety.org/10.1109/TSE.2003.1237173	http://www.ejournal.unam.mx/cys/vol03-04/CYS03407.pdf
	Ask.com	*Live*
1	http://www.colorado.edu/conflict/peace/glossary.htm	http://www.primisonline.com/cgi-bin/POL_program.cgi?programCode=HBSNA&context=
2	http://learn.royalroads.ca/tcmacam/navpages/glossary.htm	http://lsolum.blogspot.com/archives/2005_09_01_lsolum_archive.html
3	http://dieoff.org/page163.htm	http://lsolum.blogspot.com/archives/2005_11_01_lsolum_archive.html
4	http://www.peacemakers.ca/publications/ADRdefinitions.html	http://www.cs.iit.edu/~xli/cs595-game/auction.htm
5	http://www.virtualschool.edu/mon/Economics/KOMT.html	http://www.csc.liv.ac.uk/~mjw/pubs/imas/distrib/powerpoint-slides/lecture07.ppt
	Yahoo	*Auction*
1	http://www.msu.edu/course/aec/810/studynotes.htm	http://scholar.google.com/scholar?num=20&hl=en&ie=UTF-8&q=author:"Raiffa"intitle:"NegotiationAnalysis:TheScienceandArtof..."&um=1&oi=scholarr
2	http://www.cit.gu.edu.au/~s2130677/teaching/Agents/Workshops/lecture07.pdf	http://scholar.google.com/scholar?num=20&hl=en&ie=UTF-8&q=author:"Martimort"intitle:"DelegatedCommonAgencyunderMoralHazardandthe..."&um=1&oi=scholarr
3	http://home.earthlink.net/~peter.a.taylor/manifes2.htm	http://plato.stanford.edu/entries/Gametheory/
4	http://aufrecht.org/blog/swcat/39172	http://scholar.google.com/scholar?num=20&hl=en&ie=UTF-8&q=author:"Day"intitle:"EXPRESSINGPREFERENCESWITHPRICE-VECTORAGENTSIN..."&um=1&oi=scholarr
5	http://www.concurringopinions.com/archives/economic_analysis_of_law/index.html	http://www-static.cc.gatech.edu/~jp/Papers/Zagaletal-CollaborativeGames-Lessonslearnedfromboardgames.pdf
	Game theory	*Consensus*
1	http://www.msu.edu/course/aec/810/studynotes.htm	http://plato.stanford.edu/entries/Gametheory/
2	http://scholar.google.com/scholar?num=20&hl=en&ie=UTF-8&q=author:"Raiffa"intitle:"NegotiationAnalysis:TheScienceandArtof..."&um=1&oi=scholarr	http://www.ejournal.unam.mx/cys/vol03-04/CYS03407.pdf
3	http://www.ejournal.unam.mx/cys/vol03-04/CYS03407.pdf	http://updatecenter.britannica.com/eb/article?articleId=109420&pid=ursd07
4	http://www.primisonline.com/cgi-bin/POL_program.cgi?programCode=HBSNA&context=	http://www.msu.edu/course/aec/810/studynotes.htm
5	http://lsolum.blogspot.com/archives/2005_11_01_lsolum_archive.html	http://www.people.hbs.edu/mbazerman/curriculum_vitae.html

Overall, we observe that if a URL is ranked very high in at least one input set, it will be taken into consideration, though possibly in a later stages of the algorithm. This is because, the Game theory method can be seen as a method that is "seeking variety" in the combined answer set. However, this means also that, while top results will make it to the final answer (even if they are not represented across multiple answer sets), it is difficult to predict at which position they will materialize.

6.4. Consensus method

Since the Consensus method is focused on average ranks of URLs in the input result sets, in all cases, if a URL was ranked "high enough" across all inputs, it will be at one of top-most places in the final answer. Answers that are mostly ranked low may not be included, even if they are ranked high in one or two cases. In other words, to make it to the final answer it is better to be close to the top across all inputs, rather than best in a single one. For example, for the very complex query, the three top-most results from Google did not make it to the final answer. This is because these were URLs from the Google Scholar and they have "no support" from any of the remaining search engines.

Note that *none* of the final answer sets was found to be consistent (their Levenshtein distance from the input sets was too high). At this stage it is unclear if the reason is utilization of the Levenshtein distance, or the very nature of the consistency measure as related to combining answers that are ordered links originating from multiple search engines (e.g. here, the input data could be highly dispersed—and getting more and more dispersed as more results were included in the input result sets—thus making expectation that a consistent consensus can be reached highly improbable). However, when studying content of answers sets produced by the Consensus method, they look quite reasonable (except for the fact that, for the very complex query, all three top results from Google Scholar are absent) and thus it is difficult to say if it is truly necessary for the Consensus to be consistent and what would it mean for the actual user.

6.5. Coverage-based view of results

Let us now look into two coverage measures (*set coverage* and *URL-to-URL*) for the three queries. They have been summarized in Table 11.

First, let us observe that as the query becomes more complex, the value of *set coverage* drastically decreases. This can be explained by the fact that the responses generated by various search engines have less and less in common. Therefore, regardless of the method used, the final answer set becomes a collection of links that do not repeat in other search engines, chosen from each individual answer set. This trend is even more drastic in the *URL-to-URL* comparison. Here, already for the

Table 11
Comparison of coverage.

	Ask.com (%)	Live (%)	Interia (%)	Yahoo (%)	Google (%)
Auction method, simple query					
Set coverage	60	40	70	60	70
URL-to-URL	0	10	20	30	20
Game theory method, simple query					
Set coverage	60	60	70	80	60
URL-to-URL	30	10	0	50	0
Consensus method, simple query					
Set coverage	70	50	80	70	80
URL-to-URL	20	20	10	20	10
Auction method, complex query					
Set coverage	10	10	50	10	40
URL-to-URL	0	10	0	0	0
Game theory method, complex query					
Set coverage	60	40	30	40	30
URL-to-URL	40	0	10	0	10
Consensus method, complex query					
Set coverage	50	30	70	40	60
URL-to-URL	0	20	0	0	0
Auction method, very complex query					
Set coverage	0	0	30	0	40
URL-to-URL	0	0	10	0	20
Game theory method, very complex query					
Set coverage	0	40	30	10	50
URL-to-URL	0	0	0	10	0
Consensus method, very complex query					
Set coverage	10	20	90	20	40
URL-to-URL	0	0	30	0	0

Table 12
Results for *MySpiders*.

	Simple query	*Complex query*
1	http://en.wikipedia.org/wiki/Consensus_decision-making	http://www.exedes.com/main.htm
2	http://www.welcomehome.org/rainbow/focalizers/consenseus.html	http://www.peacemakers.ca/bibliography/bib50resolution.html
3	http://www.seedsforchange.org.uk/free/consens	http://www.managingwholes.com/–consensus.htm
4	http://www.actupny.org/documents/CDdocuments/Consensus.html	http://www.managingwholes.com/glossary-p/c.htm
5	http://www.casagordita.com/consensus.htm	http://www.actupny.org/documents/CDdocuments/HistoryNV.html

intermediate query practically no URL is in the same location in the answer set as it is in any of the search engines. This indicates also that this "performance measure" is not particularly useful for the application in question (i.e. if for the very complex query 4 out of 5 *URL-to-URL* results are 0% then no actual comparison takes place). Interestingly, as explained above, the Auction method has somewhat better *URL-to-URL* coverage than the other methods.

As expected, results returned by *Interia* and *Google* are very similar, with both coverage measures randomly "favoring" either one of them. Interestingly, these two search engines have highest coverage values for the very complex query. However, this may be a result of collusion, where two similar search engines "dominate" views of the others. This observation provides also a warning, that selection of "groups of experts" has to provide as "orthogonal" view of the subject as possible. Otherwise, regardless of the method used, the returned combined answer may be dominated by a few "experts" that see the problem in a similar way.

Observed results suggest also that the Consensus method does what its name suggests—delivers response that is closest to consensus (a "holistic" answer). This can be seen particularly in the case of the very complex query, where for the Consensus method the *set coverage* is non-zero *also* for search engines other than *Interia* and *Google*.

6.6. Comparison with MySpiders

Finally, we have compared performance of the three algorithms discussed above with an academic, content-based *Myspiders* metasearch engine [6,1]. In Table 12 we report results of our findings. Note that the *MySpiders* system returned *only* five results for the complex query and *no results at all* for the very complex query.

Here, we compare results presented in Table 12 with these in Tables 8 and 9. We can see that in the case of the simple query, *MySpiders* as the top link returns the Wikipedia definition of consensus decision making. This URL is contained in all result sets of the three algorithms presented above. The second URL points to the resource which is a short description on how the consensus decision making process should look like. This URL is present in the Consensus method result set at the 7th position. The third URL is another resource where consensus decision making is broadly described. The URL pointing to this resource is also present in result set of every method (at the 9th position). In general, only one URL returned by *MySPiders* (the 10th one) was not contained in any combined result set from the three algorithms. However, this webpage is pointed to by other URLs which are contained in result sets of every tested method.

Overall, the Consensus method provided the closest view to the *MySpiders* system. Furthermore, it can be stated that *MySpiders*, which is a content-based method, does not seem to produce "better" results. Almost all of its resulting URLs were found separately in the result sets of tested methods (7 in the Consensus method and 6 in the two remaining ones).

In the case of the complex query, all URLs returned by the MySpiders system were present in at least one of result sets of the three tested methods. This could mean that no matter which of those approaches (Auction, Game theory, Consensus or MySpiders) is used for preparing the final answer, some of the links will be present in at least one of them.

Recall, that *MySpiders* returned only five results for the complex query and no response for the very complex query, which is rather disappointing.

7. Concluding remarks

Overall, on the basis of all of our experiments (also these not reported here), we can state that:

- Results delivered by the *Auction method* are highly dependent on each individual result set and do not represent well the "combined view" of all search engines. Here, the "winners" are these URL's that are ranked similarly in multiple search engines.
- The *Consensus method* returned the results which represent the "common view" of participating search engines. However, returned result sets were inconsistent(!) according to consensus theory itself. This happens due to the high "position dispersion" of URL's throughout the result sets. There are situations where a URL is, for instance, on the 1st place in one result set, on the 9th place in another result set, and on the 5th in the next. For this result, the Levenshtein distance between response-sets is relatively large and thus the final result set is inconsistent. This distance is further increasing as the number of considered answers is growing. Note also that it is very likely that this observation is independent from the "distance measure" used. This raises doubts about usefulness (importance) of the consistency measure for the application in

question. Especially, since the results returned by the method seem "quite reasonable." Finally, let us observe that the *Consensus method* can skip the best result from a single source if this is the only source that suggests such result, which seems to be somewhat strange.
- *Game theory* method seems to act in a way that positions it in between the two other approaches. On the one hand, it favors the winners. On the other, if a URL is at of top places of more than one result set, it is very likely to be incorporated into the final result set (even though it may be locate much lower than its average position). It can be thus said that, when combining individual results, it is searching for a diversity in the final answer set. Note that similarity between response-sets delivered by the *Game theory* and the *Consensus method* in the case of the simple query is specific for this example (as it disappears already for the complex and very complex queries).
- Attempts at utilization of the *MySpiders* system turned to be rather disappointing, as it returned only five responses to the complex query and no responses at all to the very complex one.

What remains open is the question: which method can be considered "the best." Unfortunately, neither looking into produced answers directly, nor utilizing the coverage based view provides an answer to this question. However, based on the observations reported above, we can see that the *Consensus method* is the most "exploratory" in nature. In other words, when someone is seeking a "balanced breadth"/"egalitarian" view, then this method has most potential. On the other hand, the *Auction method* seems to be of most use, when users seek most "stable" view of multiple results. In other words, preferred are results that are similarly ranked within multiple sources, while results that are ranked very dissimilarly will have low overall ranking. Finally, the *Game theory* seems to be most balanced of the two. On the other hand it will definitely include top-most responses, but it will also include results that appear in multiple search engines. In this sense it could be conjectured that the *Game theory*-based approach, due to its balanced way of building final response-set, could be most advantageous of them all.

However, it seems obvious that for different users, seeking different information for different purposes, different approaches may result in providing them what they are actually looking for. Therefore, extensive experimental work, involving actual users, would be necessary to address this issue.

References

[1] Center for complex networks and systems research—publications, <http://cnets.indiana.edu/groups/nan/publications>.
[2] Copernic: Search, access, connect from anywhere, <http://www.copernic.com/en/products/index.html>.
[3] The dblp computer science bibliography, <http://www.informatik.uni-trier.de/ley/db/>.
[4] Metasearch engine—wikipedia, <http://en.wikipedia.org/wiki/Metasearch_engine>.
[5] Search engine optimization, <http://seo.zunch.com/search_engine_usage_statistics.htm>.
[6] Myspiders, <http://myspiders.informatics.indiana.edu/>, December 2008.
[7] Meta-search engines, <http://www.lib.berkeley.edu/TeachingLib/Guides/Internet/MetaSearch.html>%, 2009.
[8] T. Berners-Lee, <http://dig.csail.mit.edu/2007/03/01-ushouse-future-of-the-web.html>, March 2007.
[9] R. Bhagdev, S. Chapman, F. Ciravegna, V. Lanfranchi, D. Petrelli, Hybrid search: effectively combining keywords and semantic searches, in: Proceedings of the 5th European Semantic Web Conference (ESWC 2008), 2008, pp. 723–736.
[10] S. Bloehdorn, P. Cimiano, J. Heizmann, P. Haase, J. Thurlow, J. Volkero, Ontology-based question answering for digital libraries, in: L. Kovac, N. Fuhr, C. Meghini, (Eds.), Proceedings of 11th European Conference on Research and Advanced Technology for Digital Library (ECDL 2007), LNCS, vol. 4675, Springer, 2007, pp. 14–25.
[11] A.M.P. Canuto, M. Abreu, Analyzing the benefits of using a fuzzy neuro model in the accuracy of the neurage system: an agent-based system for classification tasks, in: Proceedings of the International Joint Conference on Neural Networks, 2006, pp. 2951–2958.
[12] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, R. Harshman, Indexing by latent semantic analysis, Journal American Society for Information Science 41 (1990) 391–407.
[13] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, E. Ruppin, Placing search in context: the concept revisited, in: WWW '01: Proceedings of the 10th International Conference on World Wide Web, ACM, New York, NY, USA, 2001, pp. 406–414.
[14] S. Fox, K. Karnawat, M. Mydland, S. Dumais, T. White, Evaluating implicit measures to improve web search, ACM Transactions on Information Systems 23 (2) (2005) 147–168.
[15] J. Freyne, R. Farzan, P. Brusilovsky, B. Smyth, M. Coyle, Collecting community wisdom: integrating social search & social navigation, in: Proceedings of the International Conference on Intelligent User Interfaces (IUI 2007), 2007, pp. 52–61.
[16] A. Gajek, S. Klink, P. Reuther, B. Walter, A. Weber, Bibliographical meta search engine for the retrieval of scientific articles, in: K.L.N. Fuhr, C. Meghini, (Eds.), Proceedings of 11th European Conference on Research and Advanced Technology for Digital Library (ECDL 2007), LNCS, vol. 4675, Springer Verlag, Berlin–Heidelberg, 2007, pp. 458–461.
[17] X. Jiang, A.-H. Tan, Learning and inferencing in user ontology for personalized semantic web search, Information Sciences 179 (2000) 2794–2808.
[18] J. Kacprzyk, H. Nurmi, M. Fedrizzi, Consensus under fuzziness, in: J. Kacprzyk, H. Nurmi, M. Fedrizzi, (Eds.), International Series in Intelligent Technologies, vol. 10, 1997, pp. 223–232.
[19] M. Klusch, X. Zhing, Deployed semantic services for the common user of the web: a reality check, in: Proceedings of the IEEE International Conference on Semantic Computing 2008, IEEE, 2008, pp. 347–353.
[20] Y. Lei, V. Uren, E. Motta, Semsearch: a search engine for the semantic web, in: Proceedings of the 15th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2006), Springer Verlag, 2006, pp. 238–245.
[21] T.A. Letsche, M.W. Berry, Large-scale information retrieval with latent semantic indexing, Information Sciences 100 (1997) 105–137.
[22] V. Levenshtein, Binary codes capable of correcting deletions, insertions and reversals, Soviet Physics. Doklady 10 (8) (1966) 707–710.
[23] M. Manafy, A question of semantics, EContent 7 (2008).
[24] N. Nguyen, Consensus system for solving conflicts in distributed systems, Journal of Information Sciences (147) (2002) 91–122.
[25] N. Nguyen, Processing inconsistency of knowledge at semantic level, Journal of Universal Computer Science 11 (2) (2005) 285–302.
[26] N. Nguyen, Methods for achieving susceptibility to consensus for conflict profiles, Journal of Intelligent and Fuzzy Systems: Applications in Engineering and Technology 17 (3) (2006) 219–229.
[27] N. Nguyen, M. Ganzha, M. Paprzycki, A consensus-based approach for information retrieval in internet, Lecture Notes in Computer Science, vol. 3993, Springer, 2006, pp. 208–215.
[28] N. Nguyen, M. Małowiecki, Consistency measures for conflict profiles, in: Transactions on Rough Sets, LNCS, vol. 1, Springer, 2004, pp. 169–186.

[29] N.T. Nguyen, Consensus system for solving conflicts in distributed systems, Information Sciences 147 (2002) 91–122.
[30] L. Santana, A. Canuto, M. Abreu, Analyzing the performance of an agent-based neural system for classification tasks using data distribution among the agents, in: Proceedings of the International Joint Conference on Neural Networks, 2006, pp. 2951–2958.
[31] L. Santana, A. Canuto, J. Junior Xavier, A. Campos, A comparative analysis of data distribution methods in an agent-based neural system for classification tasks, in: Proceedings of the Sixth International Conference on Hybrid Intelligent Systems, vol. 9, 2006, p. 9.
[32] C. Sherman, Search engine watch: Metacrawlers and metasearch engines, <http://searchenginewatch.com/2156241>, March 2005.
[33] J. Stadnik, Combining Information from Multiple Internet Sources, Master's Thesis, Warsaw University of Technology, 2008.
[34] J. Stadnik, M. Ganzha, M. Paprzycki, Are many heads better than one—on combining information from multiple internet sources, in: C. Badica et al. (Eds.), Intelligent Distributed Computing, Systems and Applications, Springer, Berlin, 2008, pp. 177–186.
[35] M. Sufyan Beg, N. Ahmad, Web search enhancement by mining user actions, Information Sciences 177 (2007) 5203–5218.
[36] E. Szymanska, Personal Communication.
[37] E. Tsiporkova, V. Boeva, Multi-step ranking of alternatives in a multi-criteria and multi-expert decision making environment, Information Sciences 176 (2006) 2673–2697.
[38] H. Wang, T. Tran, C. Liu, Ce2—towards a large scale hybrid search engine with integrated ranking support, in: Proceedings of the 17th ACM Conference on Information and Knowledge Management'08), 2008, pp. 1323–1324.
[39] H. Wang, K. Zhang, Q. Liu, T. Tran, Y. Yu, Q2semantic: a lightweight keyword interface to semantic search, in: Proceedings of the 5th International Semantic Web Conference (ESWC'08), Springer Verlag, 2008, pp. 584–598.
[40] H.-T. Zhenga, B.-Y. Kanga, H.-G. Kim, An ontology-based approach to learnable focused crawling, Information Sciences 178 (2008) 4512–4522.
[41] Q. Zhou, C. Wang, M. Xiong, H. Wang, Y. Hu, Spark: adapting keyword query to semantic search, in: Proceedings International Semantic Web Conference, 2007, pp. 694–707.