Experimental Investigation of Two Rank Refinement Strategies for Voting with Multiple Experts

Seppo Puuronen[1], Vagan Terziyan[2], Boris Omelaenko[2]

[1] *University of Jyvaskyla, P.O.Box 35, FIN-40351 Jyvaskyla, Finland*
sepi@jytko.jyu.fi

[2] *Kharkov State Technical University of Radioelectronics, 14 Lenin Avenue,*
310166 Kharkov, Ukraine
vagan@kture.cit-ua.net; vagan@jytko.jyu.fi

The paper presents a part of research, aimed to develop voting-type multi-expert knowledge acquisition system, able to work with large group of distributed experts. Flexible model for multi-expert knowledge acquisition is developed. Four strategies determine voting process behavior. We experimentally investigate how does one of them, rank refinement strategy, influence voting process and expertise results.

Introduction

The area of knowledge management includes the problem of eliciting expertise from more than one expert. The significance of this subject deals with fast development of telecommunications, Internet, WWW that connects people together and gives possibilities to collect knowledge from different remote sources.

Could the overlapping knowledge from multiple sources be described in such a way that it is context or even process independent? In [8], the negative answer was given. If more than one expert is used, one must either select the opinion of the best expert or pool experts' judgements [8, 3]. It is assumed that when experts' judgements are pooled, collectively they offer sufficient cues to lead to the building of a comprehensive theory.

In practice, one of the following three strategies may be used for knowledge acquisition: use opinion of only one expert; collect opinions of multiple experts, but use them one at a time, or integrate these opinions. Research described in [6] deals mainly with the strategy of integrating opinions. It is assumed that acquired knowledge has more validity if it forms a consensus among the experts. In paper [5], five techniques are discussed and compared for aggregating expertise. In this study, elicited knowledge is aggregated using classical statistical methods (regression and discriminant analysis), the ID3 pattern classification method, the k-NN technique, and neural networks. In aggregating knowledge, the authors try to identify the significance of each extracted factor and the functional inter-relationship among the relevant factors.

The logic for reasoning with inconsistent knowledge has been described in [7]. This logic suits reasoning with knowledge coming from different and not fully reliable knowledge sources. Inconsistency may be resolved by considering the reliability of used knowledge sources. The reliability relation can be interpreted as follows: if two premisses are involved in a conflict the least reliable premiss has the highest probability of being wrong.

The problems, how to collect different opinions, handle inconsistent and incomplete knowledge taken from them, find consensus, support interface between individual and collective knowledge, are discussed in [4].

Present paper continues research [4]. We present a flexible model for voting-type techniques to work within. Model is presented in Basic concepts section. Four strategies determine techniques behavior and course it's flexibility: strategy of deriving opinion, most supported by experts, strategy of it's quality evaluation, voting strategy and experts rank refinement strategy. All of them are briefly described in Developing techniques section.

We fix first three strategies to investigate how different rank refinement strategies influence voting process and expertise results. For this we proceed real-life experts opinions with two rank refinement strategies, as presented in Experimental investigation section. Rank refinement strategies are compared and paper results are discussed in Conclusions.

1. Basic Concepts

In this chapter, we define basic concepts of the method. We define knowledge about multi-expert knowledge acquisition as sixth <*S*,*D*,*Q*,*V*,*P*,*T*>. The concepts used are the following:

$S = \{S_1, S_2, ..., S_n\}$ — the set of *n* knowledge sources or experts;

$D = \{D_1, D_2, \ldots, D_d\}$ — the set of *d* domain concepts;

$Q = (Q_1, Q_2, \ldots, Q_q)$ — the set of *q* problems, or questions, asked to experts;

$V = (V_1, V_2, \ldots, V_q)$ — the set of *q* solutions, or answers on the correspondent questions from the set *Q*;

P — semantic predicate, which defines piece of knowledge about domain;

$T = \{T_1, T_2, \ldots, T_t\}$ — the set of *t* techniques to proceed expert opinions;

Every concept is described below.

2. Short Description of Basic Concepts

We introduce the set *S* of *n* experts or knowledge sources and assign a numeral rank to each expert to measure expert's authority in domain with the set $r = \{r_1, r_2, \ldots, r_n\}$ of *n* expert's ranks. Rank is a subject to change during voting process. Rank is the only parameter to evaluate expert's authority in domain and the world.

We define domain as the set *D* of domain concepts (domain relations) D_i: $D = \{D_1, D_2, \ldots, D_d\}$. Domain is structured, and each domain concept consists of *m* component relations C_i:

$$D_i = (C_1, C_2, ..., C_m) .$$

Each component takes it's values from corresponding set *E*: $C_i \in E_i$, discrete or continuos.

Usual domains have restrictions on validity of combinations $D_i = (C_1, C_2, \ldots, C_m)$. We define predicate *D* to describe these restrictions, as follows:

$$D(C_1, C_2, \ldots, C_m) = \begin{cases} 1, \ \text{if combination } (C_1, C_2, \ldots, C_m) \text{ is valid in domain } D; \\ 0, \ \text{otherwise.} \end{cases}$$

Ordered set *Q* of tasks or questions, contains all tasks to solve or questions to ask during the expertise. Each task $Q_i \in Q$ is a problem, presented somehow. This may be verbal or visual presentation, etc. It is possible, that answering some questions will force new ones to appear, and new items to add to set *Q*.

The set *V* of solutions or answers corresponds to the set of tasks. Initially, it contains undefined solutions (answers), with the meaning «no solution found». Each solution $V_i \in V$ must be filled with only one opinion, constructed with technique *T* on the basis of *n* expert opinions. We call this opinion as the most supported opinion (MSUP). The resulting opinion must belong to the set of all possible domain concepts: $V_i \in D$. Technique proceeds expert opinions on tasks one-by-one to fill corresponding solutions.

We use *P* — a semantic predicate which defines piece of expert knowledge about temporal relationships in domain by the following relation between the sets *Q*, *D* and *S:*

$$P(Q_i, D_j, S_k) = \begin{cases} 1, \ \textit{if the knowledge source } S_k \textit{ uses } D \textit{ to solve task (answer question) } Q_i; \\ 0, \ \textit{otherwise.} \end{cases}$$

Definition of this predicate is updated just after every expert vote, and it presents all our knowledge about domain and the world.

Every technique $T_i \in T$ is a fourth, as follows:

$$T_i = \langle MS, QS, RS, VS \rangle, \ T_i \in T$$

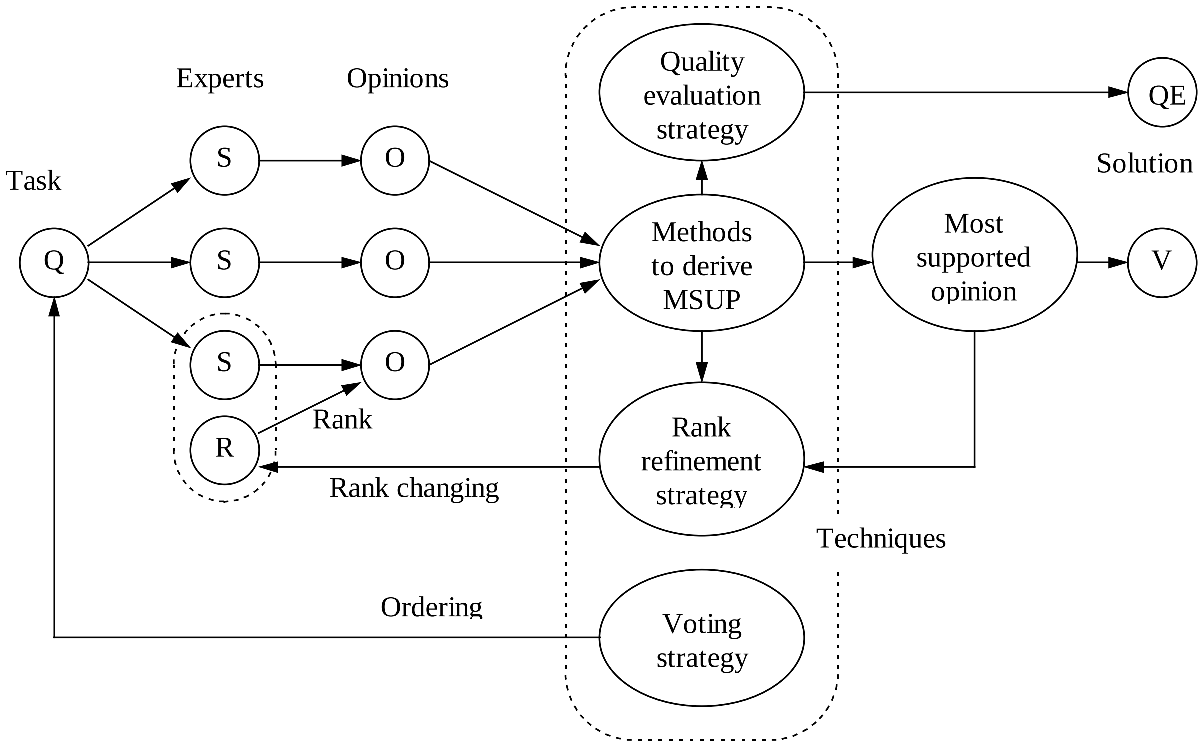


Fig. 1. Voting process scheme

Four strategies determine technique behavior: the most supported opinion deriving strategy *MS* defines the way to elicit common knowledge from the set of knowledge sources. The most supported opinion's quality evaluation strategy *QS* defines the way to understand, whether derived common knowledge corresponds to real situation. Rank refinement strategy *RS* defines a way to change expert ranks (to prize or punish experts), according to their opinions, ranks and context. Voting strategy *VS* defines the way of asking questions or giving tasks to experts.

Scheme of voting process is presented in Fig. 1. This scheme illustrates interactions between parts of the model, while processing one question or task. One of experts is shown in details, with his rank and rank changing.

The problem of selecting appropriate technique from the set of all possible ones is very important. An algorithm for such a selection will be developed in the future.

In present paper one MSUP deriving strategy is introduced in chapter 4, one quality evaluation strategy is shown in chapter 4.2, one rank refining strategy is presented in chapter 4.3, and three voting strategies are described in chapter 0. First three strategies are based on the technique, described in paper [4].

3. Allen Domain of Temporal Intervals

We use domain of Allen's relations between temporal intervals for illustrations. This domain is evidently structured, has restrictions on component combinations; it also has numerous practical implementations.

Component relations for domain concepts are as follows. We define $M = \{C_1, C_2, \ldots, C_m\}$ — the set of *m* (*m*=12) basic binary relations for temporal points, as shown in Table 1. In this Table, X^S is starting point and X^F is end point of temporal interval T_1; Y^S and Y^F are endpoints of temporal interval T_2. Component relations are binary, hence $E_i = \{1,0\}$, $i = \overline{1,m}$.

Table 1. Set M of basic endpoints' relations

M					
Triad 1: {1,2,3}			*Triad 2: {4,5,6}*		
C_1	C_2	C_3	C_4	C_5	C_6
$X^S<Y^S$	$X^S>Y^S$	$X^S=Y^S$	$X^S<Y^F$	$X^S>Y^F$	$X^S=Y^F$
Triad 3: {7,8,9}			*Triad 4: {10,11,12}*		
C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
$X^F<Y^S$	$X^F>Y^S$	$X^F=Y^S$	$X^F<Y^F$	$X^F>Y^F$	$X^F=Y^F$

We define domain *D* as the set of 13 (*d*=13) basic relations for temporal intervals. Table 2 contains the definition of predicate *D* and the correspondence between values $(C_1, C_2, \ldots, C_m)$ and domain concepts D_i, $i = \overline{1,13}$, listed in the first column.

We define domain *D* according to Allen [1,2]. The set of 13 Allen's interval relations is shown graphically in Fig. 2.

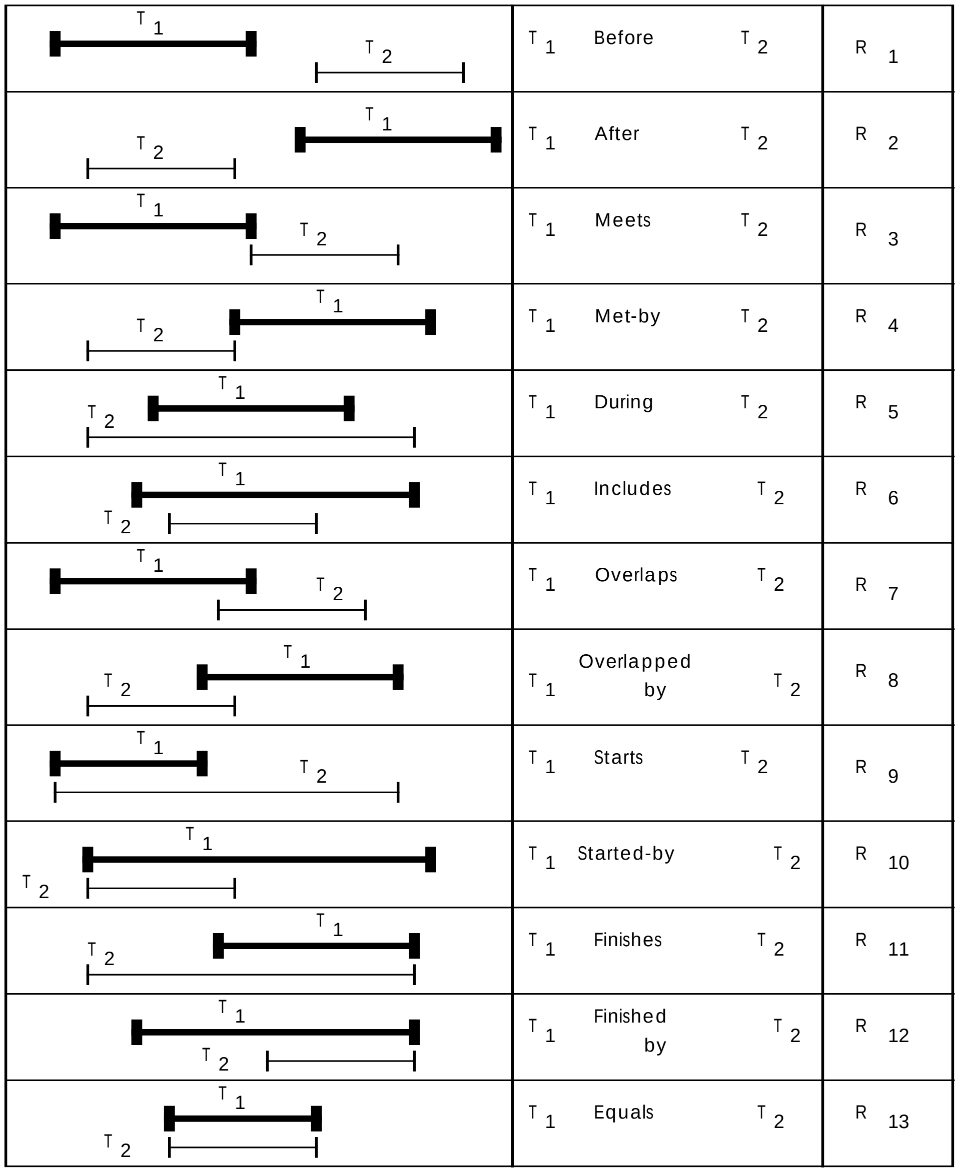


Fig. 2. The set R of Allen's basic temporal relations

Table 2. Definition of predicate D

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
D_1	1	0	0	1	0	0	1	0	0	1	0	0
D_2	0	1	0	0	1	0	0	1	0	0	1	0
D_3	1	0	0	1	0	0	0	0	1	1	0	0
D_4	0	1	0	0	0	1	0	1	0	0	1	0
D_5	0	1	0	1	0	0	0	1	0	1	0	0
D_6	1	0	0	1	0	0	0	1	0	0	1	0
D_7	1	0	0	1	0	0	0	1	0	1	0	0
D_8	0	1	0	1	0	0	0	1	0	0	1	0
D_9	0	0	1	1	0	0	0	1	0	1	0	0
D_{10}	0	0	1	1	0	0	0	1	0	0	1	0
D_{11}	0	1	0	1	0	0	0	1	0	0	0	1
D_{12}	1	0	0	1	0	0	0	1	0	0	0	1
D_{13}	0	0	1	1	0	0	0	1	0	0	0	1

4. Developing Techniques

Each technique T consists of four strategies. Each strategy determines one aspect of technique behavior: deriving the most supported opinion, it's quality evaluation, rank refinement and voting order. A combination of four strategies, one of each type, gives us a new technique to be selected context-dependently and used for expertise. All of them are described below.

4.1. Strategy for Deriving the Most Supported Expert Opinion

Method of deriving the most supported opinion concerning question, or task Q is the following. Experts give their votes about usage of each components of domain concept, seems to be an answer on question Q. Then we make the SC^{Q} matrix $n \times m$ which defines relationship between the set of knowledge sources S and their opinions about components C_i of answer V on question Q, as follows:

$$\forall S_i \in S,\ \forall D_i = (C_1, C_2, \dots, C_m),\ D(C_1, C_2, \dots, C_m) \,\&\, P(S, Q, D_i) \Rightarrow (SC^{Q}_{i,q} = C_q),\ q = \overline{1,m}.$$

The technique takes into account the rank of each expert which defines the weight of his vote among all other votes. Let r^{v}_{i} will be the rank of i-th expert before v-th voting.

We construct the vector $VOTE^{Q}$ which contains results of the current experts votes concerning question Q derived from the matrix SC^{Q} as follows:

$$VOTE^{Q}_{q} = \varphi^{Q}_{q} - \psi^{Q}_{q}, \forall q \in \overline{1,m}, \quad \text{where} \quad \varphi^{Q}_{q} = \sum^{n}_{\substack{i,\\ \forall i (SC^{Q}_{i,q} = 1)}} r^{v}_{i}, \quad \psi^{Q}_{q} = \sum^{n}_{\substack{i,\\ \forall i (SC^{Q}_{i,q} = 0)}} r^{v}_{i}.$$

Vector $VOTE$ can correspond to illegal domain concept due to inconsistent expert knowledge about domain and technique's knowledge about real expert authority in it. We must use domain-specific algorithm to produce correct most supported opinion. Such an algorithm for domain of Allen' temporal relations between pairs of temporal intervals is as presented below.

Temporal tasks Q are questions about endpoints' relations of two temporal intervals a and b. We denote triads of the vector $VOTE^{a,b}$, corresponding to endpoints' relations as $VOTE^{a,b}_{q1}, VOTE^{a,b}_{q2}, VOTE^{a,b}_{q3}$. Then we derive $MSUP^{a,b}$ as the vector which contains the most supported opinion on task Q. Every triad gives one unity to MSUP on the same place with maximum of $VOTE^{a,b}$ in this triad as follows:

$$max_t (VOTE^{a,b}_{q1,q2,q3}) \Rightarrow MSUP^{a,b}_{q}, t \in \overline{1,4}, \forall q \in \overline{1,m}.$$

If there are more than one maximal vote in a triad of endpoints' relations $VOTE^{a,b}$ then:

(a) if no one of them correspond to the relation of equivalence between temporal points then there is a conflict between two opinions and we set $MSUP^{a,b}$ as follows:

$$(VOTE^{a,b}_{q1} = VOTE^{a,b}_{q2}) \,\&\, (VOTE^{a,b}_{q1} > VOTE^{a,b}_{q3}) \Rightarrow MSUP*^{a,b}_{q_{1,2}}$$

(b) if one of them corresponds to the relation of equivalence between temporal points then set $MSUP^{a,b}$ as follows:

$$(VOTE_{q1}^{a,b} = VOTE_{q3}^{a,b}) OR (VOTE_{q2}^{a,b} = VOTE_{q3}^{a,b}) \Rightarrow MSUP_{q3}^{a,b} .$$

Other domains will require another specific algorithms to correct incorrect opinions.

Domain-independent number of conflicts $con^v{}_i$ between opinion of i-th expert and the most supported opinion is calculated through all set SC during the v-th voting:

$$con_i^v = \sum_k^m (SC_{i,k}^Q \neq MSUP_k^Q), \ \forall Q_i, \ i = \overline{1,q}, \ \forall i \in \overline{1,n}, \ \forall v \in \overline{1,\infty}$$

This number is used to refine the rank of each expert after certain vote. This takes into account how close are the opinion of this expert and the most supported opinion.

4.2. Strategy for Quality Evaluation of the Most Supported Opinion

We introduce parameter Quality to evaluate adequacy of the most supported opinion to real situation. The voting-type technique supposes that the quality of resulting opinion is better when the number of votes that are equal to the most supported opinion is large. We make the most supported opinion *quality evaluation* QE by the following way:

$$QE = \frac{Votes\ accepted\ as\ most\ supported\ opinion}{All\ votes} \quad QE_v^Q = \frac{\sum_k^m abs(VOTE_k^Q)}{m \cdot \sum_i^n r_i^v} .$$

4.3. Rank Refinement Strategies

Mechanism of expert ranking is used to improve results of voting type processing of the multiple expert’s knowledge. The main formula used to refine rank of each expert is as follows:

$$r_i^{v+1} = r_i^v + \Delta r_i^v ,$$

where the value of Δr_i^v (punishment or prize value) is equal to

$$\Delta r_i^v = \delta_i^v \cdot \sigma_i^v \cdot \frac{\mu^v - con_i^v}{con}$$

which is composed of:

$$\sigma_i^v = \frac{v}{v + n - 1}; \ con = \frac{2}{3} \cdot m; \ \mu^v = \frac{1}{n} \cdot \sum_j^n con_j^v \text{ , and}$$

the value δ_i^v depends on rank refinement strategy selected for an appropriate domain area.

The above formulas are based on the following basic assumptions:

- All experts have the same initial rank equal to $\frac{n}{2}$.
- An expert's rank should always be more than zero and less than number of experts.
- After each vote the rank of each expert should be recalculated.
- An expert improves his rank after some vote if his opinion has less conflicts with the most supported one, than the average number of conflicts among all the experts. Otherwise he loses some part of his rank. [In the main formula this requirement is reflected in the multiplier $\frac{\mu^v - con_i^v}{con}$, where *con* (maximum possible conflicts between opinions) is used to normalize the result.]
- Expert's rank should not be changed after some vote if expert does not participate it or his opinion has as many conflicts with the most supported one as the average number of conflicts among all the experts. [In the main formula this means the case: $\Delta r_i^v = \delta_i \cdot \sigma_i^v \cdot \frac{\mu^v - \mu^v}{con} = 0$.]
- The value of expert responsibility (punishment or prize value) grows from one vote to another. [It means that expert cannot loose or improve his rank essentially during the first vote. However his responsibility will grow accordingly to the multiplier σ_i further.]

In this chapter we consider two of possible strategies of making punishment/prize politics which defines the value of δ_i^v in different ways.

4.3.1. Strategy «Equal Requirements to Leaders and Outsiders»

The main formula used to define the value of δ_i^v is the following:

$$\delta_i^v = \frac{2 \cdot r_i^v \cdot (n - r_i^v)}{n}.$$

This formula provides the following requirements to the rank refinement strategy:

- The value of punishment (prize) for presence (absence) of each conflict should be maximal for expert with the rank equal to $\frac{n}{2}$ (*n* - number of experts). [It is easy to see that value $r_i^v = \frac{n}{2}$ provides maximum of δ_i^v: $\delta_i^v = \frac{2 \cdot \frac{n}{2} \cdot (n - \frac{n}{2})}{n} = \frac{n}{2}$. Within this case, if the number of experts is equal to 2 (minimal number of experts for technique presented), then $\delta_i^v = 1$.]
- The value of punishment (or prize) for presence (or absence) of each conflict should be aspire to zero for expert whose rank is close to zero or to *n*. [If $r_i^v = 0$, then $\delta_i = 0$. If $r_i^v = n$, then $\delta_i = 0$ also.]

This strategy is very demanding to the mistakes of experts. If an expert made only few mistakes in the very beginning and appeared in the group of rank "outsiders" then it is quite difficult or even impossible for him to restore his rank. On the other hand, if an expert was exact enough in the very beginning and appeared in the group of rank "leaders" then it is also quite difficult to fail. Also, accordingly to the strategy, an expert with the smallest possible rank has an equal responsibility for possible mistake as an expert with highest possible rank has. This gives no chance to an outsider to catch up a leader. It is reasonable to use such a strategy in applications where, we try to use opinions of many experts giving them equal starting point in the very beginning. After some votes we select group of rank leaders and use only their opinions in further knowledge acquisition.

4.3.2. Strategy «Leaders Meet Greater Requirements Than Outsiders»

In some cases it is reasonable to have another strategy for prize/punishment politics in rank refinement. The following strategy uses different formula to recalculate ranks in the case of prize and in the case of punishment. The main formula used to define the value of δ_i^v is as follows:

$$\delta_i^v = \begin{cases} \dfrac{(r_i^v - n)^2}{2 \cdot n}, & if\,(\mu^v - con_i^v) \geq 0; \\ \dfrac{(r_i^v)^2}{2 \cdot n}, & if\,(\mu^v - con_i^v) < 0. \end{cases}$$

This formula provides the following requirements to the rank refinement strategy:

- The value of prize should be maximal for expert which rank is close to zero. [It is easy to see that value $r_i^v = 0$ provides maximum of δ_i^v: $\delta_i^v = \dfrac{(r_i^v - n)^2}{2 \cdot n} = \dfrac{(0 - n)^2}{2 \cdot n} = \dfrac{n}{2}$ inside the interval $[0, n]$. Within this case, if number of experts is equal to 2 (minimal number of experts for presented technique), then $\delta_i^v = 1$.]
- The value of punishment should be maximal for expert which rank is close to *n*. [It is easy to see that value $r_i^v = n$ provides maximum of the $\delta_i^v = \dfrac{(r_i^v)^2}{2 \cdot n} = \dfrac{n^2}{2 \cdot n} = \dfrac{n}{2}$ inside the interval $[0, n]$. Within this case, if the number of experts is equal to 2 (minimal number of experts for presented technique), then $\delta_i^v = 1$.]
- The value of prize should be aspire to zero for expert which rank is close to *n*. [One can see that if $r_i^v = n$, then $\delta_i = 0$ in the prize formula.]
- The value of punishment should be aspire to zero for expert whose rank is close to zero. [If $r_i^v = 0$, then $\delta_i = 0$ in the punishment formula.]

This strategy is much more flexible to the mistakes of experts than previous one. If an expert made few mistakes in the very beginning and appeared in the group of rank outsiders then he will not be as much responsible for every new mistake as a leader. In the same time an outsider has still enough possibility to improve his rank in the case of producing exact opinions in future. On the other hand, if an expert was exact enough in the very beginning and appeared in the group of rank "leaders" then he became highly responsible for any mistake in the future. It is reasonable to use such a strategy in applications where we always want to use opinions of all experts and we want them to be motivated to learn. Also we want the best experts, who possibly have more monetary support for their expertise, to be highly responsible for possible mistakes.

4.4. Experts Voting Strategies

Three voting strategies were developed to manage expertise order. Each of them has it's own applications and can be selected context-dependently.

4.4.1. Real-time Voting Strategy

Real-time voting strategy uses the natural way of questions (tasks) and forces experts to start with the problem Q_1, then continue with Q_2, up to Q_q. Technique produces corresponding common opinions to fill V_1 solution at first, then V_2, up to V_q.

Each problem Q_i forces technique to derive the most supported opinion, it's quality evaluation and to recalculate expert ranks. That is, q problems require q rank recalculations.

4.4.2. Real-time Strategy With Test Questions

This strategy introduces test questions formally. We assume, that some questions are test ones, with already known answer. Such questions help to evaluate expert authority directly. Experts solve them as usual problems, but answers are already known. When question Q_k is a test one, then the reply set V contains an element V_k with already assigned most supported opinion ξ:

$$V_k = \xi,\ \xi \in D,\ V_k \in V\ .$$

This strategy works in the same way, as real-time strategy, but rank refinement uses predefined most supported opinion and ignores experts real common opinion to recalculate expert ranks.

4.4.3. Batch Strategy

Batch strategy lets experts to vote the same questions few times to repeat their correct or wrong answers. This helps to make more flexible rank evaluation.

We define the sets Q^B and V^B as follows:

$$Q^B = \underbrace{Q \cdot Q \cdot \ldots \cdot Q}_{k\ times},\ V^B = \underbrace{V \cdot V \cdot \ldots \cdot V}_{k\ times},$$

where operation « · » denotes concatenation of two ordered sets.

We use the sets Q^B and V^B instead of sets Q and V in real-time voting process. Technique produces k series of q common opinions. We take last most supported as a final common expert opinion: last q elements of the set V^B will form the resulting set V as follows:

$$V_i = V^B_{q\times(k-1)+i},\ i = \overline{1,q}\ .$$

That is, expert repeats the same answers on the same questions k times. Expert change his rank during such iterative discussion, according to relationship between his opinion and most supported one. This strategy require $k*q$ rank recalculations. The resulting common opinion V is a result of more flexible rank refinement, than in real-time technique.

Test relations can be introduced into batch technique in the same way, as in real-time technique. Experts will pass each test k times, which increases test effect k times, too.

4.4.4. Parallel Strategy

Parallel strategy is very similar to the batch. It differs only in rank recalculation. Experts pass every iteration of q questions without rank changing. Their individual prizes and punishments Δr are summarized, and their mean is final prize (or punishment) Δr, as follows:

$$\Delta r_i = \frac{1}{q}\sum_{j=1}^{q} \Delta r_{ij}, \; i = \overline{1,n} \,.$$

Parallel voting strategy demands rank recalculations after each iteration. That is, k iterations of q questions each, require k rank recalculations.

This strategy can have another interpretation. We assume, that the whole set Q is one whole task, voted by it's parts. Expert has to vote about all subtasks Q_i from Q, to express his opinion on task Q. We must recalculate his rank on the basis of all q individual opinions, and q most supported opinions, made with constant rank. We cannot change his rank on the basis of one subtask Q_i.

Parameter *Quality* can be calculated as a mean of q qualities, obtained while voting series, too. Parallel voting is necessary for compound problems, when we cannot make any judgements about experts on the basis of any part of it.

5. Experimental Investigation

Let us consider two examples of two rank refinement strategies, where experts vote on three tasks from Allen temporal domain.

Each expert expressed three opinions in three tasks (q=3), as shown in Table 3.

Table 3. Expert opinions

Expert	1st vote	2nd vote	3rd vote
S_1	T_1 during T_2	T_3 after T_4	T_5 includes T_6
S_2	T_1 overlaps T_2	T_3 meets T_4	T_5 finished by T_6
S_3	T_1 starts T_2.	T_3 overlapped by T_4	T_5 after T_6
S_4	T_1 finished by T_2	T_3 before T_4	T_5 overlapped by T_6

5.1. Strategy of Equal Demands to Leaders and Outsiders

We proceed expert's opinions with batch voting strategy and rank refinement strategy of equal demands, with seven iterations.

The most supported opinion on each vote depends on expert's ranks on this vote. Experts ranks changing is presented in Fig. 3, corresponding MSUP dynamics is listed in Table 4.

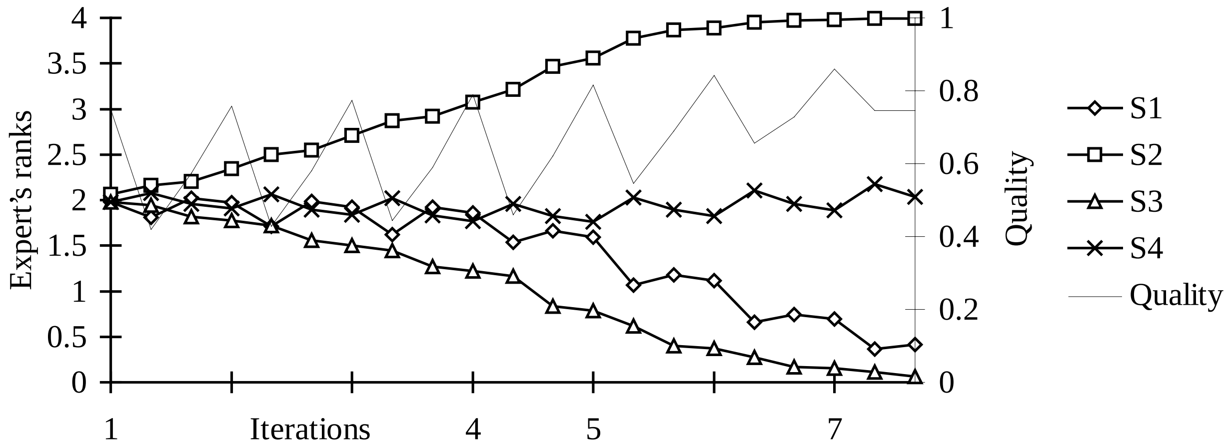


Fig. 3. Rank dynamics under the strategy of equal demands

Experts have close ranks after first iteration. Leaders and outsiders become evident later. MSUP on iterations 1-3 is «Task 1: (7) Overlaps; Task 2: (7) Overlaps; Task 3: (6) Includes». It is a common opinion of four experts with near ranks. Experts S1 and S4 seems to be good specialists, and they pass voting iterations without significant rank changes. S2 rises his rank on these iterations, and S3 looses his rank.

Iterations 4-5 are critical. High rank of S2 makes him dominant and he begins to form the most supported with his individual opinion, as shown in Table 4. New most supported conflicts with opinion of S1, so S1 begins to loose his rank quickly, as well as S3. MSUP changing does not influence on rank dynamics of expert S4.

Parameter Quality is also shown in Fig. 4. It varies greatly from vote to vote on these iterations.

Iterations 6-7 fix the situation, when S2 has the highest rank (equal to 3,9947) and forms the most supported opinion. Rank of S3 is very small, equal to 0,0642. Variance of quality parameter decreases on these iterations.

Experts opinions and vote results are presented graphically in Fig. 4. In Fig. 4 all temporal relations, supported by experts are presented in rows with Allen's temporal intervals. Experts

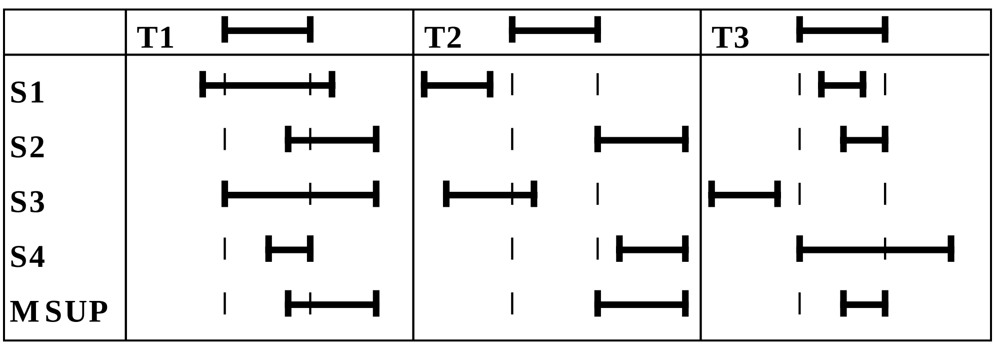


Fig. 4. Illustration for the strategy of equal demands

opinions on the first task are listed in column T1, on the second task — in column T2 and on the third task — in column T3. Each interval has it's pair, presented in the top row. That is, every opinion is a relation between interval in a corresponding cell and interval in the top of

Table 4. MSUP dynamics

Itera-tion	The Most supported opinions		
	T1	T2	T3
1	(7) Overlaps	(7) Overlaps	(6) Includes
2	(7) Overlaps	(7) Overlaps	(6) Includes
3	(7) Overlaps	(7) Overlaps	(6) Includes
4	(7) Overlaps	(7) Overlaps	(12) Finished by
5	(7) Overlaps	(3) Meets	(12) Finished by
6	(7) Overlaps	(3) Meets	(12) Finished by
7	(7) Overlaps	(3) Meets	(12) Finished by

corresponding column. The most supported opinions on every task after 7 iterations are presented in the lowest row.

5.2. Strategy of Different Demands to Leaders and Outsiders

Second rank refinement strategy is the strategy of different demands. Rank dynamics, while proceeding expert opinions with this strategy, is shown in Fig. 5. Corresponding MSUP dynamics is listed in Table 5. We see that experts change their ranks slower, than under the strategy of equal demands. Really, expert has to be more precise to increase his rank significant. From the other side, he must produce extremely wrong opinions to loose his rank greatly.

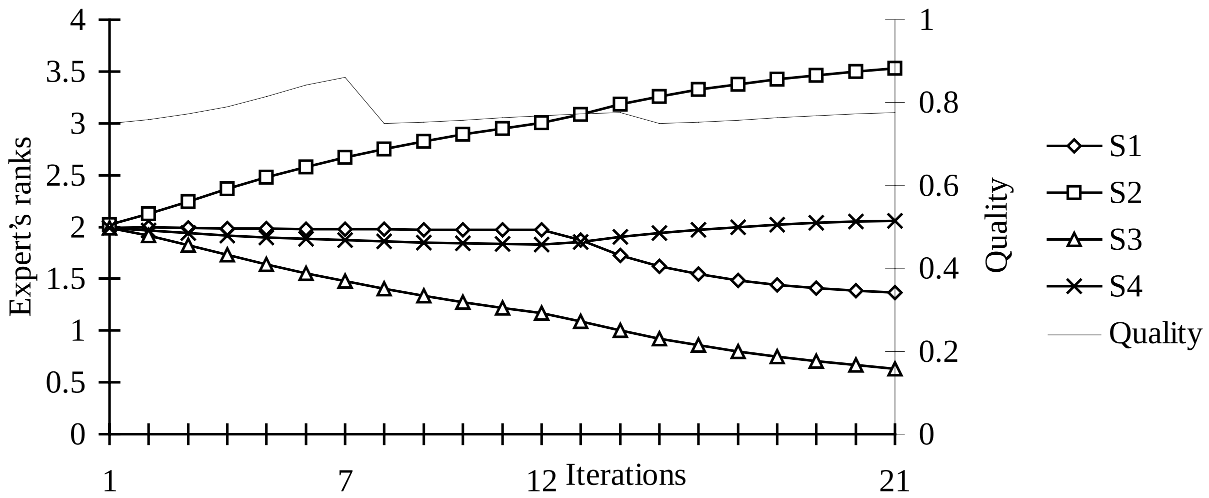


Fig. 5. Rank dynamics under the strategy of different demands

First 7 iterations give no changes in the most supported opinion because experts still have close ranks. Quality parameter rises slowly, without fluctuations. Graphical illustration for 7 iterations is presented in Fig. 6. Information in this figure is presented in the same way, as in Fig. 4.

We proceed expert opinions 3 times more and do 21 iterations to see rank changing. Iterations 1-11 give the same most supported. Expert S2 rises his rank slowly, as well as S3 falls. Experts S1 and S4 keep approximately same ranks.

Iteration 12 is a critical one. S2 begins to form most supported due to his high ranks, as it was under previous strategy. Then S1 and S3 begin to fall in their rank. S2 is still rising his rank.

Note, that S2 has rank, equal to 3,5520 versus 3.9947 under previous strategy. S3 has the lowest rank, equal to 0,6023 versus 0.0642. Expert S2 is unable to take extremely high rank, and S3 is unable to fall. Quality parameter changes with less amplitude, than under previous strategy.

Expert opinions and voting results after 21 iteration coincides with results, obtained with

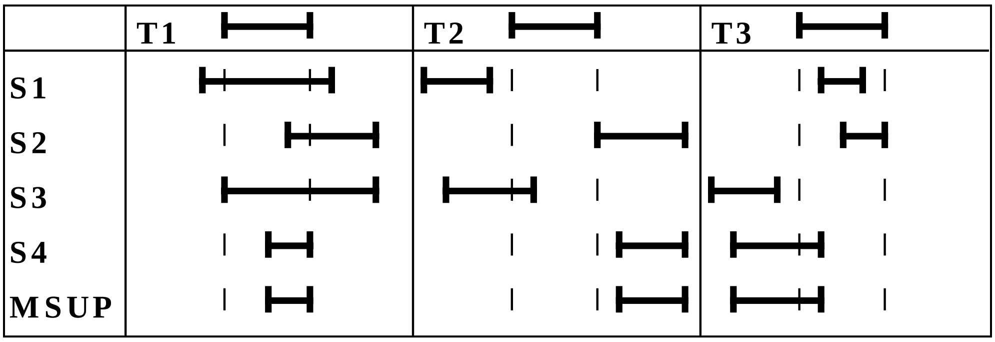


Fig. 6. Illustration for the strategy of different demands

Table 5. MSUP dynamics

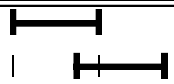
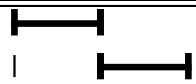
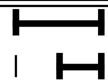
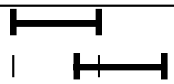
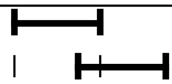
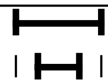
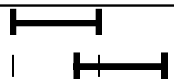
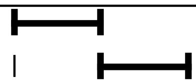
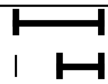
Itera-tion	Most supported opinions		
	T1	T2	T3
1	(7) Overlaps	(7) Overlaps	(6) Includes
2	(7) Overlaps	(7) Overlaps	(6) Includes
3	(7) Overlaps	(7) Overlaps	(6) Includes
4	(7) Overlaps	(7) Overlaps	(6) Includes
5	(7) Overlaps	(7) Overlaps	(6) Includes
6	(7) Overlaps	(7) Overlaps	(6) Includes
7	(7) Overlaps	(7) Overlaps	(6) Includes
8	(7) Overlaps	(7) Overlaps	(6) Includes
9	(7) Overlaps	(7) Overlaps	(6) Includes
10	(7) Overlaps	(7) Overlaps	(6) Includes
11	(7) Overlaps	(7) Overlaps	(6) Includes
12	(7) Overlaps	(7) Overlaps	(12) Finished by
13	(7) Overlaps	(3) Meets	(12) Finished by
14	(7) Overlaps	(3) Meets	(12) Finished by
15	(7) Overlaps	(3) Meets	(12) Finished by
16	(7) Overlaps	(3) Meets	(12) Finished by
17	(7) Overlaps	(3) Meets	(12) Finished by
18	(7) Overlaps	(3) Meets	(12) Finished by
19	(7) Overlaps	(3) Meets	(12) Finished by
20	(7) Overlaps	(3) Meets	(12) Finished by
21	(7) Overlaps	(3) Meets	(12) Finished by

previous strategy. They are presented in Fig. 4.

The most supported opinions, obtained after proceeding expert opinions with both strategies are presented in Table 6.

Table 6 shows, that both strategies form similar consensus after 7 iterations. Both of them give the same most supported opinions on the 1st task («Overlaps»). Most supported opinions on the

Table 6. Most supported opinions

Most supported opinions		
Strategy of equal demands, 7 iterations		
T1  (7) Overlaps	T2  (3) Meets	T3  (12) Finished by
Strategy of different demands, 7 iterations		
T1  (7) Overlaps	T2  (7) Overlaps	T3  (6) Includes
Strategy of different demands, 21 iteration		
T1  (7) Overlaps	T2  (3) Meets	T3  (12) Finished by

2nd task differ only in one endpoint relation, as well as the 3rd most supported.

Table 6 also shows that the most supported under the first strategy after 7 iterations is equal to MSUP under the second strategy after 21 iterations.

Conclusion

We developed a flexible voting-type technique for knowledge acquisition from multiple experts. It's flexibility is based on it's rank refinement strategies. Two possible strategies for rank refinement are developed and experimentally investigated.

We found, that the strategy of equal demands to leaders and outsiders produces rank leaders and rank outsiders very fast (after 4 iterations). But quality evaluation of voting process has great deviation during voting process. Strategy demands less computer resources to run, but gives rough and varying results. We must use this strategy in applications, critical to time for proceeding expert opinions. Strategy allows to agree to loose some quality of expertise results for speed.

Strategy of different demands for leaders and outsiders produces leaders and outsiders slowly, than previous strategy (after 13 iterations). But it gives us more soft quality and rank changing. It needs more computer resources to run, but produce more precise results. This strategy can be applied in areas, where high quality of expertise results are required. This is possible on low number of experts or tasks, when we can perform sufficient calculations.

Analysis showed that both strategies give the same results on our test example. But time to achieve these results differs greatly. Such a difference increases with rising the number of experts. So, we have to develop a context-dependent method to select strategy, appropriate to domain, expert stuff and other conditions.

Another way for future research is to develop a multilevel model to make multi-level and distributed expertise. Present model will form one level of it's structure. Results of present research will help to evaluate resource requirements of different levels of this future model. This problem is extremely important for distributed expertise with huge number of experts, limited resources and high demands for reply time.

References:

1. **Allen, J.F.**, Maintaining Knowledge About Temporal Intervals, *Communications of the ACM*, V.26, No.11, 1983, pp. 832-843.

2. **Allen, J.F.,** Temporal Reasoning and Planning, In: Allen J.F. (ed.), *Reasoning About Plans*, Morgan Kaufmann, San Mateo, 1991, pp. 8-33.

3. **Barrett, A.R. and Edwards, J.S.,** Knowledge Elicitation and Knowledge Representation in a Large Domain with Multiple Experts, *Expert Systems with Applications*, Pergamon, V.8, No.1, 1995, pp. 169-176.

4. **Kaikova, H. and Terziyan, V.,** Temporal Knowledge Acquisition From Multiple Experts, In: Shoval P. & Silberschatz A. (Eds.), *Proceedings of NGITS'97*, Neve Ilan, Israel, June 30 - July 3, 1997, pp. 44 - 55.

5. **Mak, B., Bui, T. and Blanning, R.,** Aggregating and Updating Experts' Knowledge: An Experimental Evaluation of Five Classification Techniques, *Expert Systems with Applications*, Pergamon, V.10, No.2, 1996, pp. 233-241.

6. **Medsker, L., Tan, M. and Turban, E.,** Knowledge Acquisition from Multiple Experts: Problems and Issues, *Expert Systems with Applications*, Pergamon, V.9, No.1, 1995, pp. 35-40.

7. **Taylor, W.A., Weimann, D.H. and Martin, P.J.,** Knowledge Acquisition and Synthesis in a Multiple Source Multiple Domain Process Context, *Expert Systems with Applications*, Pergamon, V.8, No.2, 1995, pp. 295-302.

8. **Taylor, W.A., Weimann, D.H. and Martin, P.J.,** Knowledge Acquisition and Synthesis in a Multiple Source Multiple Domain Process Context, *Expert Systems with Applications*, Pergamon, V.8, No.2, 1995, pp. 295-302.