DISTRIBUTED COMMUNITY COOPERATION IN MULTI AGENT FILTERING FRAMEWORK

Sahin Albayrak, Dragan Milosevic
DAI-Lab, Technical University Berlin, Germany, Salzufer 12, 10587 Berlin, Germany

Keywords: Cooperation in multi agent systems, Distributed information retrieval, Intelligent information agents, Cooperative filtering communities, Self improving cooperation, Filtering framework, Recommendation systems

Abstract: In nowadays easy to produce and publish information society, filtering services have to be able to simultaneously search in many potentially relevant distributed sources, and to autonomously combine only the best found results. Ignoring a necessity to address information retrieval tasks in a distributed manner is a major drawback for many existed search engines which try to survive the ongoing information explosion. The essence of a proposed solution for performing distributed filtering is in both installing filtering communities around information sources and setting a comprehensive cooperation mechanism, which both takes care about how promising is each particular source and tries to improve itself during a runtime. The applicability of the presented cooperation among communities is illustrated in a system serving as intelligent personal information assistant (PIA). Experimental results show that integrated cooperation mechanisms successfully eliminate long lasting filtering jobs with duration over 1000 seconds, and they do that within an acceptable decrease in feedback and precision values of only 3% and 6%, respectively.

1 INTRODUCTION

With an abundance of electronically available information, finding only a relevant one can amount to a real challenge. Existed search and retrieval engines, such as Google (Brin, 1998), Yahoo, AltaVista, and many others (Saurabh, 2001) provide more capabilities today then ever before, but the information that is potentially available from World Wide Web continues to grow exponentially (Mohammadian, 2004). There is unfortunately an open doubt that these centralised search engines will not be able to adequately respond to this information explosion in the future. A distributed knowledge discovery obviously becomes the only possible way to cope with these information overload problems.

While the needed information is usually scattered in a vast number of distributed sources, a typical user has no time to look all around, and its attention has become a precious resource (Yang, 2004). Individual inspection of each available source is impractical, and the tools to manage these sources effectively will become critical (Blake, 2001). Without an approach for identifying sources that potentially have relevant documents, information rich sources are almost useless. One obviously needs new technologies that will provide means for locating, retrieving and processing of data, being stored in numerous distributed sources. Authors' point of view is that a comprehensive cooperation among sources can be exactly one instance of a needed technology.

What is also very important, these cooperation mechanisms can make usable many filtering strategies (Balabanovic, 1997)(Boley, 1998)(Tauritz, 2002) (Delgado, 2000) (Michalewicz, 2000) (Zamir, 1997) (Oard, 1996) and data mining methods (Han, 2001) that are always more or less scalable. The design and structure of many strategies may not be appropriate at all for a very large when for instance the run times behave exponentially to underlying collection size. Two illustrative examples can be simultaneous clustering with a dynamic keyword weighting (Frigui, 2004) and a self organising map (Kohonen, 2000). The former gives remarkable good results, but unfortunately only on a collection with two thousand documents. The later is applied on 7 million short abstracts, but it takes 6 weeks on 6 processor computer to train SOM. For these two and many other strategies, it is crucial to have collections with desired properties, where they can give excellent results. The problem that arises with many collections is concerned with the means to organise them somehow, and hopefully this challenge can be address through the same cooperation mechanisms.

Albayrak S. and Milosevic D. (2005).
DISTRIBUTED COMMUNITY COOPERATION IN MULTI AGENT FILTERING FRAMEWORK.
In *Proceedings of the Seventh International Conference on Enterprise Information Systems*, pages 400-407
DOI: 10.5220/0002538004000407

To support information retrieval from the distributed sources, and to help to many filtering strategies that are only small scale applicable, distributed cooperation mechanisms are essential, as it is going to be shown in the rest of this paper, being structured as follows. The next section illustrates cooperation problems through one scenario. The core of this paper is then contained in the section, which gives principles being in the basis of the used cooperation approach that is naturally separated to estimating, dispatching, composing and adapting steps. A paper is finished with sections where implementation and experimental details are shortly presented.

2 PROBLEM DESCRIPTION

The unavoidable consequence of the nowadays overall information overload is that an open problem becomes answering where the relevant information is deployed. Internet users are forced to manually make decisions about the most promising sources for retrieving the desired information, being usually a time consuming activity. Obviously, they are wasting plenty of their valuable time when they are searching for the needed information at the wrong places. The following scenario, being separated into estimating, dispatching, composing and adapting steps, is going to illustrate challenges which any comprehensive cooperation mechanisms have to address in order to overcome the mentioned information retrieval problems.

Figure 1 gives a playing ground for a scenario, where this playing ground is composed of five different data sources DB_i, $i \in \{1,...,5\}$, around which distributed filtering communities FC_i are installed. Each and every community has one manager (M_i) agent that is responsible for every single cooperation activity. In order to enables cooperation, manager M_i is representing all other communities through their descriptions CD_j, $i \neq j$, which illustrate their underlying content. For example M_3 has CD_1, CD_2, CD_4 and CD_5 as descriptions of corresponding communities with whom cooperation is possible.

[Estimating] The scenario begins by estimating how good each community can be for a particular request. This estimation naturally assumes the additional usage of a past experience, which shows how reliable each community has been in providing the relevant results. The idea is to somehow make a balance between what a particular community says about itself and the lessons that have been learnt through the previous cooperation with it. The dilemma, which has to be addressed through the estimation step, is concerned with a deciding either to use for a particular job currently better suited communities or to rely more on communities that have better satisfied user needs in the past.

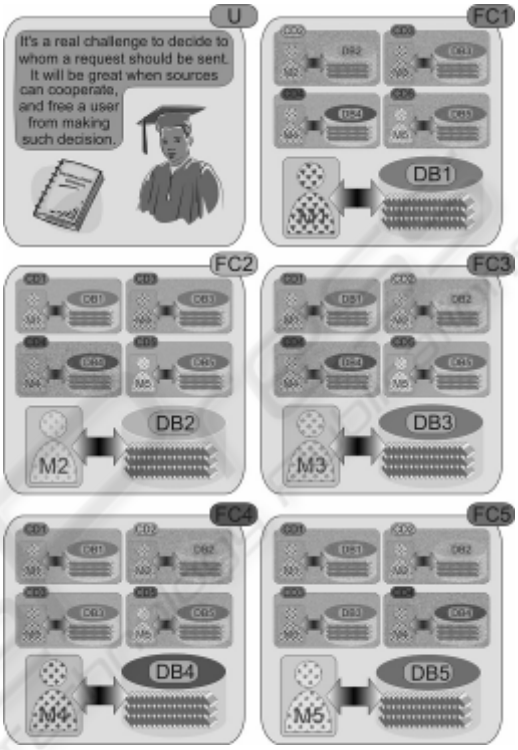

Figure 1: Cooperation playing ground

[Dispatching] By using formed estimations about how promising are communities, a decision to dispatch a job to some of them should be made. But, not all of jobs are the same, and not for all of them the same coverage concerning the found estimations exists. For some jobs, many communities can provide quite good results. On the other hand, very specific jobs can be well processed only by specialised communities, and asking others is the pure waste of resources. A decision, how many communities should contribute, is hopefully connected with how good are estimations. It is maybe reasonable not to dispatch a job to a community with a bad estimation, but it is necessary to ensure that every job will be processed at least by somebody.

[Composing] The last piece of a puzzle, known as finding the needed information through a distributed filtering, is concerned with putting all found results together. In this composing activity, being performed by the manager that was initially chosen by a user, not only the quality of results but also the community successfulness in the past should be

taken into consideration. The problem is to find out how to compare the following two results. The one was found by not so reliable community, but for that result the responsible community says that it is exactly what the user is expecting. The other is found by the community that is known as a very successful one, but at the same time that community is declaring that this particular result has weak chances to satisfy user's needs. In other words, the question is how to combine by community predicted result relevance with a reliability of that particular community.

[Adapting] As soon as user feedback about the real result relevance is received, the measure of community successfulness in finding accurate results can be adapted. The ultimate goal of these adaptation activities, is establishing a more realistic picture about the potentials of available communities. The final effect is that a system is hopefully going to learn to even better do cooperation among communities for future filtering jobs.

3 APPROACH

The authors' point of view on satisfying information needs is concerned with the fact that the asked information is usually scattered around many different distributed sources, and that a real challenge becomes both finding which sources should be searched for a particular request and putting together found results. These challenges are addressed through the installation of at least one so-called filtering community around each and every information source and by setting up sophisticated cooperation mechanisms between communities.

System architecture is given on Figure 2. *User agent* (U) is responsible for the creation of filtering jobs by collecting the user preferences. It also knows how a user feedback can be obtained and forwarded to a manager agent. *Manager agent* (M) is the cornerstone that fulfils all cooperation activities and ensures the satisfied quality of filtering services. It should be seen as the entity that first performs the estimation of sources in order to be able to dispatch the received filtering job to the right communities. As soon as the activated communities have produced results, manager will then compose the final result set that will be returned back to the user agent. In the case of receiving any feedback from the user agent about the result relevance, manager agent will perform the adaptation of knowledge that it has about the responsible communities.

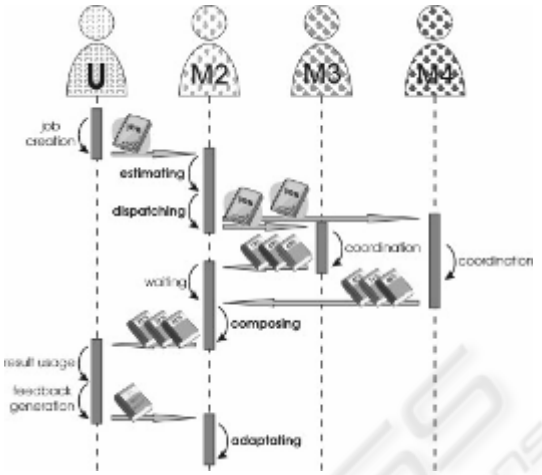

Figure 2: Estimating, dispatching, composing and adapting cooperation steps

The ultimate goal of the deployed cooperation mechanisms is always to find which communities are the most promising for providing results to the received request. This can be achieved through estimating communities, dispatching request, composing results and adapting reliability steps, as it is going to be shown in the following sub-sections.

In order to make the ongoing discussions more understandable, terms filtering community, coordination, cooperation, filtering request and community description will be defined as follows:

Def. 1. *Filtering community FC* is a collection of many different filtering agents that are tailored to efficiently do searching on the underling collection of objects. Instead of having only filtering agents, each and every community has also one so-called manager agent that is mainly responsible for performing coordination and cooperation tasks.

Def. 2. *Coordination* is a comprehensive activity, being performed by a manager agent with an ultimate goal to find which filtering agent from its community is the most promising for doing filtering in a current system runtime situation. The way how one such coordination can be achieved is out of scope of this paper, and it is addressed in (Albayrak, 2004b)(Albayrak, 2004c).

Def. 3. *Cooperation* is performed between manager agents in order to find the most competent communities for processing the received request. This paper is focused on showing how that can be achieved.

Def. 4. *Filtering request FR* describes user preferences towards the documents that will satisfy the imposed information needs in the best manner. Each filtering request can be formally represented as the collection of pairs $\{t_i, \omega_i^{(r)}\}$, where t_i corresponds to

a term in a request, and $\omega_i^{(r)}$ represents the importance of the corresponding term t_i. While larger positive value of $\omega_i^{(r)}$ means that it is more important that t_i is found, larger negative value of $\omega_i^{(r)}$ assumes that it is more important that t_i is not present in the analysed document.

Def. 5. *Community description CD* represents corresponding community in concise and descriptive way both to ensure an efficient exchange of descriptions between communities and at the same time to provide enough information to other communities that want to cooperate with it. Formally, community description CD is $\{CCD, r_C\}$, where CCD is a description of the underlying collection of documents and r_C is community reliability in processing filtering tasks in the past. Community content description CCD is a collection of given number of the most important pairs $\{t_i, \omega_i^{(d)}\}$, where t_i corresponds to a particular term in a collection, and $\omega_i^{(d)}$ is the number of documents from the underlying collection that have term t_i.

3.1 Estimating Communities

The usage of the appropriate distance function is a critical point in estimating how successful each filtering community can be in processing the actual request. Because both filtering request and community content description are defined in highly sparse space, the appropriate modification of the weighted Jaccard index (Michalewicz, 2000)(Han, 2001) is expected to give the distance function with a desired behaviour. Formally, filtering request $FR(\{t_i, \omega_i^{(r)}\})$ and community description $CCD(\{t_i, \omega_i^{(d)}\})$ can be compared as:

$$d_{WJ}(FR, CCD) = \frac{\sum_{i \in both\ FR\ and\ CCD} \frac{2}{\pi} arctg(\beta \omega_i^{(d)}) \omega_i^{(r)}}{\sum_{i \in FR} \left| \omega_i^{(r)} \right|}.$$

While weights $\omega_i^{(d)}$, being present in a community description, always take only positive values, weights $\omega_i^{(r)}$ from a request can be negative. Such negative $\omega_i^{(r)}$ are bound to unwanted terms, and always when unwanted term from a request is present in a community description, that community pays a penalty. The penalty is larger when unwanted term is more important in a community description, i.e. d_{WJ} will be reduced more when $\omega_i^{(d)}$ is larger and at the same time $\omega_i^{(r)} < 0$.

The introduced penalties should facilitate the selection of more specialised communities, always when such communities are available. In the case where unwanted terms are present in a particular community description, the probability that good recommendations will be found by such community is small. This holds because of the assumption that community description contains the most representative terms from the underlying document collection. The *arctg* function is used to bound values for $\omega_i^{(d)}$ always to $[0, \pi/2]$, where a tuning parameter β controls the level of translation. Larger β means, that value $\pi/2$ is reached faster, and reverse. Such bounding is necessary because of a $\omega_i^{(d)}$ nature of representing a number of documents that have corresponding term t_i, that can be a ordinary big number.

A denominator in d_{WJ} expression ensures that $d_{WJ} \in [-1,1]$, where larger d_{WJ} means that particular community has more promising content for processing particular request. It is expected that communities, having in their content descriptions a lot of wanted terms and omitting unwanted ones, will have larger d_{WJ} values.

3.2 Dispatching Request

The main cooperation objective is to dispatch the actual request only to the potentially good filtering communities, being the ones that both have access to the most relevant needed information and at the same time have acceptable reliability in processing past filtering tasks. While the idea about which communities have the best underlying documents can be assessed through the usage of in the previous step found d_{WJ} values, each community description CD has the reliability r_C, showing how good the corresponding community has performed in the past.

The used dispatching principle is to send the current request to k communities, which have the best combination of d_{WJ} and r_C values. One simple solution of combining d_{WJ} and r_C values, is to define so-called community promise-ness p_C as:

$$p_C = \frac{\beta_d d_{WJ} + \beta_r r_C}{\beta_d + \beta_r},$$

where tuning parameters β_d and β_r control the influence of d_{WJ} and r_C in making a final judgement about how promising is a particular community. According to the fact that d_{WJ} and r_C take values from $[-1,1]$ and $[0,1]$, respectively, it follows that $p_C \in [-1,1]$, where bigger p_C means that a particular community is more promising.

The unnecessary loading of not promising enough communities is avoided by additionally requesting that each community, being a candidate for activation, has to have at least given minimal promise-ness. More precisely, the number of communities that will receive a dispatched request is flexible, and it is limited with $k_{\min}$ and $k_{\max}$, i.e. $k_{\min} \le k \le k_{\max}$. While in the case where many promising communities exist, at most $k_{\max}$ will be activated, in a opposite case at least $k_{\min}$ communities will ensure that at least somebody will try to find recommendations.

3.3 Composing Results

The fact, that multiple communities work together, comes to the point when the results, being produced by different communities, should be put together in order to create the unique set of recommendations. A community, having received the original request and having initiated cooperation, will collect all the results, being found by different communities, and will decide which results are good enough to be returned as recommendations. In that way the performed cooperation is completely transparent for the sender of a filtering job, i.e. user agent does not think about where the retrieved data were deployed.

It is assumed that each result comes together with the predicted quality q_p, showing how good is a particular result, and being a number from $[0,100]\%$. These qualities are set by communities that have found corresponding results, and consequently the community that is putting results together does not have any influence on them. In order to protect from the malicious communities, saying that their results are always the perfect ones, the community reliability r_C is also taken into account when composing results. Instead of ranking results based only on their quality q_p, a product $r_C q_p$ is used to better assess the real quality, and the asked number of results with the largest $r_C q_p$ values will be chosen. Results, being found by communities with low reliability, will actually pay penalties and reduce their chances to be included in a final recommendation set.

3.4 Reliability Adaptation

After completing a recommendation set and receiving a user feedback about the actual relevance of found results, learning through reliability value adaptation takes place in order to ensure that the assigned reliability value reflects as accurate as possible corresponding filtering community ability to satisfy the imposed information needs. The adaptation of r_C is based on the comparison between by filtering community predicted result relevance q_p and the actual relevance q_a, being generated from a user feedback (Figure 3). The used adaptation rule can be expressed as

$$\Delta r_C = \gamma_C l(t)(\varepsilon - |q_a - q_p|)^{2k+1}$$

In the last expression ε is a tolerance, which defines how close the predicted relevance of results should be to the actual relevance in order to reward the responsible filtering community, k $(k > 0)$ increases the influence of large q_a deviations from q_p, γ_C is a tuning parameter, and $l(t) = l_0 e^{-\gamma_t t}$ is a decreasing learning rate which insures that already learnt reliability value will not be easily destroyed. The filtering communities with the solid history will, according to such learning rate, change their reliability values in smaller extent than the novel ones.

Figure 3: Adaptation of community reliability r_C

While a reward is limited to $\gamma_C l_0 \varepsilon^{2k+1}$, a penalty for really bad estimations of q_p, being quite different from q_a, is theoretically unlimited. In reality, penalties are also limited because q_a and q_p take values from $[0,100]\%$, which gives $|q_a - q_p| \le 100\%$.

4 IMPLEMENTATION

Filtering communities are implemented in JIAC IV (Java Intelligent Agent Component-ware, Version IV) being a comprehensive service-ware framework for developing and deploying agent systems covering design methodology and tools, agent languages and architecture, a FIPA compliant infrastructure,

management and security functionality, and the generic scheme for the user access (Fricke, 2001). The agent system JIAC IV is based on CASA (Component Architecture for Service Agents) having a modular internal structure consisting of an open set of components that can be adjusted to different requirements at the design-time as well at the run-time (Sesseler, 2002). This flexibility facilitates not only the easy integration of new agents inside filtering communities, but also enables experiments with different cooperation mechanisms.

In order to estimate in a real time conditions both the flexibility of JIAC IV service framework and the applicability of a self adapting cooperation among filtering communities, comprehensive personal information assistant (PIA) has been developed. A typical user is guided by PIA starting from a collection of both information interests and delivery preferences to a relevant article presentation and feedback gathering. The authors' believe is that PIA is going to demonstrate in real conditions the applicability of both agent-based filtering systems and cooperation mechanisms being situation aware.

5 EXPERIMENTAL RESULTS

The expected benefit of the presented cooperation approach should be found first in making possible to install cooperative communities around many small sub-domains and then in using many wonderful filtering strategies, which are only small-scale applicable. Such a distributed system should manage to retrieve data almost as good as the centralised one, however together with the great reduction of filtering time. While PIA system is going to be used for making comparisons in response time and user feedback domains, one small simulated environment will give the comparisons of precision and recall values.

5.1 Long Lasting Job Elimination

A trial to escape long lasting filtering jobs, being usually a consequence of the small scale strategy applicability inside a single community, was a main motivation for the realisation of multiple communities that are mutually organised through the presented self adapting cooperation. Even though these long lasting jobs will probably produce perfect results in next few hours, to obtain nearly perfect results within few minutes is usually much more appropriate. Because a final judgement, concerning such trade-off statements, is always given by the user, this section gives comparisons between PIA, based on a centralised single community, and being based on multiple cooperative communities, in both user feedback and response time domains. As a perfect test environment, PIA system has been chosen mostly because it currently supports more than 120 different web sources, grabs daily around 3 thousand new semi-structured and unstructured documents, has almost 500 thousand already pre-processed articles, and actively helps to 26 workers from authors' laboratory in their information retrieval activities.

Before the 18th of July 2004, PIA was working without cooperative communities, and the 37 last received feedback and corresponding response time values are given on Figure 4 and Figure 5.

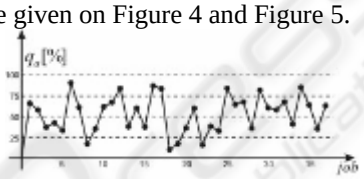

Figure 4: Feedback without community cooperation

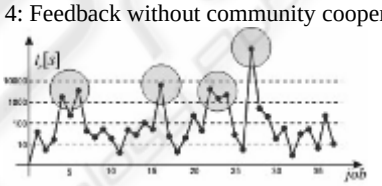

Figure 5: Response time without community cooperation

After the 18th of July 2004, PIA is working with five cooperative communities and in the first ten days feedback values were received for 37 jobs. Figure 6 and Figure 7 respectively present these feedback and corresponding response time values.

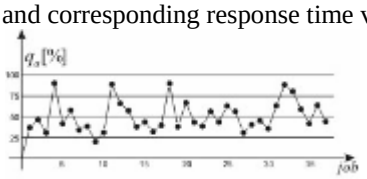

Figure 6: Feedback with community cooperation

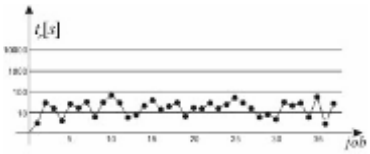

Figure 7: Response time with community cooperation

The given figures clearly show that while the integration of multiple cooperative communities does not significantly affect user feedback (Figure 4 and Figure 6 show only a slight feedback value decrease that is hopefully within 3%), it successfully eliminates long lasting jobs (7 problematic long lasting jobs marked with circles on Figure 5, having a response time that is longer than 1000s, do not occur anymore on Figure 7). While an even bigger fluctuation in quality was not detected by users probably

because they were satisfied with a shorter waiting time, what is more important, by integrating cooperative multiple communities, PIA can provide filtering services on significantly larger collections.

5.2 Precision Versus Recall

The ability of the presented cooperation mechanisms to successfully find communities, which will provide the most relevant results for the actual request, can be best assessed by using precision (p), recall (r) and fallout (f) values (Han, 2001), being broadly accepted measures for the comparison of different algorithms in the area of information retrieval. On the one hand, both for precision, being the proportion of retrieved documents that are relevant, and for recall, representing the proportion of relevant documents that are retrieved, greater values correspond to a system with better properties. On the other hand, fallout relates to the proportion of irrelevant documents that are retrieved, and consequently smaller values are preferred. Formally, precision, recall and fallout measures are defined as $p = n_r^{(r)}/(n_r^{(r)} + n_{ir}^{(r)})$, $r = n_r^{(r)}/(n_r^{(r)} + n_r^{(nr)})$, $f = n_{ir}^{(r)}/(n_{ir}^{(r)} + n_{ir}^{(nr)})$, where $n_r^{(r)}$, $n_{ir}^{(r)}$, $n_r^{(nr)}$ and $n_{ir}^{(nr)}$ correspond to either retrieved $n^{(r)}$ or not retrieved $n^{(nr)}$ documents that are either relevant n_r or irrelevant n_{ir} as in Table 1.

Table 1: $n_r^{(r)}$, $n_{ir}^{(r)}$, $n_r^{(nr)}$ and $n_{ir}^{(nr)}$ definitions

	relevant	irrelevant
retrieved	$n_r^{(r)}$	$n_{ir}^{(r)}$
not retrieved	$n_r^{(nr)}$	$n_{ir}^{(nr)}$

To compare the information retrieval system based on a centralised single community with the one that has multiple cooperative communities, a small controlled domain $D_{all}^{(500)}$ with only 500 documents is formed. For each of 10 testing filtering requests $\{FR_i\}_{i=1}^{10}$ at least 10 and at most 20 documents are manually selected as the relevant ones. In the first scenario, requests $\{FR_i\}_{i=1}^{10}$ will be resolved by using a system that has only one centralised community. That centralised community will use as the underlying document collection $D_{all}^{(500)}$ and will return 10 results for each request from $\{FR_i\}_{i=1}^{10}$. The precision, recall and fallout values, corresponding to such centralised system, are given in Table 2.

Table 2: Precision, recall and fallout for the system with single centralised community for each of 10 request from $\{FR_i\}_{i=1}^{10}$, where the last row gives the average values

$n_r^{(r)}$	$n_r^{(nr)}$	$n_{ir}^{(r)}$	$n_{ir}^{(nr)}$	p	r	f
8	12	2	478	80	40	0.42
7	11	3	479	70	38.89	0.62
7	5	3	485	70	58.33	0.61
8	7	2	483	80	53.33	0.41
5	11	5	479	50	31.25	1.03
7	5	3	485	70	58.33	0.61
9	8	1	482	90	52.94	0.20
6	4	4	486	60	60	0.82
8	6	2	484	80	57.14	0.41
7	6	3	484	70	53.84	0.62
7.2	7.5	2.8	482.5	72	50.41	0.58

In the second scenario, domain $D_{all}^{(500)}$ is manually split on agent technology $D_{at}^{(108)}$, telecommunication $D_t^{(83)}$, renewable energy $D_{re}^{(147)}$, sport news $D_{sn}^{(87)}$ and political news $D_{pn}^{(75)}$ domains, where the index in exponent says how many documents belong to the particular sub-domain. Around every sub-domain a separate community is installed, and the same 10 filtering request $\{FR_i\}_{i=1}^{10}$ from the first scenario are again resolved by using such a system with 5 distributed communities. The obtained precision, recall and fallout values are given in Table 3.

Table 3: Precision, recall and fallout for the system with 5 distributed communities for each of 10 request from $\{FR_i\}_{i=1}^{10}$, where the last row gives the average values

$n_r^{(r)}$	$n_r^{(nr)}$	$n_{ir}^{(r)}$	$n_{ir}^{(nr)}$	p	r	f
7	13	3	477	70	35	0.63
7	11	3	479	70	38.89	0.62
6	6	4	484	60	50	0.82
6	9	4	481	60	40	0.82
5	11	5	479	50	31.25	1.03
7	5	3	485	70	58.33	0.61
8	9	2	481	80	47.06	0.41
6	4	4	486	60	60	0.82
7	7	3	483	70	50	0.62
7	6	3	484	70	53.85	0.62
6.6	8.1	3.4	481.9	66	46.44	0.70

As it has been expected, the centralised system has slightly better precision, recall and fallout values than the distributed one. While the average precision and recall values have decreased for 6% and 4%, respectively, the average fallout value has increased for 0.13%. This is unfortunately the price that has to

be paid for the great reduction of a response time, which has been reported in the previous sub-section. This much shorted duration of filtering can be also the only possible explanation that the received feedback value has been reduced for only 3%, which is two times less that is the reduction of a precision.

6 CONCLUSION

The goal of this paper was to provide solutions to the challenges in filtering community cooperation, being unavoidable in nowadays rich information society. This was achieved through methods being able both to determine how promising is each available community for a particular request and to compose the final recommendation set by choosing the best from the found results.

Even though the first solutions for describing information being stored at each community are given, a future work will be focused on the usage of shared ontologies for taking care about very diverse semantics that the same word has in different communities. The price, being paid in a user feedback, precision and recall domains, will be tried to be reduced also through the application of specialised strategies for keeping updated the content descriptions of distributed, heterogeneous and dynamic filtering communities. As soon as it becomes unfeasible that each and every community has descriptions of all others, the Time-To-Live parameter will be assigned to every filtering job, which will enable their further propagation, being the basic idea behind all P2P data sharing systems. Even though given results are just the initial step towards intelligent cooperation in a multi agent framework, authors' hope is that the deployed cooperation lays the foundation for the provision of sophisticated filtering services.

REFERENCES

Albayrak, S., Milosevic, D., 2004a. Generic Intelligent Personal Information Agent. *IPSI-2004S*. Serbia.

Albayrak, S., Milosevic, D., 2004b. Self Improving Coordination in Multi Agent Filtering Framework. *IEEE, WIC and ACM Conference on IAT and WI*. China.

Albayrak, S., Milosevic D., 2004c. Situation-aware Coordination in Multi Agent Filtering Framework. *The 19th Symposium on Computer and Information Sciences (ISCIS 04)*. Antalya. Turkey. pp 480-493.

Balabanovic, M., Shoham Y., 1997. Fab: Content-Based Collaborative Recommendation. *Communication of the ACM*. 40(3). pp. 66-72.

Blake, C., Pratt W., 2001. Better Rules, Fewer Features: A Semantic Approach to Selecting Features from Text. *Proceedings of the IEEE International Conference on Data Mining*. San Jose. California. pp. 59-66.

Boley, D., 1998. Principal Direction Divisive Partitioning. *Data Mining and Knowledge Discovery*. 2(4).

Brin, S., Page L., 1998. The anatomy of a large-scale hyper textual (Web) search engine. *7th WWW Conference on Computer Networks*. 30(1-7). pp. 107-117.

Delgado, J., 2000. *Agent based information filtering and recommender systems*. Ph.D. Thesis, Nagoya Institute.

Faloutsos, C., Oard, D., 1995. *A Survey of Information Retrieval and Filtering Methods*. Technical Report CS-TR-3514. University of Maryland.

Fricke, S., Bsufka, K., Keiser, J., Schmidt, T., Sesseler, R., Albayrak, S., 2001. Agent-based Telematic Services and Telecom Applications. *Comm. of ACM*. pp. 43-48.

Frigui, H., Nasraoui, O., 2004. Simultaneous clustering and dynamic keyword weighting for text documents. *Survey of Text Mining Clustering, Classification, and Retrieval*. Springer-Verlag. pp. 45-72.

Han, J., Kamber, M., 2001. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers.

Kohonen, T., Kaski, S., Lagus, K., Salojärvi, J., Honkela, J., Paatero, V., Saarela, A., 2000. Self organization of massive document collection. *IEEE Transactions on Neural Networks*. 11(3). pp. 574-585.

Michalewicz, Z., Fogel, D., 2000. *How to Solve It: Modern Heuristics*, Springer-Verlag New York.

Mohammadian, M., Jentzsch, R., 2004. Computational Intelligence Techniques Driven Intelligent Agents for Web Data Mining and Information Retrieval. *Intelligent Agents for Data Mining and Information Retrieval*. Idea Group Publishing. pp. 15-29.

Oard, D., Marchionini, G., 1996. *A Conceptual Framework for Text Filtering*. MD 20742. Maryland.

Saurabh, S., Eui-Hong, H., Vipin, K., 2001. *Web Search Engine Technologies: A Survey*. Minnesota. USA.

Sesseler, R., Albayrak, S., 2002. JIAC IV – An Open, Scalable Agent Architecture for Telecommunications Applications, *1st International NAISO Congress on Autonomous Intelligent Systems*, Geelong. Australia.

Sheth, B. D,, 1994. *A Learning Approach to Personalized Information Filtering*. M.S. Thesis. MIT- EECS. USA.

Tauritz, D., 2002. *Adaptive Information Filtering: concepts and algorithms*. Ph.D. Thesis. Leiden University.

Yang, H., Zhang, M., 2004. Potential Cases, Database Types, and Selection Methodologies for Searching Distributed Text Databases. *Intelligent Agents for Data Mining and Information Retrieval*. IG. pp. 1-14.

Zamir, O., Etzioni, O., Madani, O., Karp, R., 1997. Fast and Intuitive Clustering of Web Documents. *Knowledge Discovery and Data Mining (KDD 1997)*. pp. 287-290.