Supporting Intelligence Analysts with a Trust-based Question Answering System

Maaike de Boer (3370852)

Master Thesis
30 ECTS
Cognitive Artificial Intelligence
University of Utrecht
Faculty of Humanities

First Supervisor & Reviewer:
dr. G.A.W. Vreeswijk
Second Supervisor & Reviewer:
dr. P.-P. van Maanen
Third Reviewer:
dr. Stefan van der Stigchel

Abstract

Intelligence analysts have to deal with massive amounts of data, severe time constraints and highly dynamic environments. This results in high workload and causes mistakes with severe consequences, which is the reason that support systems for intelligence analysts have been developed. The support system proposed in this thesis is based on a trust-based question answering system, named T-QAS. An important part of T-QAS are trust models which keep track of trust in each of the agents gathering information. Using these trust models, T-QAS decides which agents receive a question and the order of answers.

T-QAS is evaluated in an experiment in which participants perform a decision making task with information from unreliable sources and time constraints. Results indicate that system support helps participants to execute their task.

Acknowledgements

I would like to thank the following people for their contribution to my thesis:

- Peter-Paul van Maanen for supervising and reviewing my thesis from TNO
- Gerard Vreeswijk for supervising and reviewing my thesis from the University of Utrecht
- Stefan van der Stigchel for being my third reviewer
- Jan-Willem Streefkerk for reviewing my thesis
- Myrthe Tielman, Jacco Spek and Tinka Giele for participating in the pilot and reviewing parts of my thesis
- Suzanne van Trijp, Piet Bijl and Wim-Pieter Huijsman from TNO for their information and collaboration
- Iris van Dam, Mirjam de Haas, Martijn van den Heuvel, Pauline Hovers, Michael de Jong, Oscar van Loosbroek, Marcel Nihot, Christian van Rooij, Gert van der Schors, Wessel van Staal and Hanna Zoon for participating in the pilot
- DIVI, especially lieutenant-colonel M. Hagoort and major M. Hagoort, for providing the LOWLAND scenario
- Judith Ritmeijer - Dijkstra for gathering participants
- TNO for providing the opportunity to do my graduation project

Contents

Abstract ii

Acknowledgements iv

1 Introduction 4
- 1.1 Research Question . . . 5
- 1.2 Relevance to CAI . . . 6
- 1.3 Structure . . . 6

2 Theoretical Background 7
- 2.1 Support Systems for Intelligence Analysts . . . 7
  - 2.1.1 Support for Planning & Direction . . . 9
  - 2.1.2 Support for Collection . . . 10
  - 2.1.3 Support for Processing . . . 11
  - 2.1.4 Support for Analysis & Production . . . 11
  - 2.1.5 Support for Dissemination . . . 13
  - 2.1.6 Summary . . . 13
- 2.2 Question Answering Systems . . . 14
  - 2.2.1 Generic Architecture of a QA System . . . 14
  - 2.2.2 Types of QA Systems . . . 18
  - 2.2.3 Obstacles in QA . . . 18
  - 2.2.4 Standards in QA . . . 19

3 Support Model 20
- 3.1 Question Delivery . . . 22
  - 3.1.1 Question Analysis . . . 23
  - 3.1.2 Agent Selection . . . 23
  - 3.1.3 Modality Selection . . . 23
  - 3.1.4 Time Selection . . . 23
- 3.2 Trust Management . . . 24
  - 3.2.1 Agent Information . . . 25
  - 3.2.2 QA Information . . . 25
  - 3.2.3 Trust Maintenance . . . 25
- 3.3 Selection of Answer(s) . . . 26
  - 3.3.1 Answer Tiling . . . 26
  - 3.3.2 Answer Ranking . . . 27
  - 3.3.3 Response Formulation . . . 27
- 3.4 Example . . . 28

4 Implementation **32**
4.1 Stage One . . . 32
4.2 Stage Two . . . 33

5 Hypotheses **34**
5.1 Task Performance . . . 34
5.2 Trust Estimation Performance . . . 35
5.3 Cognitive Workload . . . 36

6 Method **37**
6.1 Participants . . . 37
6.2 Task . . . 37
6.2.1 Task Description . . . 38
6.2.2 User Interface . . . 39
6.3 Design . . . 40
6.4 Independent Variables . . . 41
6.5 Dependent Variables . . . 42
6.5.1 Task Performance . . . 42
6.5.2 Trust Estimation Performance . . . 43
6.5.3 Trust Estimation Compliance . . . 43
6.5.4 Cognitive Workload . . . 43
6.5.5 Personality and Preferences . . . 44
6.6 Apparatus . . . 44
6.7 Procedure and Instructions . . . 44
6.8 Analyses . . . 45

7 Results **46**
7.1 Task Performance . . . 46
7.1.1 Good vs. Poor Performers . . . 47
7.2 Trust Estimation Performance . . . 48
7.2.1 Trust Estimation Compliance . . . 49
7.3 Cognitive Workload . . . 51
7.4 Personality and Preferences . . . 51

8 Discussion **52**
8.1 Task Performance . . . 52
8.2 Trust Estimation Performance . . . 52
8.3 Cognitive Workload . . . 53
8.4 Further Research . . . 54

9 Conclusions **55**

References **56**

Appendices **61**

A Questionnaires (Dutch) **61**

B Instructions (Dutch) **67**

1 Introduction

Intelligence analysts supervise, coordinate and participate in the analysis, processing and distribution of intelligence in defense organizations (Powers, n.d.) (M. Powell, 2012). Intelligence analysts have to gather and provide timely and relevant information to decision makers. If crucial information is missed or not provided timely, friendly fires or wrong decisions can have severe consequences. An example is an attack by American helicopters which killed twenty-three civilians (Shanker & Richtel, 2011). In this example a team of operators of drones failed to pass crucial information about the makeup of a crowd. Reports that indicated that the group included children were missed due to a massive amount of data and time constraints.

Not only a massive amount of data and time constraints are a problem, but also ambiguous questions, the cumbersome data gathering process, a dynamic environment and uncertainty, ambiguity and complexity of information are problems. Many solutions have been proposed to deal with these problems. Both JIGSAW (Cook & Smallman, 2008) and AICMS (Gonsalves & Cunningham, 2001) are tools that help analysts decompose information needs into questions to solve the problem with ambiguous questions. McGrath, Chacón, and Whitebread (2000) and Van Diggelen, Grootjen, Ubink, van Zomeren, and Smets (2012) propose to change the data gathering process. If data is gathered from soldiers, soldiers have to remember the questions to know which answers to search for. At the end of a patrol, soldiers write a patrol report in which all found answers are reported. In the proposed solutions, the questions and available information can be pushed to soldiers. Soldiers can place an answer with the question. In this way answers to the questions can be placed with the question directly and are provided timely to the intelligence analyst. The problem with a dynamic environment is that situations unfold and evolve over time. In order to picture these situations, timeline analysis, the recognition of updates which overturn previous information and a near-time updating knowledge database are proposed (Patterson et al., 2001). Dealing with high uncertainty, ambiguity and complexity of information will likely remain beyond the capabilities of software tools for some time (Badalamente & Greitzer, 2005). Several solutions to support intelligence analysts in dealing with uncertain, ambiguous and complex information are proposed by G. M. Powell and Broome (2002), Jenkins and Bisantz (2011), Patterson et al. (2001) and Newcomb and Hammell II (2012). It remains unclear what the added value is of these proposed systems. Many solutions have been proposed to deal with the massive amount of data. First, the processing of data can be automated with video tagging systems and systems such as an automated system from the Navy Research Lab (*NRL Flight-Tests Autonomous Multi-Target, Multi-User Tracking Capability*, 2012) which are able to spot, track and identify objects in real-time. The system of the Navy Research Lab can deal with vehicle-size and human-size targets on the ground. The processed data, which is now information, can be analyzed with cloud computing techniques (Chen & Deng, 2009), filtering techniques (Lusher & Stone III, 1997) and context aware information retrieval (Bahrami, Yuan, Smart, & Shadbolt, 2007).

The overflow of information remains a challenge for intelligence analysts (Matthews, 2012). According to G. M. Powell and Broome (2002) approximately 10,000 messages per hour are received and only 15,000 messages can be scanned a day (Jenkins & Bisantz, 2011). Eppler and Mengis (2003) states that heavy information load affects performance, measured as accuracy or speed, negatively. With too much information people may have difficulties with identifying relevant information or relationships between details and the overall perspective.

Not only intelligence analysts deal with too much information, other levels of the military (McGlaun, 2009) (Hsu, 2011), other organizations and many people have to deal with too much information, for example by searching information on the World Wide Web, attending social media websites and getting a lot of e-mails. Search engines such as Google, Yahoo!, Bing, Ask and Blekko use word matching and ranking techniques to select relevant information. Question answering websites such as 3form, Answerbag, Answers, Askeet, Askalo, Askville, ChaCha, Eduflix and Quora use reputation and votes to display the most probable answers to a question.

The biggest problems are thus the uncertain, ambiguous and complex information and the amount of information. In order to solve the problem with uncertain information trust models could be used. Trust models can indicate the reliability of the information from a source. Low trust in certain sources indicates less reliable information. With this reliability most probable answers to a question can be determined. Trust models are also used to determine which sources receive a question. If not all sources receive a question, the amount of incoming information is reduced.

1.1 Research Question

In this thesis a support model for intelligence analysts is proposed to deal with uncertain information (due to possibly unreliable sources) and too much information. It is, therefore, interesting to see if this support model helps intelligence analysts. The support model can deal with all kinds of sources, but in this thesis the focus will lie on human sources. The research question is:

Does trust-based decision support help intelligence analysts in the task of decision making with information from possibly unreliable sources?

1.2 Relevance to CAI

The program of Cognitive Artificial Intelligence has two tracks: Cognitive Modelling and Logic and Intelligent Systems. Aspects of both tracks can be found in this thesis. Cognitive Modelling is focused on understanding human intelligence and experimental research. In this thesis a model for gathering intelligence from humans is proposed. An experiment is set up in order to test this model. The thesis is, therefore, relevant to the track Cognitive Modelling. The other track, Logic and Intelligent Systems, is focused on autonomous intelligent computer systems and computational linguistics. An autonomous intelligent computer system is developed with the use of techniques acquired from the field of computational linguistics. This makes this thesis also relevant to the track Logic and Intelligent Systems.

Aside from the different tracks both the Cognitive and the Artificial Intelligence part from CAI are covered. Cognition is about information processing, memory, producing and understanding language, learning, problem solving and decision making. Most, if not all, aspects of cognition can be found in the model. Several techniques used in the field of Artificial Intelligence are used in order to improve the system. Furthermore the interdisciplinarity of Artificial Intelligence is also present in this thesis. Methods from psychology are used in order to set-up an experiment, programming skills from computer science are used for the implementation from the model into a system and approaches from (computational) linguistics are used for the model.

To summarize, this thesis is very relevant to (Cognitive) Artificial Intelligence.

1.3 Structure

Chapter 2 will discuss the theoretical background of this thesis. The theoretical background consists of information about existing and proposed support systems for intelligence analysts and question answering systems. In chapter 3 the proposed support model for intelligence analysts is introduced. Chapter 4 contains the hypotheses. Two stages of the model are implemented and information about the implementation can be found in chapter 5. Both implementations are evaluated in an experiment of which the task, design, independent and dependent variables, apparatus, procedure and instructions are explained in chapter 6. Chapter 7 gives the results of the experiment. Chapter 8 contains the discussion and chapter 9 consists of the conclusions.

2 Theoretical Background

The theoretical background consists of two parts. The first part contains information about proposed and existing support systems for intelligence analysts. The second part focuses on question answering systems.

2.1 Support Systems for Intelligence Analysts

Before proposed and existing support systems for intelligence analysts can be explained, the different work processes and problems of intelligence analysts have to be clarified. First we can divide intelligence into strategic and operational intelligence (Staff, 1996). Strategic intelligence integrates information about politics, military affairs, economics, societal interactions, and technological developments. This information is used to make national policy or decisions of long-lasting importance. Operational intelligence is used to determine current and near-term events instead of long-term projections. Besides different kinds of intelligence, multiple intelligence sources or collection disciplines exist (Boury-Brisset, Frini, & Lebrun, 2011): Human Intelligence (HUMINT), Imagery Intelligence (IMINT), Geospatial Intelligence (GEOINT), Measurement and Signature Intelligence (MASINT), Signal Intelligence (SIGINT) with subdisciplines Communications Intelligence (COMINT) and Electronic Intelligence (ELINT), Open Source Intelligence (OSINT) and Technical Intelligence (TECHINT). All-source or multi-INT deals with intelligence from all above mentioned collection disciplines. Intelligence analysts can also work on different levels in the organization, for example in the Collection Coordination and Information Requirement Management (CCIRM), the G2, S2 or Team Intell Cell (TIC).

Although the different intelligence types, levels and disciplines require a slightly different work process, general intelligence cycles are proposed. Four, five, six and seven processes or steps are proposed. Biermann (2006) uses a four-stepped intelligence cycle: direction, collection, processing and dissemination. Staff (1996) and the CIA (*The Intelligence Cycle*, 2007) add a step and get the following five steps: planning & direction, collection, (analysis &) processing, (analysis &) production and dissemination (see Figure 1). In the first step a commander formulates questions to which he will require answers in order to successfully perform the operations (Biermann, 2006). These questions are collected and prioritized by the Collection Coordination and Information Requirement Management (CCIRM). The questions are now called Priority Intelligence Requirements (PIR) and further broken into individual Intelligence Requirements (IR) and Essential Elements of Information (EEI). With EEIs CCIRM and G3 Plans decide which sources are suitable to provide this information. These sources can be controlled, uncontrolled or casual (Biermann, 2006). Controlled sources are under the control of the intelligence staff and can be tasked to answer questions. The uncontrolled sources are not under the control of the intelligence staff and are for example media. The casual sources produce information from an unexpected quarter. An example is a refugee. In the second step raw data is gathered from the sources. These sources can be people, but also listening devices, hidden cameras and satellite photography.

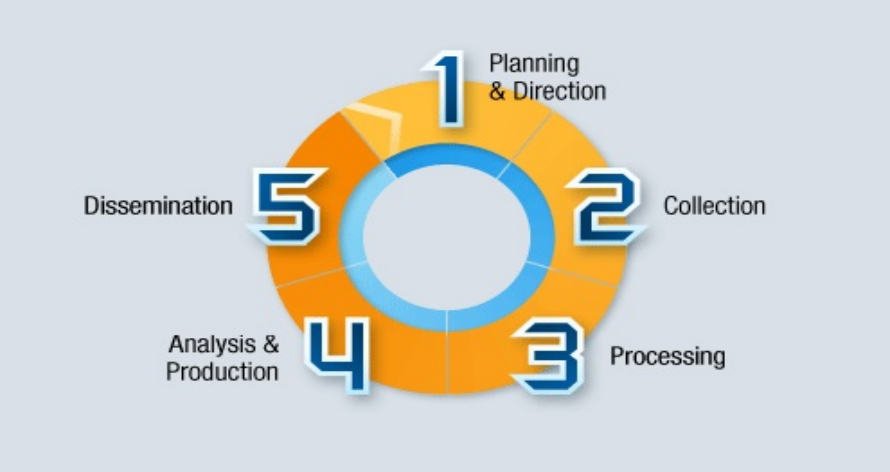


Figure 1: Intelligence Cycle (from *What we do* (n.d.))

In the third step the data is processed into a form that is suitable for analysis and production. The fourth step is production in which the processed data, now called information, is analyzed, evaluated, interpreted and integrated to form an intelligence product. The product may be developed from a single source or the all source department (Staff, 1996). In this step the analyst must weight the reliability, validity and relevance of the information for the information requirement. The information has to be put in context and additional questions are to be formed to fill in the gaps left by the collection or existing intelligence databases. In the last step of the intelligence cycle the final or finished intelligence report is distributed to the consumers, for example the commander who posed the questions. With the new formed questions the intelligence cycle can start again. The FBI (*Intelligence Cycle*, n.d.) proposes a step before these 5 steps: requirements. The DIA (*Vector 21 - A strategic plan for the Defense Intelligence Agency*, n.d.) adds feedback and evaluation within the cycle.
Jenkins and Bisantz (2011) describe seven different stages of which the six stages of the FBI are used and production is split into the moment of hypothesis and additional data collection & hypothesis testing. Each of these processes pose problems for intelligence analysts, so in the next sections the problems and proposed support systems to reduce the problems for each process step are explained.

2.1.1 Support for Planning & Direction

Input for the process *planning* are questions to which a commander will require answers in order to successfully perform the operations. Problems in this process can be that information needs are poorly expressed or communicated and the operational context of the question is unclear, which results in ambiguous questions (Jenkins & Bisantz, 2011) (Roth et al., 2010). These problems become worse if the analysts cannot directly ask for feedback or clarification because of the distributed structure of the intelligence analysis (Jenkins & Bisantz, 2011). This distributed structure also poses a problem with the direction of the information request. Analysts may not have enough knowledge about the available resources and the capabilities of the forces (Roth et al., 2010). The problems can result in defining the information need too narrowly, too broadly or incorrectly focusing on irrelevant details (Roth et al., 2010). A too narrowly defined information need results in missing information, whereas a too broadly defined information need results in a situation in which the request cannot be effectively answered in the time available. A wrong focus of the information need results in gathered information that does not directly support the decision context. This wrong focus can also mean prioritizing the wrong information requirements. Well-trained intelligence analysts may be able to reduce the problem with experience, but in many cases analysts are underskilled (Roth et al., 2010). Training focusing on strategies for interpreting information and strategies for seeking out and working with the request originator are proposed as a solution by Roth et al. (2010). Another solution proposed by Roth et al. (2010) is to create more effective tools to support analysts in interpreting and translating information requests to appropriate collection requirements. This solution is picked up by Cook and Smallman (2008) with a Joint Intelligence Graphical Situation Awareness Web (JIGSAW) and by Gonsalves and Cunningham (2001) with an Automated ISR Collection Management System (AICMS). The JIGSAW tool helps analysts to decompose information requests and defines and explores hypotheses with collaborative visualization tools. This tool is extremely useful if the intelligence cycle is in the second or third round, because the selection bias of hypothesis can be significantly reduced. Analyst tend to mention and share only consistent information, which can lead to under-weighted or misinterpreted evidence. Using a second analyst as support did not reduce the selection bias (Cook & Smallman, 2008). The Automated ISR Collection Management System (AICMS) can decompose higher-level information needs (PIRs or IRs) into lower more specific information requirements (EEIs) using a Bayesian belief network and it can generate a collection plan to satisfy these requirements using fuzzy logic reasoning.

Other proposed solutions are a system that supports multiple searches for situations of interest (Jenkins & Bisantz, 2011) and a facilitation of the communication between the analyst and the commander (Roth et al., 2010).

2.1.2 Support for Collection

In the process in which data is collected several problems occur. The biggest problem remains the distributed structure. It is possible that data from another organization or expert is needed, which can result in language or jargon barriers. Each extra step from the analyst to the data provider increases the likelihood of miscommunication and staleness of the information (Jenkins & Bisantz, 2011). If the common ground is not maintained, relevant information may not be communicated. This may result in adding new information requirements or creating new hypotheses. The problem space is, however, theoretically infinite, so an open world of possible explanations creates a wide range of unpredictable hypotheses. This unbounded range of possibilities results in a high degree of mental workload (Jenkins & Bisantz, 2011).

A solution to the problem of creating new hypotheses is already explained in the previous section. A solution to the problem of gathering data from HUMINT is explained in McGrath et al. (2000) and Van Diggelen et al. (2012). The biggest problem with gathering data from HUMINT is that all EEIs have to be remembered by the soldiers before they are collecting the data. This can result in missing data, because soldiers forgot to search for that data. Instead of having to remember the EEIs, this information can be made digital and be supported by a system. McGrath et al. (2000) explains three capabilities of a Domain Adaptive Information System: information push, information pull and sentinel monitoring. The Domain Adaptive Information System has mobile and static agents that can automatically send (push) information to soldiers that may need it and retrieve (pull) relevant information from soldiers. An multi-agent system is used in which an Information Agent is able to find data, a Monitor Agent monitors data sources for potential conflicts or critical conditions and a Scout Agent is searching in the network for information about a specific hypothesis. In this way the information is found and sent sooner, and placed with the right hypothesis. It is, however, not yet known how reliable and useful the agents are.

A system that focuses on pushing information is proposed by Van Diggelen et al. (2012). The system can deliver the Right Message at the Right Moment in the Right Modality. This is done with smart questions. Smart Questions is a technology developed by TNO (Kerbusch, Paulissen, van Trijp, Gaddur, & van Diggelen, 2012) that provides the opportunity to directly ask a question within an organization. The question can be asked with an electronic form that has tags providing meta-information such as the time after which the question is not interesting anymore, the domain of the question, the security level of the question, a specific rank needed for the answer, the location in which the question can be answered and the time it possibly (or minimally) takes to find the answer to the question. The system works with databases and a multi-agent system. The databases contain information about the content of the question, the users and the user interfaces. The agents can retrieve information, match information to users, match the best user interface to a user and send the information to an agent. A user monitoring agent and user interface monitoring agent keep track of the available users and user interfaces per user. Both systems could resolve the problem of the distributed structure and the gathering of data needed from soldiers.

2.1.3 Support for Processing

After the collection of data, the data has to be processed. The biggest problem of this process is the volume of data available to analysts. The capabilities to collect, communicate and store data is rising which results in a rising amount of data. According to G. M. Powell and Broome (2002) approximately 10,000 messages per hour are received and only 15,000 messages can be scanned per day (Jenkins & Bisantz, 2011). Data is also received in different formats (also depending on the intelligence department); images, written reports, verbal messages, numerical technical data. This data has to be processed in different ways, but in all cases the context is an important factor. Several solutions to the problem of data overload have been proposed. Patterson et al. (2001) discusses four commonly used approaches. The first approach is to reduce the available data. Less available data implies less processing. A problem with this approach is the context. In some contexts data elements are more important than in other. If these critical data elements are removed, the analysis and production will produce other results. Another problem with this approach is the narrow keyhole. If the data elements are not removed but pushed on more displays, the recognition and determination of relevant information is not easier. The second approach is to only show what is 'important'. This implies that the data set is divided in two or more 'levels of importance'. A problem with this approach is, again, context. It is difficult to assign individual elements to an importance dimension. Another problem is inhibitory selectivity. With different levels of importance, analysts are able to ask for data from a lower level, but will most likely not do that. In this lower level, the analysts, again, have to recognize and explore the data. The third approach is an approach where a machine computes what is important. This shifts the problem from the human to the system. Systems are not able to identify all relevant data within a context. The last approach is uses syntactic or statistical properties of text as cues to semantic content. If the data contains text, keyword search systems, web search engines and information visualization algorithms can use similarity metrics based on statistical properties of text to place documents in a visual space. A problem with this approach is that the system is often opaque. Analysts cannot know how good the relevance measure of the system is and can therefore over- or underrely on the system.

2.1.4 Support for Analysis & Production

The problem of data overload has a great influence on the processes of analyzing and production. If not all data is processed, this data cannot be analyzed and not used to answer the information requirements. According to G. M. Powell and Broome (2002) 1,000 of the 10,000 messages are superficially analyzed and only a few hundred are fully analyzed. Superficially analyzed messages are defined as messages for which the context cannot be properly considered, whereas fully analyzed messages are messages of which all reasonable implications are considered in the context. In a stability and support operation, 70% of the data received is based on HUMINT of which 5% of all incoming messages is fully analyzed. Furthermore, the data is highly complex (Jenkins & Bisantz, 2011). Within the dataset a high number of relationships is found. Intelligence analysts, therefore, have a cognitive challenge of associating the different pieces of data with one another.

Biermann, Chantal, Korsnes, Rohmer, and Ündeger (2004) proposes automated support with the use of semantic nets and ontological techniques of link analysis. Another solution is a wiki to provide a structured environment for analysis (Eachus & Short, 2011).

Another problem is that the world is highly dynamic (Jenkins & Bisantz, 2011). Situations unfold and evolve over time as well as domain factors and relationships (Roth et al., 2010). Solutions to this problem are to update a knowledge database in near-time with most recently received data (Jenkins & Bisantz, 2011), the recognition of updates which overturn previous information (Patterson et al., 2001) and timeline analysis (Patterson et al., 2001).

Another problem is that the data is qualified by meta-information (Roth et al., 2010). Standard types of meta-information are degree of trust in the source (source reliability), the likelihood that the information is accurate (credibility), actual and anticipated information age (staleness) and uncertainty. Jenkins and Bisantz (2011) identifies a lack of meta-information, whereas Pfautz et al. (2006) identifies three other problems. First, decision-makers may fail to recognize relevant meta-information. Second, decision-makers may not process meta-information appropriately. If an analyst cannot correctly integrate reliability and credibility of a message, the overall confidence in the message may be wrong. Third, the decision-maker may not properly utilize the meta-information in the integration of multiple information sources. A solution proposed by Jenkins and Bisantz (2011) is a support system that maintains and fuses meta-information during data association processes. Wright, Mahoney, Laskey, Takikawa, and Levitt (2002) proposes an uncertainty calculus as a supporting inference engine.

Intelligence analysts also have to deal with inconsistent data and a high degree of uncertainty (Jenkins & Bisantz, 2011). This can pose 'cognitive tunnel vision' or biases (Hutchins, Pirolli, & Card, 2004). Patterson et al. (2001) proposes solutions that should help analysts identify, track and revise judgments about data conflicts and aid in the search for updates on thematic elements. A Knowledge Environment for the Intelligence Analyst (KE-IA) that can account for knowledge generation and explanation by putting together a coherent situation description, alert the analyst when certain events of interest can be hypothesized, suggest answers to PIRs and evaluate them against the scenario is proposed by G. M. Powell and Broome (2002). This should be done by access to data bases, information push techniques and tools that support automated and user-directed Knowledge Discovery.

The last difficulty is that intelligence analysts should avoid prematurely closing the analysis process (Patterson et al., 2001). A proposed solution is to track 'loose ends' that need to be resolved later.

2.1.5 Support for Dissemination

A problem with the dissemination is that the analyst has to maintain understanding about the discipline, subject-environment and sources of information of the consumer in order to successfully communicate the intelligence report. Meta-information should also be made available to the commander, because meta-information affects a commander's situation awareness, decision-making performance, workload and trust (Pfautz et al., 2006). A decision making support for commander's should, therefore, be able to present meta-information in the display design, user interaction design and automation design (Pfautz et al., 2006).

2.1.6 Summary

If we look back at problems in the intelligence cycle, we can see that the problems with planning are solved with tools such as AICMS and JIGSAW. The proposed systems from McGrath et al. (2000) and Van Diggelen et al. (2012) could solve the problems with the collection. In the step of processing, the best solution is an approach in which syntactic and statistical properties of text are used as cues to semantic content. The system using this approach has to be transparent. In the step of analysis and production, many problems occur. The problem of a highly dynamic environment is solved with updating a knowledge database in near-time with most recently received data (Jenkins & Bisantz, 2011), the recognition of updates which overturn previous information (Patterson et al., 2001) and timeline analysis (Patterson et al., 2001). A support system proposed by Jenkins and Bisantz (2011) can maintain and fuse meta-information, but meta-information has to be available. The problem of information overload is not solved, but automated support is proposed by Biermann et al. (2004). The problem of inconsistent data and a high degree of uncertainty is solved on a high level by the KE-IA (Patterson et al., 2001).

What seems to be missing is a system that can deal with information overload and inconsistent data. The system of Van Diggelen et al. (2012) can be extended with a transparent system which uses the fourth approach of Patterson et al. (2001) and a tool that can use meta-information for analyzing information. In this way information overload is reduced, because in the system of Van Diggelen et al. (2012) answers to questions are directly put together. This also solves the problem of the high dynamic environment. With information about semantic content and meta-information inconsistent data can be analyzed. Systems that also deal with information overload and inconsistent data are question answering systems. The next section discusses question answering systems.

2.2 Question Answering Systems

Question Answering (QA) is the task where a system tries to automatically provide the correct information as an answer to an arbitrary question formulated in natural language (Moll & Vicedo, 2007). Question Answering is developed from two different scientific perspectives: Artificial Intelligence and Information Retrieval. From the field of Artificial Intelligence, question answering systems used knowledge encoded in databases. These systems were popular in the sixties and early seventies (Simmons, 1965) (Coles & Center, 1972) (Nagao & Tsujii, 1973) (McSkimin & Minker, 1977). Two of the best-known question answering systems from that time are BASEBALL (Green, Wolf, Chomsky, & Laughery, 1961) and LUNAR (Hirschman & Gaizauskas, 2001). BASEBALL was able to answer questions about baseball games played in the American league over one season. LUNAR answered questions about the analysis of rock samples from Apollo moon missions and was able to answer 90% of the questions within the domain (Hirschman & Gaizauskas, 2001). These systems were described as natural language interface to databases (Moll & Vicedo, 2007) or structured knowledge-based question answering systems, because the source of information was a database about a specific topic. In the later seventies and eighties research focused on theoretical bases in order to test general Natural Language Processing theories. This gave rise to large and ambitious projects such as the Berkeley Unix Consultant. This project used UNIX to develop a help system that combined research in planning, reasoning, natural language processing and knowledge representation.

Question answering systems became popular from the Information Retrieval point of view in 1999, because a QA track was introduced in the Text Retrieval Conference (TREC) competitions (Voorhees, 1999). The QA track is focused on text-based, open-domain question answering. Information Retrieval was focused on returning relevant documents as search engine machines. These systems are described as free-text based question answering systems. The last Question Answering Track ran in 2007, but was followed by two relevant tracks: a Crowdsourcing Track and a Web Track. The most popular question answering system nowadays is probably Watson developed by IBM's DeepQA project that won the quiz show *Jeopardy!* in 2011.

2.2.1 Generic Architecture of a QA System

Hirschman and Gaizauskas (2001) proposes a generic architecture for a question answering system (see figure 2). In the different question answering systems built over time not all steps or methods are used. In this generic architecture a natural question input is received from a user. This question is analyzed in order to get an appropriate form and interpretation of the question. With a preprocessed document collection the most likely documents to contain the answer to the question are selected and analyzed. After the analysis, the answer is extracted from each candidate document and a response is generated. Each of these steps will be discussed in the paragraphs below.

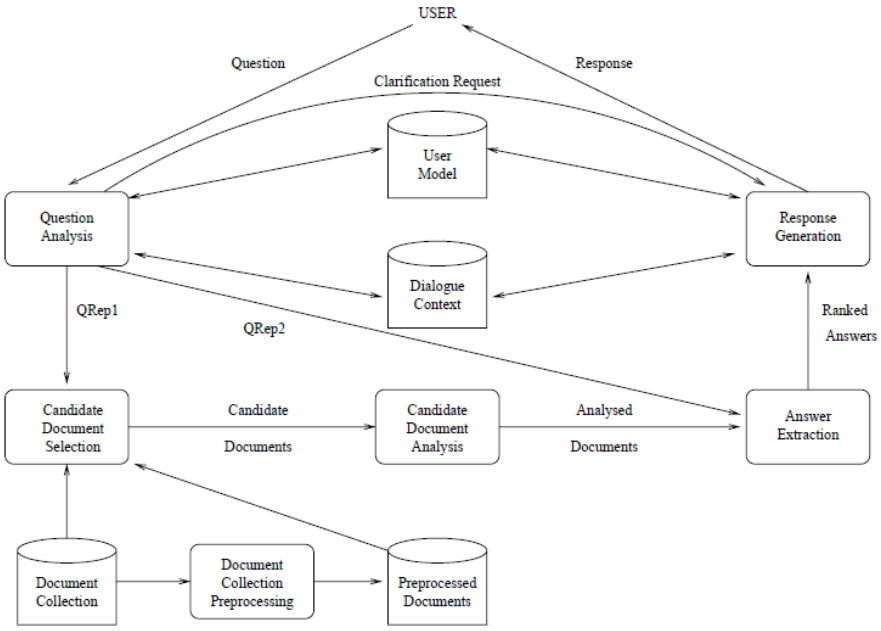


Figure 2: Generic architecture for QA proposed by Hirschman and Gaizauskas (2001)

Question Analysis

The first step of a question answering system is to analyze the question. Most of the time two actions are taken: 1) identifying the semantic type of the entity asked for in the question; 2) identifying key words or relations. The first action involves searching for key questions words (*when, what, why, who, where, which*) or placing the input question in a category in a predefined hierarchy of, for example, question types. The second action involves extracting key words (semantically or syntactically) or relations between the identified entity asked for in the question and other entities in the question. This collection of words can be expanded with, for example, synonyms and morphological variants of the words. Techniques used for this step are full-blown query expansion techniques, wide-coverage statistical parsers and robust partial parsers (Hirschman & Gaizauskas, 2001).

Document Collection Preprocessing

In order to answer questions online, it can be an advantage to preprocess the documents you are using for finding the answer to the question. Besides document indexing engines, logical representations (Aliod, Berri, & Hess, 1998), ternary relation expressions (Katz, 1997) or even shallow linguistic processing with tagging, named entity recognition and chunking can be done offline (Milward & Thomas, 2000) (Prager, Brown, Radev, & Czuba, 2001).

Candidate Document Selection

Once the question is analyzed and the documents are preprocessed, a selection of relevant documents for the question can be made. Two conventional methods from information retrieval are search engines that use a boolean system and search engines that use a ranking system. Well-known ranked retrieval engines are Okapi, Lucene and Z-PRISE (Saggion, Gaizauskas, Hepple, Roberts, & Greenwood, 2004). Okapi selects documents with a bm25 weighting scheme. Documents with a higher weight than a certain threshold are broken into passages. Each passage is scored and the document is re-ranked according to the score of the best passage of this document. Okapi returns only the best passage of each document. Lucene uses the standard tf.idf weighting scheme with the cosine similarity measure. The ranking algorithm prefers short passages that contain all question words over longer passages that do not contain all question words. The Z-PRISE IR engine uses a cosine vector model in order to select documents. This engine only extracts documents, not passages, that contain one or more of the key words (Moldovan et al., 1999).

Search engines that use a boolean system for document selection are for example MURAX (Kupiec, 1993), Falcon (Harabagiu et al., 2000) and LASSO (Moldovan et al., 1999). In MURAX a query is a conjunction of terms with a specified number of terms allowed between these terms and a specific order which is used to seek answers in Grolier's on-line encyclopedia. With the amount of passages that are returned the query is narrowed (decreasing number of hits) or broadened (increasing number of hits) until the number of hits is in a certain range or no further queries can be made. The hits are ranked according to the overlap with the original question. The Falcon system uses disjoint terms as morphological, lexical and semantic alternations for the query. LASSO uses both OR and AND operators as well as a PARAGRAPH operator. Only paragraphs that contain all keywords are used. For both types of search engines the restriction in the amount of returned documents is important as well as parameters in passage retrieval (length and window interval).

Candidate Document Analysis

In some cases it is necessary to analyze the candidate documents, especially if not all documents are fully preprocessed. Methods used are sentence splitting, part-of-speech tagging and chunk parsing (Hirschman & Gaizauskas, 2001). Some QA systems use a syntactic parser to map candidate documents into a logical or quasi-logical form prior to answer extraction (Aliod et al., 1998) (Scott & Gaizauskas, 2000) (Zajac, 2001). Even semantic role information (Hovy, Gerber, Hermjakob, Junk, & Lin, 2000) and grammatical relations (Buchholz & Daelemans, 2001) can be extracted in this step.

Answer Extraction

With candidate documents the answer has to be extracted. A good answer has to match the category (Abney, Collins, & Singhal, 2000) or type (Hirschman & Gaizauskas, 2001) and possible additional constraints. According to Lin et al. (2003) a good answer is represented in focus-plus-context style. This means that the answer must directly answer the question and provide additional contextual information.

The best answers are gathered and ranked in many ways depending on answer type and methods used for document extraction. The main stream of answer extraction technology uses feature based methods of sorting (Ramprasath & Hariharan, 2012) with the use of for instance neural networks (Pasça & Adviser-Harabagiu, 2001), maximum entropy, Support Vector Machines (Suzuki, Sasaki, & Maeda, 2002) and logistic regression (Li, Wang, Guan, & Xu, 2006). Another method is to use multi-stream QA, which focuses on the occurrences of answers in multiple streams (Ramprasath & Hariharan, 2012). Both probabilistic (Radev, Fan, Qi, Wu, & Grewal, 2005) and statistical models can be used. With multiple possible answers similar answers are identified and merged (Jijkoun & De Rijke, 2004), often through mining and tiling of n-grams (Azari, Horvitz, Dumais, & Brill, 2002) (Dumais, Banko, Brill, Lin, & Ng, 2002).

Response Generation

With a list of possible answers, a response to the question has to be generated. For TREC (Text Retrieval Conference) evaluations a ranked list of top five answers is generated, where each answer is a text string of a certain amount of bytes. The disadvantage is that these answers are most likely not grammatical, have too little context and are too complex or extensive for the user. Several important parts for this part are answer fusion, overlapping, contradiction, time stamping, event tracking and inference (Burger et al., 2001).

In order to get a better system, the response has to be evaluated. According to Breck et al. (2000) the following criteria should be considered in judging an answer: relevance, correctness, conciseness, completeness, coherence and justification. Past years the focus has been on relevance with the well-known methods precision and recall. Precision is the fraction of retrieved documents/answers that are relevant and recall is the fraction of relevant documents/answers that are retrieved. Other measures for relevance are fall-out and f-measure. In TREC often an answer key is available to judge answers or a person judges the answers.

2.2.2 Types of QA Systems

Although a generic architecture for question answering systems exists, different types of question answering systems can be distinguished. First, question answering systems can be build to answer questions in a specific or closed domain or an open domain. Question answering systems for a closed domain often use techniques such as an ontology from Natural Language Processing (Ramprasath & Hariharan, 2012). Question answering systems for an open domain often use techniques from Information Retrieval. Second, question answering systems can use documents or people to find the answer to the question. Document are used in most question answering systems, but question answering websites such as Quora, Yahoo! Answers, AnswerBag, Blurt it, Anybody out there, WikiAnswers, FunAdvice, Askville, Ask me helpdesk and Answer Bank use people. These people voluntarily ask and answer questions. The evaluation of answers from people is hard, because the information relative and unstable (Kim, Oh, & Oh, 2007). The 'best' answer can, however, not only be determined by the system itself but also by the person who asked the question or by other people by placing the question-answer pair in a voted stage (Liu, Chen, Kao, & Wang, 2012). Third, question answering systems can be implemented from different underlying models for document representation (Baeza-Yates, Ribeiro-Neto, et al., 1999). In the first kind of models set-theory is used to represent documents as a set of words or phases. Common models of this kind are the Standard Boolean model, the Extended Boolean model and the Fuzzy Set model. The second kind of models is algebraic and represents documents as vectors, matrices or tuples. Common algebraic models are (Generalized) Vector Space models, Latent Semantic Indexing models and Neural Network models. The third kind of models is probabilistic. These models compute the probability that a document is relevant for a query. Probabilistic theorems as Bayes are often used. Common models are Bayesian Networks, the Binary Independency Model and the Inference Network Model. A relatively new kind of models is feature-based (Ramprasath & Hariharan, 2012). Neural networks, maximum entropy, Support Vector Machines and logistic regression belong to this kind of models.

2.2.3 Obstacles in QA

Several obstacles exist for question answering systems. Many obstacles stem from the general problem of natural language understanding. In order to get a good mechanization of question-answering, understanding of natural language, especially the meaning of concepts and propositions drawn in natural language, is necessary. This brings us to the first obstacle: world knowledge. In order to be able to understand the meaning of concepts a system has to have world knowledge. Humans acquire world knowledge through experience, education and communication. Much world knowledge is perception-based and therefore intrinsically imprecise, which is a problem for systems. A second obstacle is deduction from perception-based information or precision of meaning. Many concepts have an imprecise meaning as *few* and *many*. New tools as Precisiated Natural Language, Protoform Theory and Generalized Theory of Precisiation of Meaning could be used in order to tackle this obstacle. A third obstacle is the concept of relevance. Relevance cannot be bivalent, but is a matter of degree (Zadeh, 2006).

The concept of relevance can be approached with semantics or statistics. A fourth obstacle occurs primarily in question answering websites. A lot of questions are waiting to be answered and people who can answer the question may not find the question, which can result in obtaining non correct answers. The solution to this problem could be to find experts automatically. The first attempts were done with link analysis approaches. Influencing approaches were HITS (Kleinberg, 1999) and PageRank (Page, Brin, Motwani, & Winograd, 1999). In finding an expert for a particular question user subject relevance, user reputation and authority of a category are important (Liu et al., 2012). User subject relevance is the relevance of the domain knowledge of the user to the question. This relevance is measured with content and quality measures as voting and evaluation of the historical answers given in that category. A user reputation is also derived from the previous question-answering records, but is in this case based on the ratio of the user's answers that are marked as the best answers (Liu et al., 2012). The authority of category looks at the link analysis of the category based asker-answerer network.

2.2.4 Standards in QA

According to Burger et al. (2001) the following standards are necessary for real-world users of question answering systems. The first standard is timeliness. Users want real-time answers with the most recent information to a question. The second standard is accuracy. Incorrect answers are worse than no answers, according to Burger et al. (2001). Question answering systems should, therefore, incorporate world knowledge and use a mechanism that is able to mimic common sense inference. The third standard is usability. The question answering system has to have ontologies (domain specific and a connection to open-domain ontologies) and a mechanism to deliver answers from different data source formats in any desired format. Users must have the opportunity to describe the context of the question. The fourth standard is completeness. Users want a complete answer to their question and answer fusion is, therefore, necessary. Reasoning with world knowledge, domain-specific knowledge, implicatures and analogies can be incorporated in question answering systems. The fifth and last standard is relevance. If answers are not relevant to the real-world user, users will not use the QA system. In the evaluation of a QA system relevance is often the most important factor, as mentioned before.

3 Support Model

As mentioned in subsection 2.1.6, a system that can deal with information overload and inconsistent data is needed. A combination of the system of Van Diggelen et al. (2012), a transparent system that can use syntactic and statistical properties of text as cues to semantic content (Patterson et al., 2001) and a tool that can use meta-information for analyzing information can be used. The support model proposed in this thesis uses a trust-based question answering system, named T-QAS (see Figure 3). The question answering system uses open domain knowledge, people and probability measures, because intelligence analysts have to deal with the same options. A support system is proposed instead of an automated system, because humans have natural language understanding and world knowledge which is a major problem for question answering systems. The differencing element of the proposed question answering system as support model for intelligence analysts is the process *Trust Management* which keeps track of trust in the difference sources. This trust is used in both other processes of T-QAS.

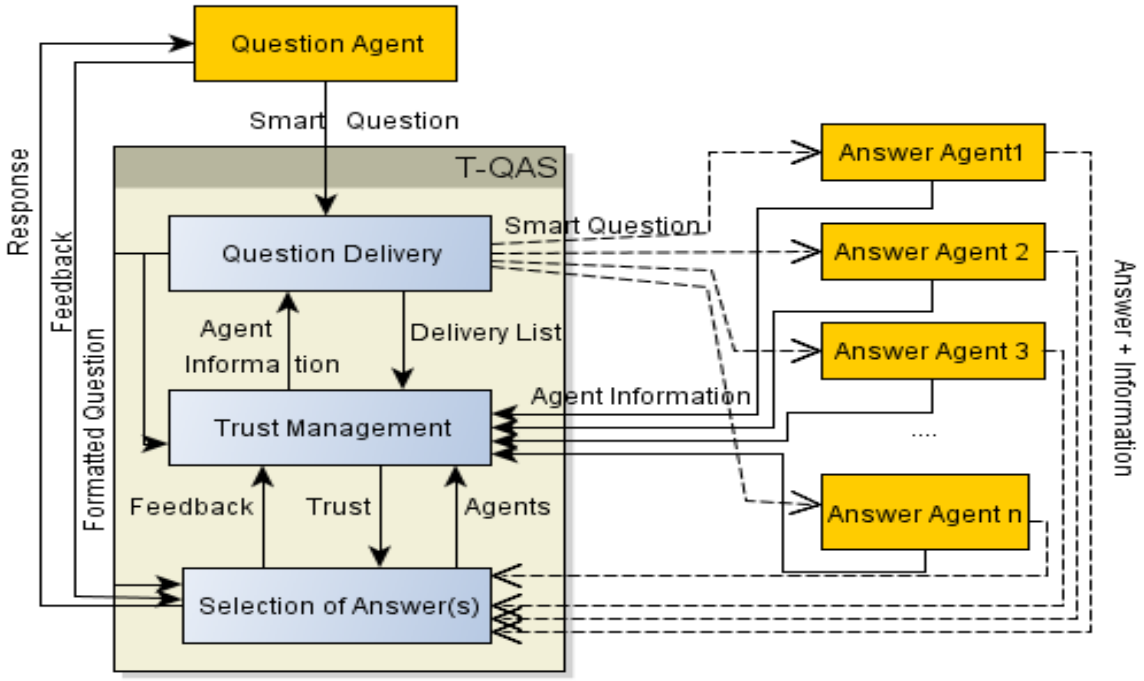


Figure 3: Proposed general model

This support model can deal with both strategic and operational intelligence, but the focus will be on operational intelligence because of the current and near-time character. This support model is mainly for the professional intelligence analyst from human sources, because the biggest problems remain in this field. Other intelligence departments could also use this model. All kinds of questions and answers can be used in this model.

As is shown in Figure 3, T-QAS is provided with a smart question from the question agent (the intelligence analyst). The smart question is received by the process *Question Delivery*. This process manages the delivery of questions to answer agents. The question is formatted and send to the processes *Selection of Answer(s)* and *Trust Management*. From this last process information about each agent such as availability and location is send back to the process *Question Delivery*. With this information the appropriate answer agents can be selected. A delivery list is send to the process *Trust Management*. In this process trust about the agent is managed. Information about agents such as location is automatically send to this process. Agents could be people, but also sensors or texts from the World Wide Web. When the appropriate answer agents are selected in the process *Question Delivery*, the smart question is send to these answer agents. The other agents do not receive the question. Answer agents can give an answer to the question. This answer with information such as ID of the agent, the current location and the time of answering is send to the process *Selection of Answer(s)*. In the process *Selection of Answer(s)* the possible answers are weighted, tiled and ranked with use of trust in each of the agents gathered from the process *Trust Management*. A response is sent to the question agent. The question agent gives feedback about the response such as confirmed or rejected, which is received by the process *Selection of Answer(s)* and sent to the process *Trust Management*. The different processes from this model are further explained in the following sections. An example is given in section 3.4.

3.1 Question Delivery

Zooming in on the process of *Question Delivery* new processes appear (see Figure 4). When a user has sent a smart question, this question is first analyzed in the process *Question Analysis*. The formatted question which contains the smart question with tags, key words and the category of the answer is sent to the process *Agent Selection*, *Selection of Answer(s)* and *Trust Management*. Agent information from the process *Trust Management* is used to select appropriate answer agents. A delivery list is sent to the process *Trust Management*. For the selected answer agents the best modality and time for presenting the question is selected in the processes *Modality Selection* and *Time Selection* and at the best time with the best modality the smart question is sent to the answer agent. The processes *Modality Selection* and *Time Selection* are inspired by the idea developed by TNO that the Right Message should be delivered at the Right Moment in the Right Modality (Van Diggelen et al., 2012).

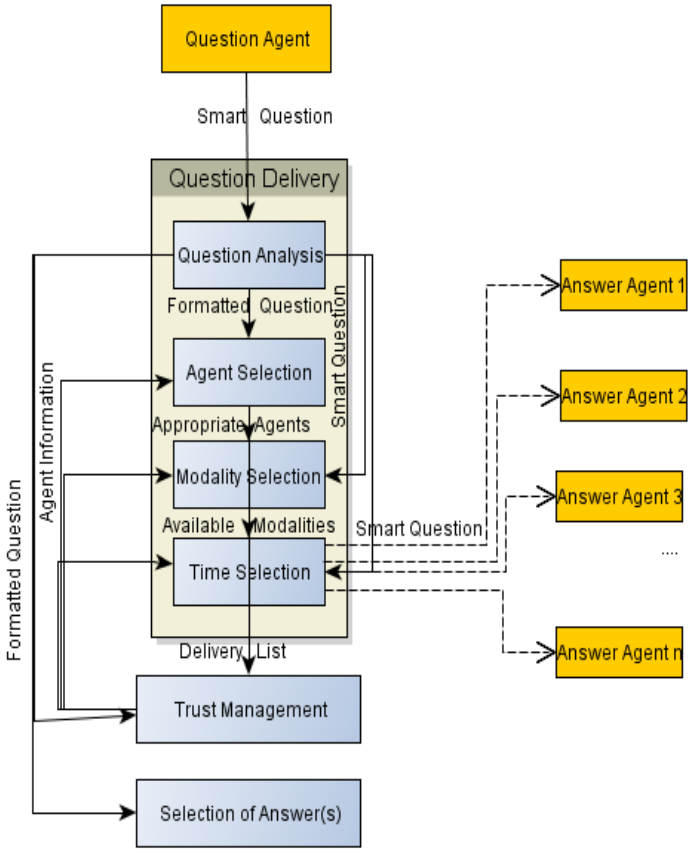


Figure 4: Question Delivery

3.1.1 Question Analysis

For the process *Question Analysis* the two actions from the generic architecture of a QA system (see section 2.1.1) are taken. First, the (semantic) type or category of the entity asked for in the question is identified. This is done by searching for key question words, key words and words after the key question word. Key words are defined as the nouns and verbs found with Part-of-Speech tagging and the named entities found with Named Entity Recognition. The key question words give a general category of the entity searched for. The key words further determine the category. Missing tags such as location and domain could be filled in with key words. The analyzed question is then put in the format that could be used in the processes *Agent Selection*, *Trust Management* and *Selection of Answer(s)*. This format contains the smart question with tags, key words and category of the answer.

3.1.2 Agent Selection

In the process *Agent Selection* the appropriate answer agents are selected. Which answer agents are appropriate depends on the question and the context. Appropriate agents have to be available before the question is expired, be able to go to the area in which the answer has to be found and have a security level that is not lower than the security level presented with the question. If too much agents are appropriate, trust and domain of expertise can further determine the most appropriate agents. The appropriate agents are selected with agent information gathered from the process *Trust Management* and put in a delivery list, which is sent to the processes *Modality Selection* and *Trust Management*. The process *Agent Selection* is similar to *Candidate Document Selection* of the general architecture of question answering systems (section 2.2.1).

3.1.3 Modality Selection

When the appropriate answer agents are selected, the appropriate modality has to be selected. With use of the available modalities and preferences of each agent gathered from the process *Trust Management* and the modalities in which the question can be displayed (audio, text et cetera) a ranked list of available modalities for each agent can be made. This information is added to the delivery list and send to the process *Time Selection* and *Trust Management*.

3.1.4 Time Selection

In the process *Time Selection* the smart question is send to each agent from the delivery list in the best modality gathered from the process *Modality Selection*. The best time to send the question has to be determined. In principle the best strategy is to send the message as soon as possible, because in this case the answer agent has the most time to find the answer to the question. This implies that the smart question is directly sent in the best modality that is currently available for an agent at the first minute the agent is available. This message can later be popped up.

3.2 Trust Management

The process *Trust Management* consists of a database with information about all agents, a database with information about questions asked before and answers given to that question and the process *Trust Maintenance* (see Figure 5). Agents send information (automatically) to the process *Trust Management* and this information is stored in the Agent Information database. If a formatted question is received from the process *Question Delivery*, the formatted question is stored in the Question Answer (QA) Information database. From the Agent Information database information about all agents is gathered. This information is sent to the process *Question Delivery*. From this process a delivery list is sent. The delivery list is added to the QA information. If the answer agents have given the answers, the IDs of the agents are sent from the process *Selection of Answer(s)* to the process *Trust Maintenance*. Trust in each of these agents is sent back. If the end of the process of question answering, the question agent gives feedback. This feedback is sent to the process *Trust Maintenance* and trust is adjusted. Feedback and adjusted trust values are sent to the QA Information database and the Agent Information database, respectively.

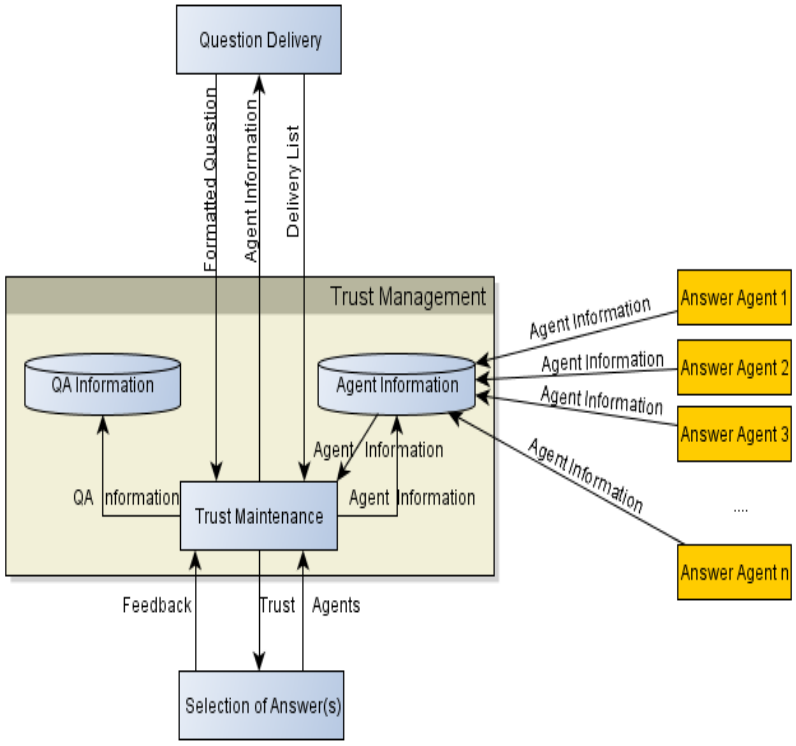


Figure 5: Trust Management

3.2.1 Agent Information

The Agent Information database can best be seen as a social network. The nodes or agent have information as shown in Table 5-2 of Pfautz et al. (2006). The edges or relations have information about the relation between agents. If the process *Trust Maintenance* has received a formatted question information about the agents is gathered from the database. Information in the nodes and edges is adjusted with feedback and information from the answer agents.

3.2.2 QA Information

The database with QA Information contains information about questions and answers such as the question agents and answer agents dealing with a certain question and feedback.

3.2.3 Trust Maintenance

In the process *Trust Maintenance* trust in each agent is calculated and selected. If a formatted question is sent, information about agent is sent through this process to the process *Question Delivery*. The formatted question is then put in the QA Information database. Changes of location or availability are also sent to the process *Question Delivery*. If a list of agents is sent from the process *Selection of Answer(s)*, trust in each of these agents is gathered from the Agent Information database and is sent back to the process *Selection of Answer(s)*. If feedback is sent from this process, this feedback is put in the QA database with the question and trust is changed.

An independent trust model (as described in Van Maanen (2010)) is used to estimate the trust value of an agent. This trust value is based on experiences between the agent and, in this case, the system. The influence of past experiences is be varied with a decay factor λ.

The estimation of trust in an agent at a certain time is calculated with the following formula (an adjusted formula from Van Maanen (2010)):

$$T_i(t) = T_i(t-1) \cdot \lambda_i + (E_i(t) \cdot (1 - \lambda_i)) \tag{1}$$

where $T_i(t-1)$ is trust in agent i at time $t-1$, λ_i is a decay parameter, $E_i(t)$ is the experience for agent i at time t, where $0 \leq E_i(t) \leq 1$.

3.3 Selection of Answer(s)

The process *Selection of Answer(s)* is shown in Figure 6. The answers with information such as ID of the agent, location and time of answering are stored in a list of possible answers. This list is sent to the process *Answer Tiling*. In this process the answers are tiled. This means that the same answers are put together. The tiled answers are sent to the process *Answer Ranking*. In this process the different answers are ranked. In order to rank the answers information about the question and the agent is needed from the process *Question Delivery* and *Trust Management*, respectively. The ranked answers are sent to the process *Response Formulation*. In this process the response is formulated and sent to the question agent and feedback received from the question agent is sent to the process *Trust Management*.

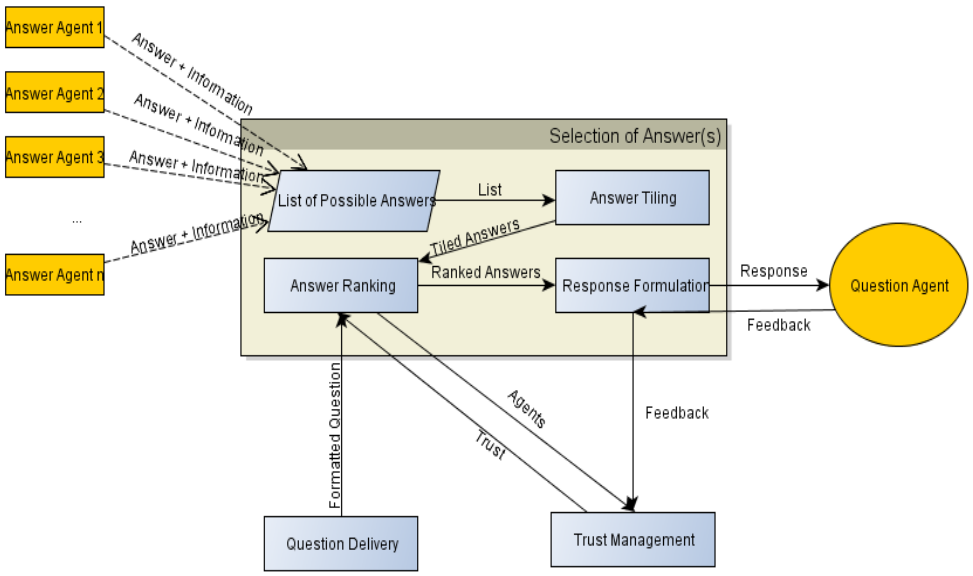


Figure 6: Selection of Answer(s)

3.3.1 Answer Tiling

Answer Tiling is part of the process *Answer Extraction* in a lot of question answering systems (Jijkoun & De Rijke, 2004) (Azari et al., 2002) (Dumais et al., 2002). In this process multiple possible answers are identified and merged. For the identification of similar answers, the definition of Jijkoun and De Rijke (2004) is used. In this definition two answers are considered similar if the strings are identical, one answer is a substring of another answer or the edit distance between the strings is small compared to their length. Furthermore, longer answers are constructed from sequences of overlapping shorter n-grams (Dumais et al., 2002). If the answer is a number only the exactly matching answers are tiled. The similar answers are now considered as one answer given by multiple agents.

3.3.2 Answer Ranking

Lopez et al. (2009) uses three different methods to rank: ranking by semantic similarity, ranking by confidence and ranking by popularity. T-QAS uses a combination of ranking by confidence and ranking by popularity. Ranking by confidence takes trust in each of the answer agents into account. Ranking by popularity takes the amount of agents that gave a similar answer into account. The ranking of answers is done with an estimated value of each answer. The estimated value of the answer (EV_a) is calculated with the following formula:

$$EV_a = 1 - \prod_{i=1}^{n}(1 - T_i(t)) \tag{2}$$

where n is amount of agents who gave answer a and $T_i(t)$ is trust in agent i at time t.

If for example two sources, one with a trust value of 0.7 and one with a trust value of 0.4, give answer a, the estimated value is:

$$EV_a = 1 - ((1 - 0.7) \cdot (1 - 0.4)) = 0.82 \tag{3}$$

A complement of both the trust value and the product is taken, because otherwise the estimated value is lower instead of higher in case of multiple agents (ranking by popularity):

$$EV_a = \prod_{i=1}^{n} T_i(t) = (0.7 \cdot 0.4) = 0.72 \tag{4}$$

With the estimated value a ranking can be made. The ranked answers with answer agents, trust and the estimated value are sent to the process *Response Formulation*.

3.3.3 Response Formulation

With the ranked answers a response has to be formulated. This response depends on the preferences of the question agent. Preferences could include the amount of answers given, the presentation of ranking, the presentation of the estimated value and an evaluation. In this model the ranked answers with answer agents and the trust value of each agent are sent to the question agent in order to maintain transparency of the system.
When the response is sent to the question agent, the question agent gives feedback. This feedback is send to the process *Trust Management*.

3.4 Example

In this section an example is given to illustrate the support model. An intelligence analyst (question agent) is investigating the potential hostility in an area. One of the questions the intelligence analyst wants answered is: What is the number of armored vehicles in area X? The smart question is displayed in Figure 7.

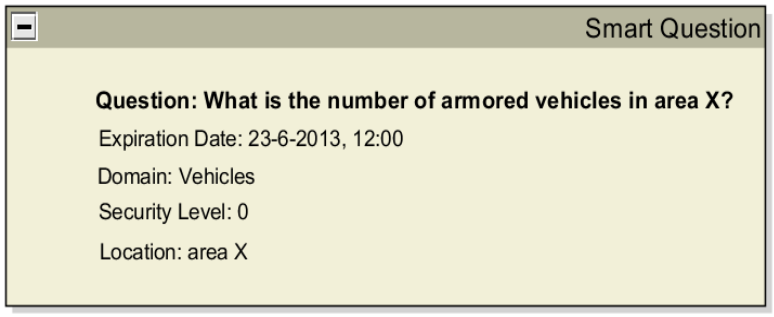


Figure 7: Example of Smart Question

The smart question is sent to T-QAS and received by the subprocess *Question Analysis* of the process *Question Delivery*. In this process the answer category and key words are identified and put with the smart question to create the formatted question (see Figure 8).

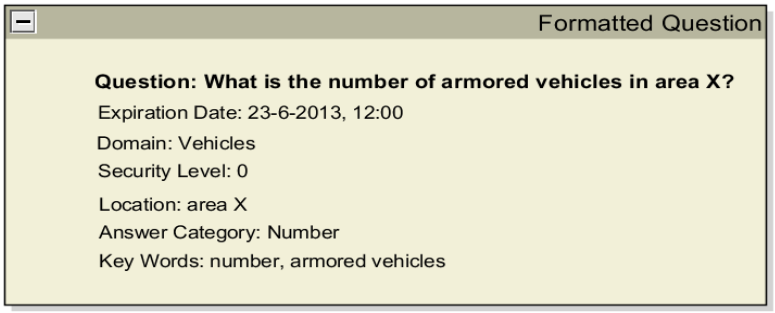


Figure 8: Example of Formatted Question

The formatted question is sent to the processes *Agent Selection*, *Trust Management* and *Selection of Answer(s)*. From the process *Trust Management* agent information is sent to the process *Agent Selection*. This agent information is the information of the nodes from the Agent Information database, as presented in Figure 9.

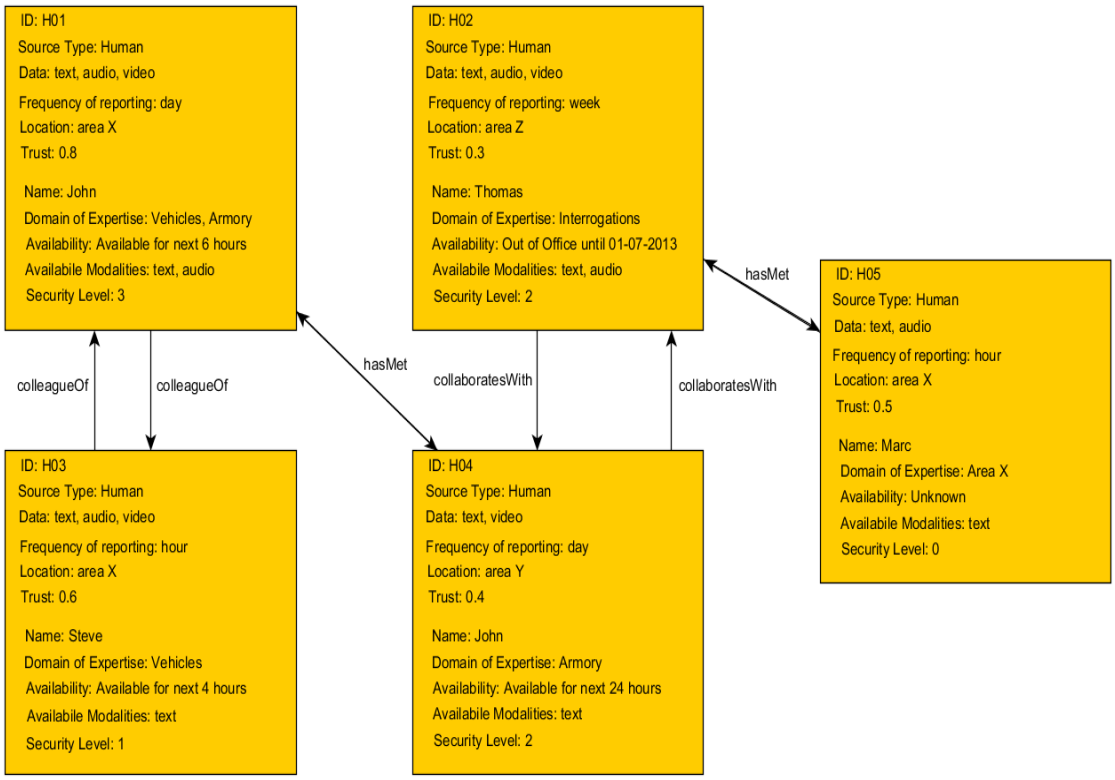


Figure 9: Example of Agent Information Database

In the process *Agent Selection* appropriate agents are selected. Appropriate agents are agents that are available before the expiration date is exceeded and have at least an equal security level as asked for in the smart question. In this example only H02 is not appropriate, because of the availability. A list with appropriate agents is sent to the process *Modality Selection*. From the agent information the available modalities are put with the appropriate agents. The first modality is the modality preferred by the answer agent, so all appropriate agents receive the smart question in the modality *text*. In the process *Time Selection* the appropriate time is selected. In this example the smart question is directly sent, because H01, H03 and H04 are currently available and H05 has an unknown availability. In this example H01 answers with *20*, H03 and H04 give the answer *15* and H05 does not answer (see Figure 10).

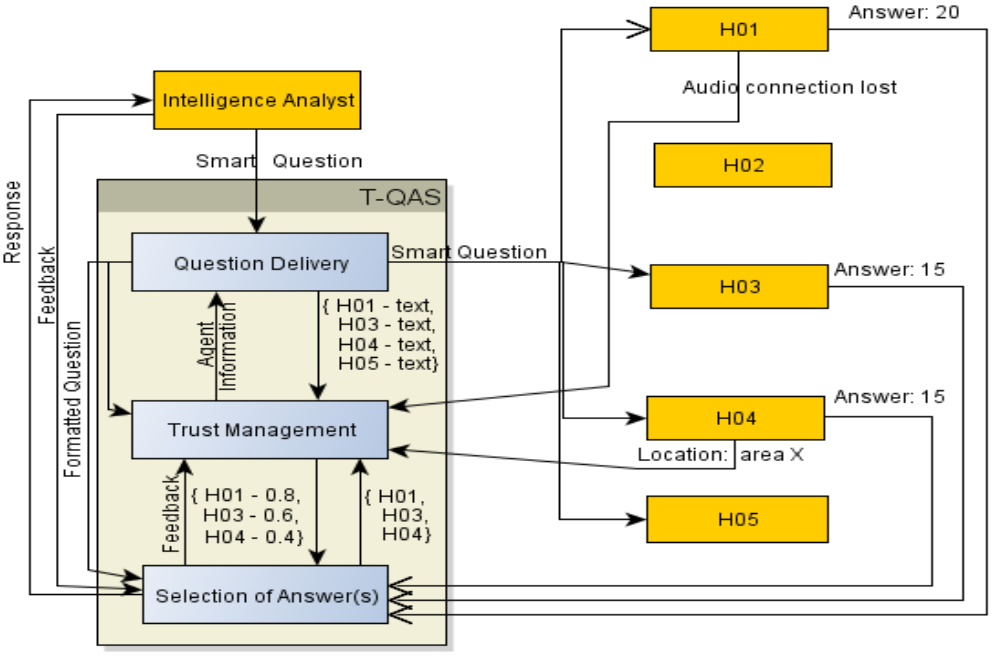


Figure 10: Example of Support Model

In the mean time H04 is moved from area Y to area X and the audio connection of H01 is lost. This information was sent to the Agent Information database in the process *Trust Management*. In the process *Selection of Answer(s)* the answers are tiled, so only two answers (15 and 20) remain. These two answers (and the corresponding agents) are sent to the process *Answer Ranking*. The ID of the agents is sent to the process *Trust Management* and the trust values are sent back to the process *Selection of Answer(s)*. With these trust values the estimated values of the answers is calculated:

$$EV_{20} = 1 - (1 - 0.8) = 0.8 \tag{5}$$

$$EV_{15} = 1 - ((1 - 0.6) \cdot (1 - 0.4)) = 0.76 \tag{6}$$

Answer *20* is thus the estimated best answer and answer *15* is the estimated second-best answer. The ranked answers are sent to the process *Response Formulation*. The intelligence analyst receives the response as presented in Figure 11.

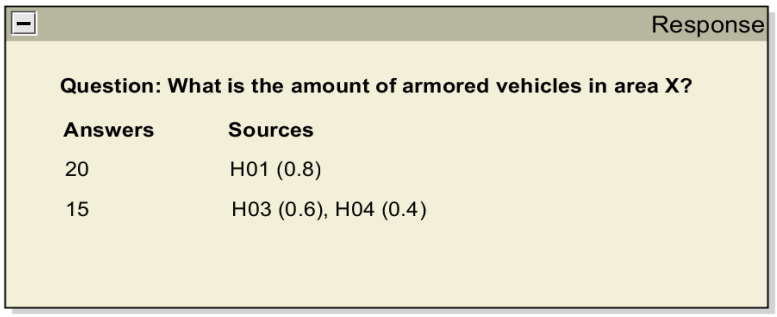


Figure 11: Example of Response

The intelligence analyst sends feedback. In this case the feedback is a confirmation of the ranking. The response and feedback is sent to the process *Trust Management*. In this process the trust values are changed with the experience. In this case the answer *20* was the best answer and therefore a positive experience (1) with that source. The answer *15* is the worst answer, so this is a negative experience (0). With the formula explained in section 3.2.3 the trust values are changed. λ is in this example 0.7:

$$T_{H01} = 0.8 \cdot 0.7 + (1 \cdot (1 - 0.7)) = 0.86 \tag{7}$$

$$T_{H03} = 0.6 \cdot 0.7 + (0 \cdot (1 - 0.7)) = 0.42 \tag{8}$$

$$T_{H04} = 0.4 \cdot 0.7 + (0 \cdot (1 - 0.7)) = 0.28 \tag{9}$$

The changed trust values are placed in the Agent Information database.

4 Implementation

The size of the model makes implementing the whole support model not feasible for this thesis, so two stages of the support model are implemented. The process *Trust Management* is the most important and differentiating part of T-QAS, so this process is the first stage of the implementation. In order to use this process a small part of the process *Selection of Answer(s)* is implemented. In the second stage a larger part of the process *Selection of Answer(s)* is implemented, because in previous research (Van Diggelen et al., 2012) (McGrath et al., 2000) the process *Question Delivery* has already been implemented and implementing a larger part of the process *Selection of Answer(s)* is more likely to help intelligence analysts. In this implementation the degree of trust in the source is used as meta-information in the process *Selection of Answer(s)* to deal with inconsistent information and a high degree of uncertainty. The implementations of stage One and Two are further explained in the next sections. C# is used for both implementations.

4.1 Stage One

Figure 12 shows the processes of stage One. In this stage tiled answers with the names of the agents that provided each of the answers are sent to the process *Response Formulation*. The names of the agents are sent to the *Trust Management*. In the process *Trust Maintenance* trust in each agent is gathered from the *Agent Information Database*, implemented as a dictionary. The trust values and the corresponding names of the agents are sent back to *Response Formulation*. In this process the answers, agents and trust values are transformed in a response for the question agent. The question agent can provide feedback about the response, such as the correct ranking of the answers.

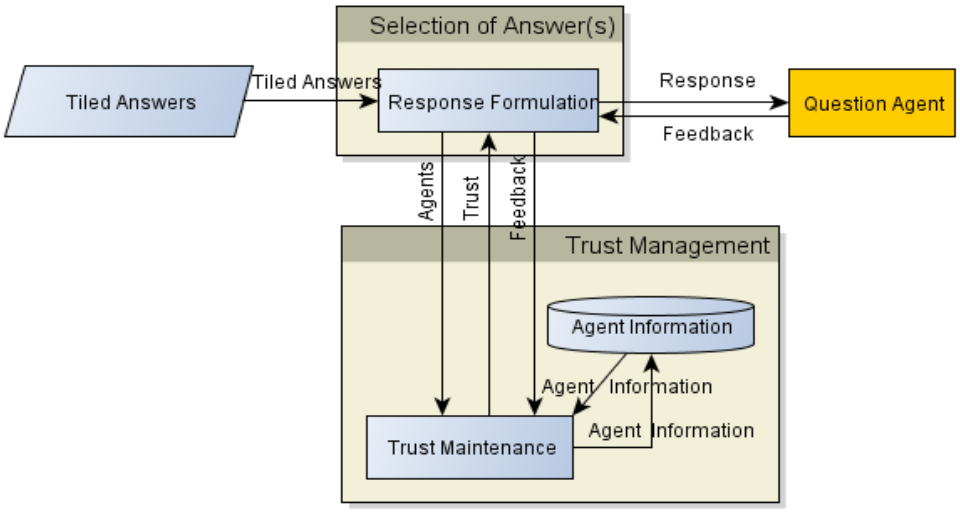


Figure 12: Processes Stage One

Feedback about the answers with agents that provided the answer is send to the process *Trust Management*. In the *Trust Maintenance* the trust values of the agents can be adjusted with the adjusted formula from Van Maanen (2010) as explained in 3.2.3. For experience the following formula is used (see also Table 1):

$$E_i(t) = ((r-1)/(m-1)) \cdot -1 + 1 \tag{10}$$

where r is the rank number of the answer of agent i, m is amount of answers.

If one answer is given, the experience is 1.

Amount of Answers	r = 1	r = 2	r = 3	r = 4	r = 5
2	1	0			
3	1	0.5	0		
4	1	0.66	0.33	0	
5	1	0.75	0.5	0.25	0

Table 1: Experience table

4.2 Stage Two

In stage Two one process is added in comparison to stage One (see Figure 13 for the processes of stage Two). The tiled answers are sent to the process *Answer Ranking* instead of directly to the process *Response Formulation*. In the process *Answer Ranking* the trust values of the agents are gathered in the same way as in stage One. With these trust values an estimated value of each answer is calculated as formulated in 3.3.2. The answers, estimated value, agents and trust values are sent to the process *Response Formulation* and transformed in a response for the question agent. Feedback is processed in the same way as stage One.

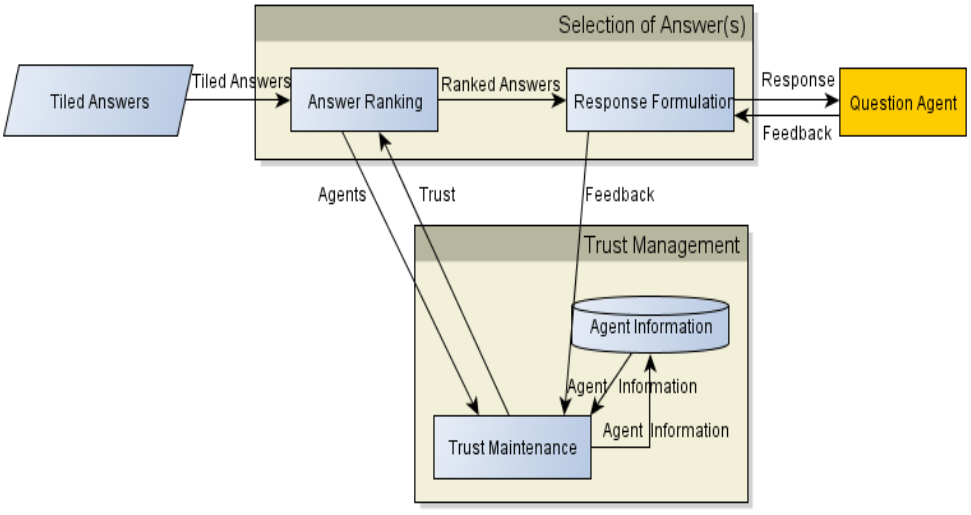


Figure 13: Processes Stage Two

5 Hypotheses

In order to determine if trust-based support helps the intelligence analyst, several hypotheses can be formed. In the hypotheses we want to compare the human alone (H) with a team consisting of the person and the stage One implementation of T-QAS (T1) , a team consisting of the person and the stage Two implementation of T-QAS (T2) and the stage Two implementation of T-QAS alone (S). The following sections explain the hypotheses about task performance, trust estimation performance and workload.

5.1 Task Performance

If T-QAS helps the intelligence analyst in the task of decision making, performance on this task has to be increased with support compared to without support.

Hypothesis 1. Task performance in T1 and T2 is higher than task performance in H.

As mentioned before, T-QAS has, unlike humans, no natural language understanding. A cooperation of a human and T-QAS should, therefore, have a higher performance than T-QAS alone.

Hypothesis 2. Task performance in T1 and T2 is higher than task performance in S.

T2 cannot only support the person in estimating trust values but can also give an estimation about the ranking of the answers. T1 can only support the person in estimating trust values and leaves the interpretation of these trust values to the human. T2 can, therefore, support an additional step in the decision making process and probably increase task performance more than T1.

Hypothesis 3. Task performance in T2 is higher than task performance in T1.

Figure 14 shows hypotheses 1, 2 and 3. An important factor determining the previous two hypotheses is performance in H. Good performance in H may result in a lower improvement of performance in T1 and T2, because T-QAS cannot add as much value to good performers compared to poor performers. Poor performers may benefit more from T-QAS, because they are not able to perform the task good themselves.

Hypothesis 4. The difference in task performance between H and both T1 and T2 is higher for poor task performers than for good task performers.

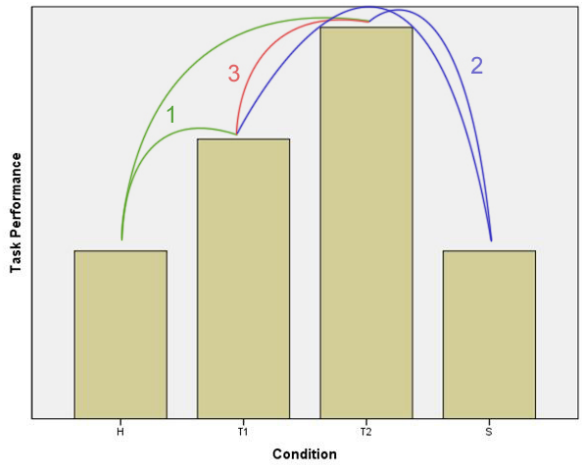


Figure 14: Hypotheses 1, 2 and 3

Previous hypotheses indicate a low influence of compliance on task performance, because S is hypothesized as good as H. If both a good performer in H and a poor performer in H comply to T-QAS in every case in which T-QAS is correct, the good performer will have a higher performance than a poor performer. In all cases T-QAS is not correct, performance is influenced by the human. This is the reason why task compliance is not included in the hypotheses.

5.2 Trust Estimation Performance

As mentioned before, the hypothesis is that T-QAS is good at objectively estimating trust values, whereas a human is not. In the task of estimating trust a human-system team should, therefore, outperform the human alone.

Hypothesis 5. Trust estimation performance in T1 and T2 is higher than trust estimation performance in H.

Trust estimation performance of both T1 and T2 is not higher than S, because T-QAS is good in objectively estimating trust values and the human cannot outperform T-QAS.

Hypothesis 6. Trust estimation performance in T1 and T2 is not higher than trust estimation performance in S.

Both implementations of the T-QAS use the same calculations for the trust value estimation, so no performance differences between a team with either implementation should occur.

Hypothesis 7. Trust estimation performance in T1 not different from trust estimation performance in T2.

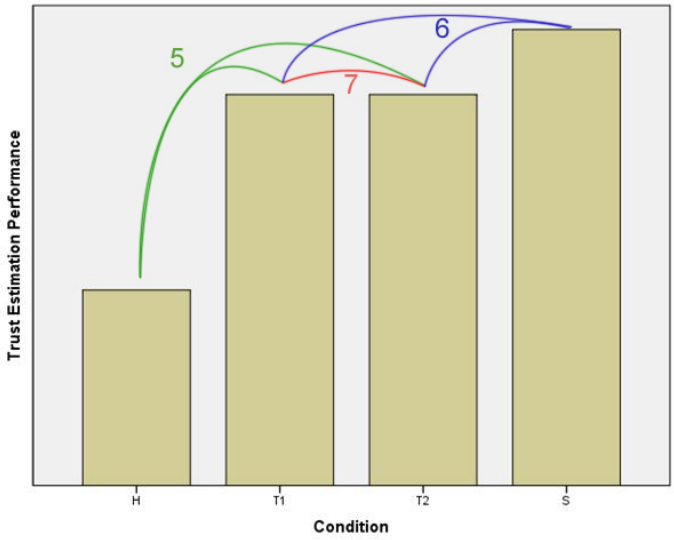


Figure 15: Hypotheses 5, 6 and 7

Figure 15 shows hypothesis 5, 6 and 7.

Compliance is in this case an important factor, because hypothetically will more compliance lead to a better performance, because S is hypothesized better than H. Performance in H will have less influence, because both poor and good performers can benefit from T-QAS.

Hypothesis 8. Trust estimation performance and compliance are positively related to each other.

5.3 Cognitive Workload

Not only performance could be increased with T-QAS, but cognitive workload could also be reduced. Reducing cognitive workload will also help intelligence analysts in their task. In T2 an additional step is supported compared to in T1, which will probably result in a lower workload. The hypotheses concerning workload are (see also Figure 16):

Hypothesis 9. Cognitive workload in T1 and T2 is lower than cognitive workload in H.

Hypothesis 10. Cognitive workload in T1 is higher than cognitive workload in T2.

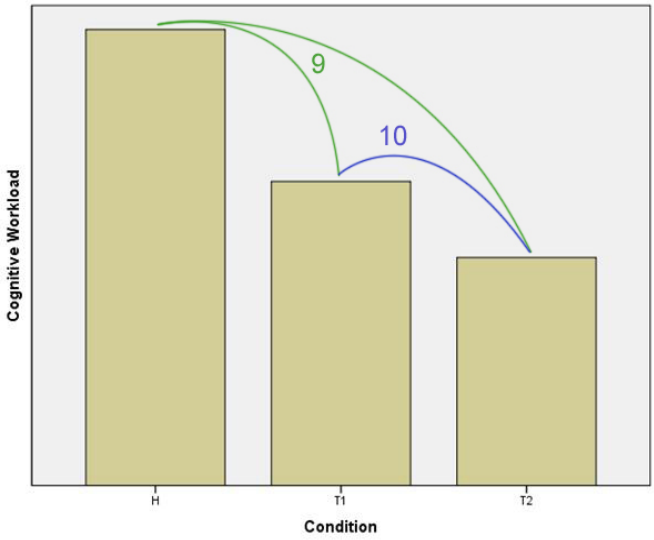


Figure 16: Hypotheses 9 and 10

6 Method

In order to test the implemented stages of T-QAS as described in chapter 4 and the hypotheses stated in chapter 5, an experiment is executed. This chapter describes the method of this experiment.

6.1 Participants

Thirty-six participants aged between eighteen and thirty-five (M = 24.2, SD = 4.3; 15 male, 21 female) with a higher education level participated in the experiment as paid volunteers. The participants are selected from the database of TNO Human Factors. No special training or military experience is required. The participants are not dyslectic, have no concentration problems and have no RSI.

6.2 Task

In order to test both stages of T-QAS and the hypotheses, the task has to meet several demands. The task has to be realistic for an intelligence analyst, so time constraints, inconsistent information due to unreliable sources, uncertainty, a highly dynamic environment and high workload have to be available. The task also needs an estimation of trust and context, because the hypothesis is that T-QAS is good at objectively estimating trust values but has no natural language understanding and world knowledge, whereas the participant can understand natural language and has world knowledge but is not very good at objectively estimating trust values. With context common world knowledge is created in order to give participants the same background.

The LOWLAND scenario from the Dutch Defence Intelligence and Security Institute (DIVI) is used for the task as it meets the demands described above.

6.2.1 Task Description

In the experiment participants had to form an accurate intell picture of the situation in a certain area in LOWLAND, the same manner in which an intelligence analyst would do so. The task had to be simplified, because the participants had no experience with analyzing intelligence. The task was set up as a quiz in which participants had to rank the given answers to a question. No further analysis or implications of this ranking had to be made by the participants. Feedback about the ranking is given, because intelligence analysts experience the consequences of their decisions. The quiz can, therefore, be seen as an accelerated version of the work of an intelligence analyst. Participants had not much time to rank the answers, so time constraints are available. Two, three, four or five answers were be presented to create different levels of inconsistent information. Both context information and estimation of trust values of different sources were available in the task. The participants had to keep a record of these trust ratings themselves. Uncertainty was available, because no feedback about the trust ratings was given.

The sources had a certain overall reliability, which resulted in giving a certain answer. In each condition each question has five answers with a certain correctness (see Table 2) and six sources with a certain reliability (see Table 3). With the following formula the probability that a source gives a certain answer is calculated:

$$P_a = (1 - R_s) \cdot 1/m + (R_s \cdot C_a) \tag{11}$$

where a is the answer, R_s the the reliability of the source, m is the amount of answers (5) and C_a is the correctness of the answer.

With this probability the amount of times a source gives a certain answer is calculated for 10 practice questions and 40 test questions. Given the amount of times a match is gathered between the sources and the answers in a way that two, three, four and five answers are given 10 times in the test questions. In the task it was possible that a source gives the best answer, followed by giving the worst answer.

Answer	**Correctness**
1	0.6
2	0.3
3	0.06
4	0.03
5	0.01

Table 2: Answers

Source	Reliability Scenario 1	Reliability Scenario 2	Reliability Scenario 3
1	0.9	0.9	0.94
2	0.8	0.75	0.85
3	0.7	0.65	0.7
4	0.55	0.5	0.5
5	0.4	0.3	0.4
6	0.2	0.2	0.25

Table 3: Sources

Together with changing contexts and questions, this results in a highly dynamic environment. Keeping track of the trust values and ranking answers with time constraints results in a high workload.

6.2.2 User Interface

Figure 17 shows the user interface.

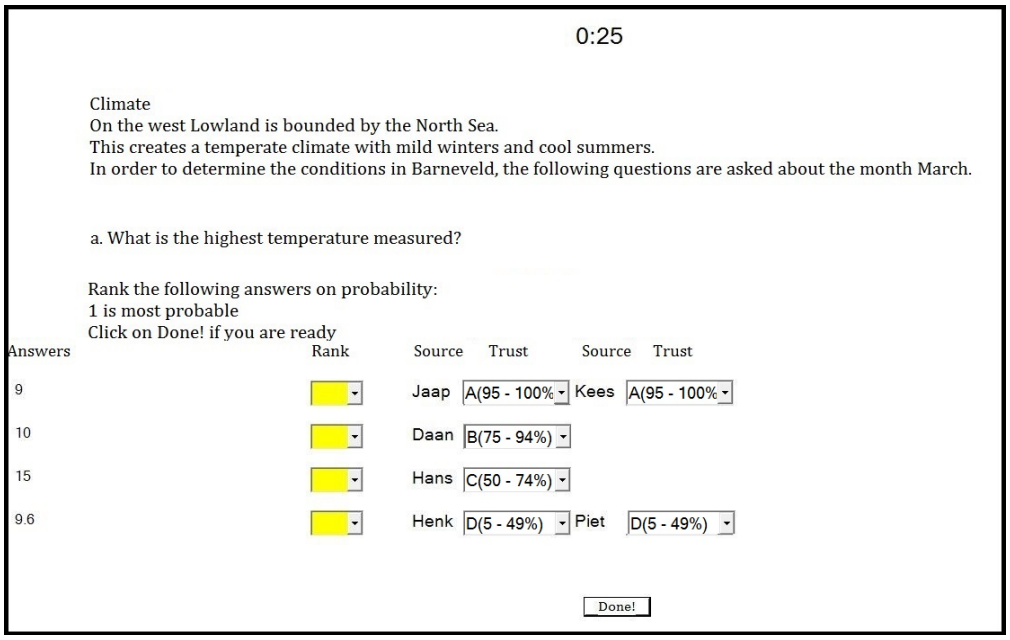


Figure 17: User Interface (translated)

The digits at the top display a timer indicating the time that is left to perform the task. Below the timer the context is shown. In the middle of the user interface the question asked to the sources is shown, followed by the task of the participant. Below the task on the left side the answers given are shown. These answers are given by the sources which are shown on the same horizontal line as the answer is presented, so in this case both Jaap and Kees gave the answer *9*.

Next to the answers, the rank of the answers is shown. Next to the sources the trust in the sources is presented. The trust given in the previous question is displayed in each of the comboboxes. The participants have to go through four steps per question, displayed in the user interface as yellow marking. The first step is shown in Figure 17. Participants have to fill in the ranking by clicking on the comboboxes and selecting the right number, in this case 1 to 4. Number 1 presents the best answer and the lowest number presents the worst answer. The participants get 30 seconds to fill in the ranking. If the participants are done with the step, they can click on the button *Done!* and go to the next step. If the time is up, the participant also goes to the next step. In the second step the right ranking of the answers is displayed next to the ranking comboboxes. The third step is to adjust (or fill in) their trust in each of the six sources with use of the evaluation categories from Army (2006) (see Table 4). They get 25 seconds to do this. In the fourth and last step, all objects except the timer are removed and a RSME scale if presented at which participants have to fill in their perceived cognitive workload. After the fourth step a new question is presented. If the question was the last in a certain set, the consequences of the ranking were shown, together with a new context. The participant could read the consequences and context and go on with the next question.

Category	Percentage	Name	Description
A	95 - 100%	Reliable	No doubt of authenticity, trustworthiness, or competency; has a history of complete reliability
B	75 - 94%	Usually Reliable	Minor doubt about authenticity, trustworthiness, or competency; has a history of valid information most of the time
C	50 - 74%	Fairly Reliable	Doubt of authenticity, trustworthiness, or competency but has provided valid information in the past
D	5 - 49%	Not Usually Reliable	Significant doubt about authenticity, trustworthiness, or competency but has provided valid information in the past
E	0 - 4%	Unreliable	Lacking in authenticity, trustworthiness, and competency; history of invalid information
F	0 - 100%	Cannot Be Judged	No basis exists for evaluating the reliability of the source

Table 4: Evaluation of Source Reliability from Army (2006), percentages added

6.3 Design

This experiment employs a 4 (condition) x 3 (scenario) within-subjects design. The four conditions are: H, T1, T2 and S. In the H condition the participant has to perform the task alone and in the S condition the stage Two implementation of the T-QAS has to perform the task alone. The S condition was done offline (without the participants). H and S are baseline conditions. In the T1 condition the stage One implementation of the T-QAS provided advice about the trust in each of the sources. In the T2 condition the stage Two implementation of the T-QAS provided advice about the trust in each of the sources and advice about the ranking.

We created three similar scenarios to provide the participants a similar scenario in the H, T1 and T2 conditions. The online conditions (H, T1 and T2) and scenarios were balanced using a latin square resulting in nine (3 x 3) combinations.

6.4 Independent Variables

This experiment has three independent variables: support type, scenario and amount of inconsistent information. The two support types are the stage One implementation and the stage Two implementation of the T-QAS. The first support type provides advice about the trust in the sources as presented in Figure 18.

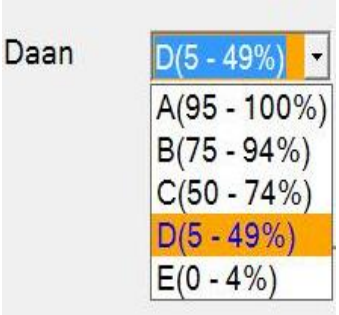


Figure 18: Advice about the trust in the sources

The second support type provides advice about the trust in the sources as presented in Figure 18 and provides advice about the ranking as presented in Figure 19.

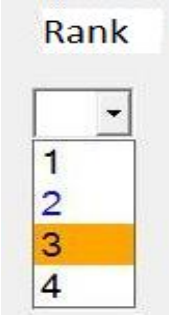


Figure 19: Advice about the rank of the answers (translated)

The second independent variable is the scenario, as explained in the previous section. The three scenarios represent the (fictional) situation in the cities *Barneveld*, *Scherpenzeel* and *Leusden*. The questions in each of the scenarios are equal, but the answers and sources differ. In each of the scenarios six sources presented answers to the questions. All scenarios were perceived equally difficult and a pilot study has indicated that the independent variables are not influenced by the scenario.

The third independent variable is the amount of inconsistent information. Two, three, four or five answers are presented, each occurring 10 times in each condition.

6.5 Dependent Variables

This experiment has five dependent variables: task performance, trust estimation performance, trust estimation compliance, cognitive workload and preferences. Each of these dependent variables is explained in the subsections below.

6.5.1 Task Performance

Task performance is measured as

$$P_t = \frac{\sum_{q=j}^{n} (\frac{\sum_{a=1}^{m} (1 - \frac{d}{md})}{m})}{n - j} \tag{12}$$

where q is number of question, a is answer, d is distance, md is maximal distance.

The chance levels in each of the amount of inconsistent information is different, resulting in a performance in which one amount could influence the outcome of the performance measure more than other amounts. Task performance is, therefore, measured for each amount of inconsistent information separately. In section 6.8 is explained how the task performances of each amount are taken together to create one value.

The distance measure in the formula ("De Boer Distance") is chosen above other distance measures, because in this way the size of the error is measured in the best way (see Table 5). Whereas edit distance and hamming distance do not make a difference between swapping the best and the worst answer compared to the best and the second best answer, this distance measure does.

Method	"1 - 2 - 3 - 4"	"4 - 2 - 3 - 1"	"2 - 1 - 3 - 4"
De Boer Distance	0	6	2
Edit Distance	0	1	1
Hamming Distance	0	2	2

Table 5: Distance Measurements

6.5.2 Trust Estimation Performance

Trust estimation performance is measured as

$$P_{te} = \frac{\sum_{q=j}^{n}(\frac{\sum_{s=1}^{k}(\frac{r}{ts})}{k})}{n-j} \tag{13}$$

where q is number of question, s is the source, r is the right trust (1) or not (0), ts is the total amount of sources.

With this formula only correct classifications increase performance. Difference measurement as with task performance do not work in this case, because if someone would fill in the middle category (C) at all times performance would be 0.833%. With this formula performance is only .33%.

6.5.3 Trust Estimation Compliance

Trust estimation compliance is measured as:

$$C_{te} = \frac{\sum_{q=j}^{n}(\frac{\sum_{s=1}^{k}(\frac{st}{mst})}{k})}{n-j} \tag{14}$$

where q is number of question, s is the source, st is similar trust as system (1) or not (0), mst is maximal amount of trust values similar.

With this formula for every source the trust given by the participant and the trust of the system are compared to each other. Only the same trust categories are counted and divided by the maximal amount of trust values.

6.5.4 Cognitive Workload

The cognitive workload is measured with the RSME scale (see figure 20) after each question. The RSME scale is a subjective scale, so only the perceived cognitive workload can be measured.

Figure 20: RSME

6.5.5 Personality and Preferences

The personality factors and preferences are gathered with the use of questionnaires (see Appendix A).

6.6 Apparatus

For this experiment four rooms were used: three experiment rooms and a room for the experiment leader. This experiment can also be done with one experiment rooms and a room for the experiment leader. Each of the rooms was equipped with a computer, a computer display (17"), a (computer) mouse, a chair, a table and a camera. The computers in the experiment rooms were able to run an executable file. The computers in the experiment room contained three scenarios and an executable file. The experiment leader room had access to the cameras in the experiment rooms. The experiment room had a pen, questionnaires, an acceptance paper and instructions.

6.7 Procedure and Instructions

An overview of the procedure is given in Table 6. After the experiment leader has welcomed the participants with a short introduction, participants fill in an acceptance form and a general questionnaire and read the instructions to the experiment (see Appendix B). Before each condition one question is practiced in presence of the experiment leader in order to determine if the participant understands the condition. If the participant fully understands the condition, the condition starts, otherwise the question is practiced again. Each condition consists of two phases: a learning phase with 10 practice questions and an experimental phase with 40 test questions. After each condition participants have to fill in a questionnaire and the participants have a five-minute break.

Activity	**Duration in minutes**
Introduction: Explanation Experiment Permission participant Coffee Personal Data Instructions	15
Practice Session + questions	2
Condition 1	45
Questionnaire + Break	2 + 5
Practice Session + questions	2
Condition 2 Questionnaire + Break	45 2 + 5
Practice Session + questions	2
Condition 3 Questionnaire + Ending	45 10
Total	**180**

Table 6: Procedure

6.8 Analyses

The data is analyzed with IBM SPSS Statistics (Statistical Product and Service Solutions). For both performance measures z-scores are used. As mentioned in the section 6.5, task performance is measured separately for each amount of answers. The z-scores are also separately calculated and put together to create one value for task performance. A Repeated Measures ANOVA is used to compare means of task performance, trust estimation performance and workload. Linear regression is done to see if trust estimation performance and trust estimation compliance have a positive linear relation. Fourth, multiple regression analysis is done to test if personality factors predict task performance, trust estimation performance, task compliance or trust estimation compliance.

7 Results

In this chapter the results of the experiment are reported based on the hypotheses described in chapter 5. The first section shows the results for task performance in order to test hypotheses 1 and 2 and 3 and the results regarding good and poor performers in order to test hypothesis 4. The second section shows the results regarding trust estimation performance in order to test hypotheses 5, 6 and 7 and the results regarding trust estimation compliance in order to test hypothesis 8. The results for cognitive workload are shown in the third section and the last section shows the results for the personality and preferences.

7.1 Task Performance

Repeated Measures ANOVA showed a statistically significant effect on task performance ($F(3, 105) = 7.252$; $p < .0005$). Post hoc tests using Bonferroni correction revealed a significant difference ($p < .0005$) between task performance in the T1 condition ($.287 \pm .592$) and task performance in the S condition ($-.200 \pm .000$) and a significant difference ($p < .0005$) between task performance in the T2 condition ($.172 \pm .477$) and task performance in the S condition ($-.200 \pm .000$). No other significant differences were revealed (see Figure 21 for the graph). If we compare these results to hypothesis 1, which states that task performance in condition T1 and T2 is higher than task performance in H, we can see that hypothesis 1 is rejected. Hypothesis 2, however, is accepted, because task performance in both T1 and T2 is significantly higher than in S. Hypothesis 3 is rejected, because no difference between T1 and T2 is found.

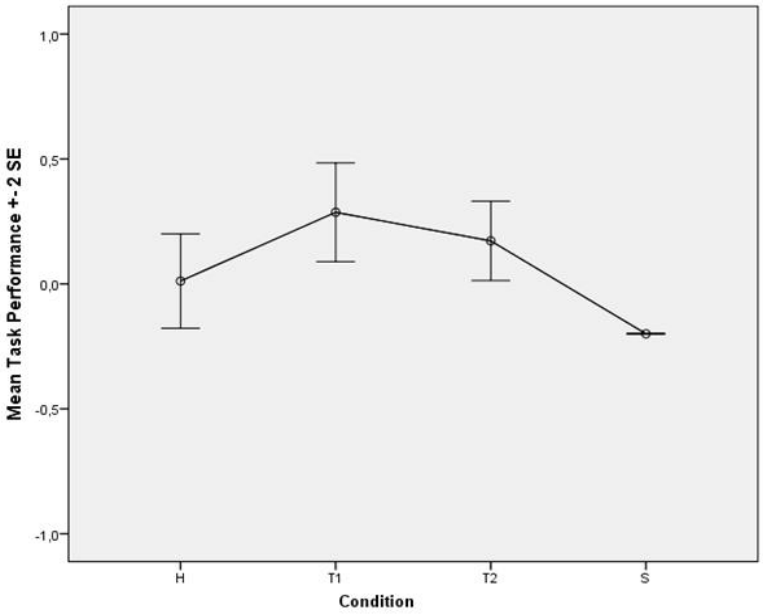


Figure 21: Task Performance

7.1.1 Good vs. Poor Performers

Repeated Measures ANOVA showed a statistically significant effect on task performance of poor performers (n = 20; F(2.042, 38.790) = 7.95, p = .002) and a significant effect on task performance of good performers (n = 16; F(3, 45) = 7.312, p < .0005). The significant differences revealed by post hoc tests using Bonferroni correction are shown in Table 7 and Table 8. The graphs are shown in Figure 22 and Figure 23. The results indicate that poor performers can perform significantly better with support from T-QAS compared to without support. On the other hand, good performers perform not significantly worse with support from T-QAS compared to performing alone, even if T-QAS performs significantly worse than the good performers. Hypothesis 4, which states that the difference in task performance between H and both T1 and T2 is higher for poor task performers than for good task performers, can be accepted. Poor task performers perform significantly better in both T1 and T2 compared to H and good task performers do not perform significantly better in T compared to H. If we look back at hypothesis 1, we can see this hypothesis is accepted for poor performers.

Difference	**Condition 1**	**Condition 2**
p = .021	T1 (.231 ± .635)	H (-.363 ± .323)
p = .007	T2 (.106 ± .454)	H (-.363 ± .323)
p = .041	T1 (.231 ± .635)	S (-.200 ± .000)
p = .043	T2 (.106 ± .454)	S (-.200 ± .000)

Table 7: Significant Differences in Task Performance of Poor Performers

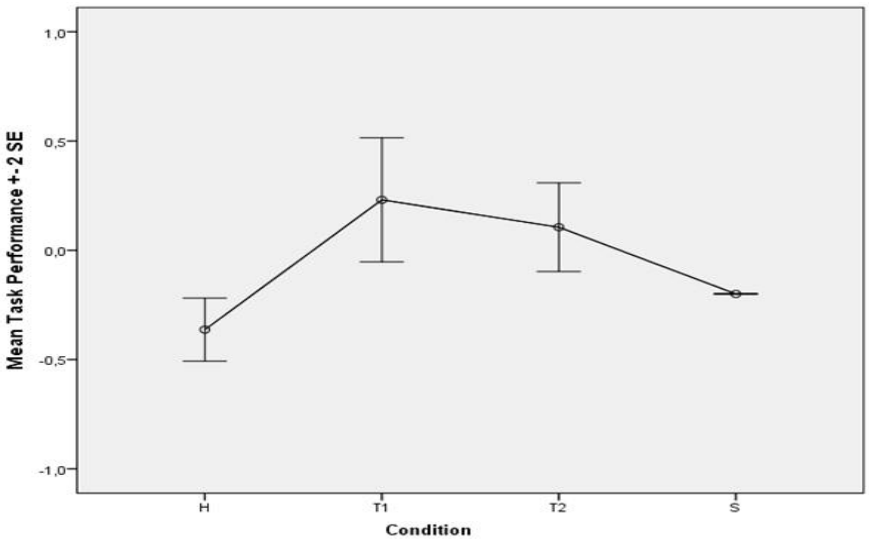


Figure 22: Task Performance of Poor Performers

Difference	Condition 1	Condition 2
p < .0005	H (.479 ± .447)	S (-.200± .000)
p = .007	T1 (.356 ± .546)	S (-.200 ± .000)
p = .041	T2 (.256 ± .507)	S (-.200 ± .000)

Table 8: Significant Differences in Task Performance of Good Performers

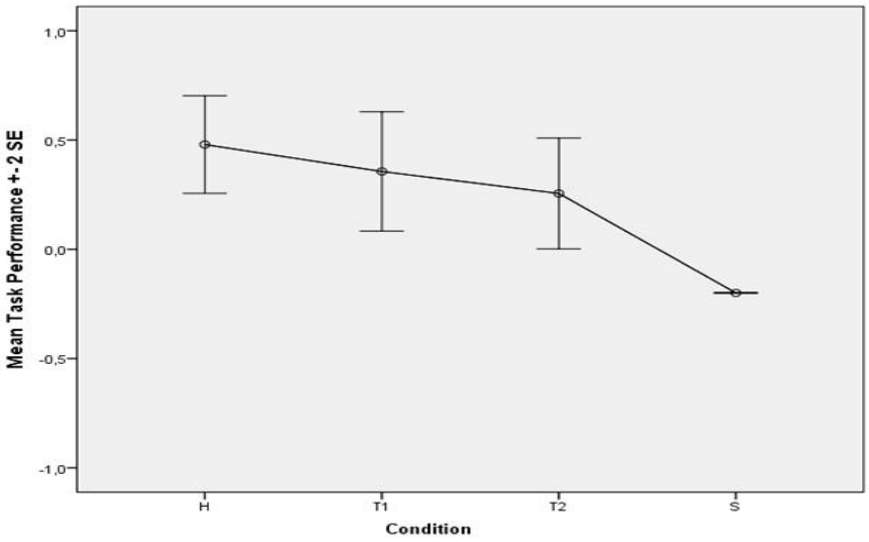


Figure 23: Task Performance of Good Performers

7.2 Trust Estimation Performance

Repeated Measures ANOVA showed a statistically significant effect on trust estimation performance ($F(3,105) = 14.080$; $p < .0005$). The significant differences revealed by post hoc tests using Bonferroni correction are shown in Table 9. Comparing the results to hypothesis 5, which states that performance in T1 and T2 is higher than in H, we can see that this hypothesis is not accepted. Performance in T1 and T2 is higher than in H, but the difference between T2 and H is not significant ($p = .318$). Hypothesis 6 is accepted, because performance in both T1 and T2 is lower than in S. Hypothesis 7 is also accepted, because no difference between T1 and T2 is found.

Difference	Condition 1	Condition 2
p = .017	H (-.253 ± 1.081)	T1 (.220 ± .999)
p < .0005	H (-.253 ± 1.081)	S (.693 ± .000)
p = .045	T1 (.220 ± .999)	S (.693 ± .000)
p < .0005	T2 (.033 ± .880)	S (.693 ± .000)

Table 9: Significant Differences in Trust Estimation Performance

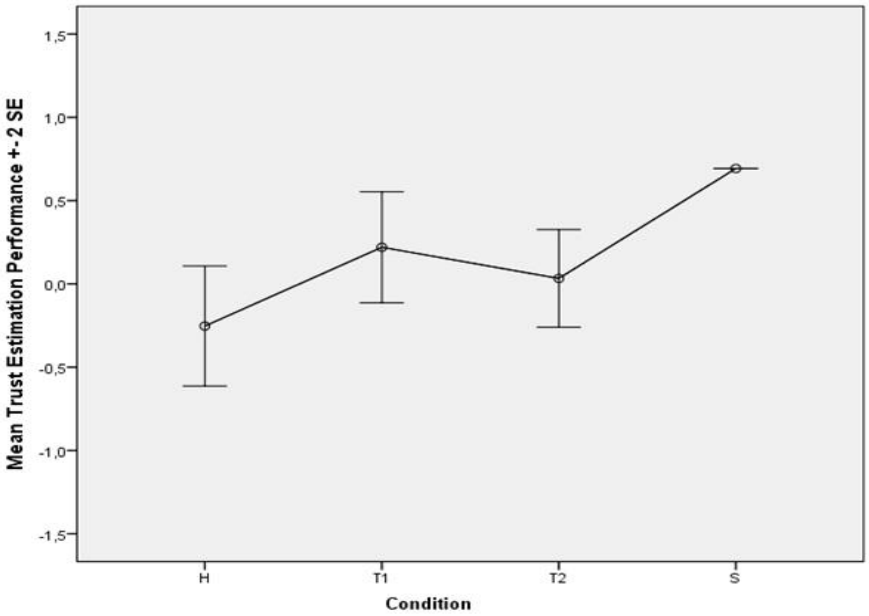


Figure 24: Trust Estimation Performance

7.2.1 Trust Estimation Compliance

In order to test hypothesis 7 (trust estimation performance and trust estimation compliance have a positive linear relation), linear regression analysis is performed. Trust estimation compliance (β = .877, p < .05) in condition T1 explains 76.2% of the variance of the trust estimation performance in the same condition (see Figure 25 for the graph) and in condition T2 76,1% of the variance of the trust estimation performance is explained by trust estimation compliance (β = .876, p < .05). The graph is shown in Figure 26. Hypothesis 7 is thus accepted.

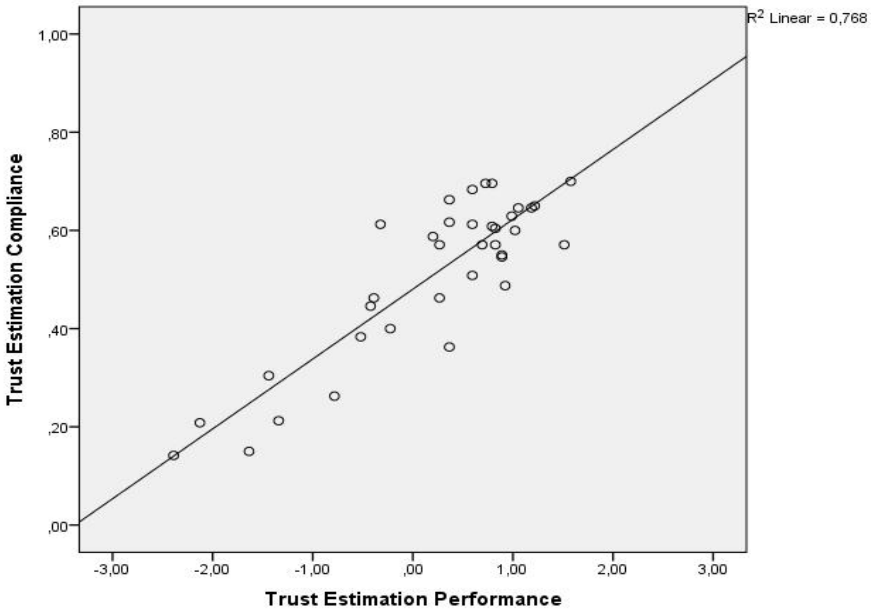


Figure 25: Relation between TEP and TEC in condition T1

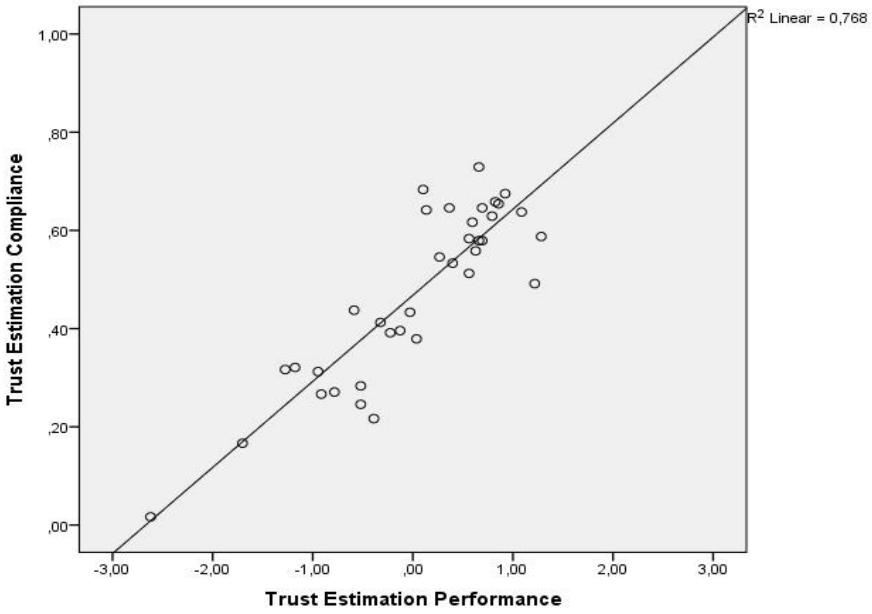


Figure 26: Relation between TEP and TEC in condition T2

7.3 Cognitive Workload

Repeated Measures ANOVA showed a statistically significant effect on cognitive workload ($F(3,105) = 84.750$; $p < .0005$). Post hoc tests using Bonferroni correction only revealed significant differences between the S condition, in which the cognitive workload is zero, and the other conditions, but no difference between the conditions (see Figure 27). Hypothesis 9 and hypothesis 10 are rejected, because cognitive workload is not reduced and cognitive workload is not lower in T2 compared to T1.

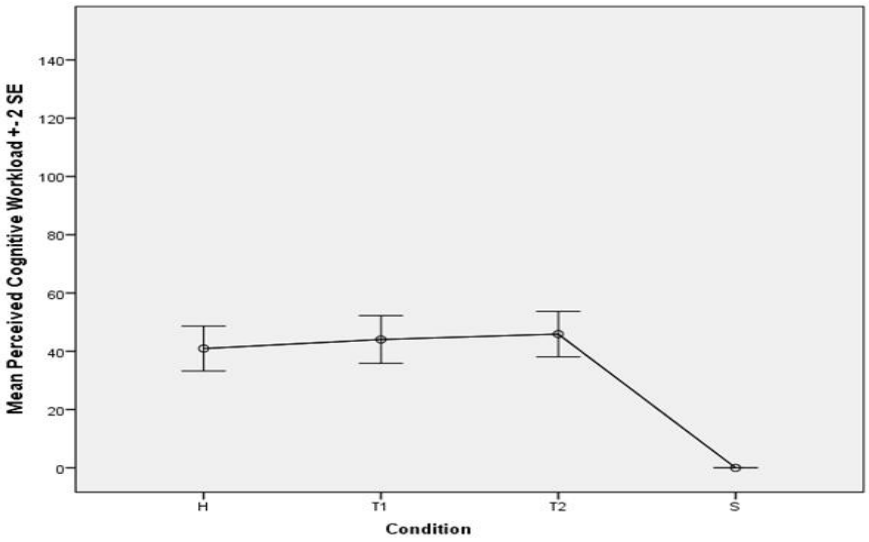


Figure 27: Perceived Cognitive Workload

7.4 Personality and Preferences

Multiple regression analysis was used to test if personality factors (age, gender, education level, hours of computer use, hours of quiz watching, neurotism, extraversion, openness, altruism, conscientiousness and trust) significantly predicted the task performance, task compliance, trust estimation performance or trust estimation compliance in any of the conditions. The perceived cognitive workload ($\beta = .554$, $p < .05$) , the trust estimation performance ($\beta = .367$, $p = .01$) and the amount of sleep ($\beta = -.351$, $p = .013$) explained 39,0% of the variance of the task performance. Age ($\beta = -.481$, $p < .05$) , gender ($\beta = -.527$, $p < .05$) and hours of quiz watching ($\beta = -.292$, $p < .05$) explained 32,5% of the variance of trust estimation performance in the H condition. Age ($\beta = -.523$, $p < .05$) and gender ($\beta = -.351$, $p < .05$) also significantly predicted the variance (24,7%) of the trust estimation performance in the T1 condition. A Pearson correlation for the data revealed that trust estimation performance in the H condition and altruism are significantly related ($r = -.365$, $p = 0{,}029$).

Participants indicated that advice about the trust (4.11 (of 7) $\pm$ 1.53) and advice about the ranking (4.19 (of 7) $\pm$ 1.51) is useful.

8 Discussion

In this chapter the results described in the previous chapter are discussed. In the light of the results, the support model is discussed. This chapter ends with further research.

8.1 Task Performance

If we compare the hypothesized task performance graph (Figure 14) with the task performance graph from the experiment (Figure 21), we see almost the same pattern. This is a good finding, because those pictures represent the hypothesis that a human-system team outperforms both the system and the human. Hypothesis 1 is, however, not fully accepted because the human-system team outperforms only poor performers. A reason that good performers have no better performance in T1 and T2 could be that these performers cannot estimate when to use the systems advice. No significant differences in task performance were found for good performers, which indicates that the addition of T-QAS, even if it performs significantly worse than the human, does not reduce performance. Hypothesis 2 is accepted, so an addition of a human to the system significantly increases performance. No differences between T1 and T2 are found, which resulted in a rejection of the third hypothesis. An interesting observation is that task performance in T1 is slightly higher (not significant) than the task performance in T2. In T1 participants are only indirectly supported in the task, whereas in T2 the indirect support is converted to direct support. A problem with this direct support could be that participants get out of the loop by fully complying to the system. With indirect support participants still have to reason about the task. This could be the reason why the performance in T1 is slightly higher than the performance in T2. If we look at hypothesis 4, we can see that this hypothesis is accepted. As mentioned before, poor performers can benefit from the T-QAS is more cases than good performers. Good performers have to estimate when to use the systems advice.

An implication for the support model is that supporting part of the process can improve performance as good as supporting the whole process. This could, however, be due to the systems task performance. Good performers performed significantly better than the system, which resulted in less benefit of the system for good performers. In order to let good performers benefit from the system as well, the system has to perform better. One option to accomplish a better system performance is to add a ranking by semantic similarity measure to the formula of the estimated value. Similar answers such as successive numbers or hyponyms, synonyms or hypernyms could increase the estimated value of both. Another option is to use the method from Newcomb and Hammell II (2012) to estimate the ranking. A third option is to give T-QAS (limited) natural language understanding and world knowledge. In this way T-QAS can not only reason with values, but also with natural language.

8.2 Trust Estimation Performance

If the hypothesized trust estimation performance graph (Figure 15) and the trust estimation performance graph from the experiment (Figure 24) are compared, we see a quite similar pattern.

T-QAS is good in objectively estimating trust values, as expected, and humans are not. This is confirmed by the acceptance of hypothesis 8, because trust estimation compliance in both the T1 and T2 condition has a positive linear relation with trust estimation performance. This positive linear relation is probably also the reason for the rejection of hypothesis 5. Performance in T1 and T2 is not significantly higher than in H, because of a lack of compliance. With a compliance of 100% performance would be significantly higher in T1 and T2 compared to H (because of the significant difference between H and S). Both hypothesis 6 and 7 are accepted. Interesting is that trust estimation performance is slightly higher in T1 compared to T2, which was also the case with task performance. A possible explanation is that the two types of advice were confusing for the participant. T-QAS is good in estimating trust values and should be complied to get a better performance. T-QAS is not very good in the task and should only be complied in cases in which T-QAS is better. Participants get feedback about the task, so participants have an indication of the task performance of T-QAS. Participants get no feedback about the calculation, but they might have an idea about the calculation performance of T-QAS. If these estimations are put together, this results in an under-compliance of the advice about trust values and an over-compliance of the advice about the task, which is exactly what we see in the results.

The results concerning trust estimation performance imply that trust estimation could be automated. T-QAS performs better than the human and both T1 and T2 were not able to reach performance of T-QAS. The multiple trust models, one for every source, create a good trust estimation performance, so this approach does not have to be adjusted. Adding more meta-information as proposed by Jenkins and Bisantz (2011) could, however, further improve performance.

8.3 Cognitive Workload

The reported perceived cognitive workload did not meet the expectation of hypotheses 9 and 10. Two explanations of these differences between the reported and expected workload are session order and the amount of times the RSME had to be filled in. First, a significant difference between sessions is found. The perceived cognitive workload is highest in the first session and lowest in the last session. Between-subject analysis showed no significant differences, so this might not be the best explanation. Another explanation could be that participants had to fill in the RSME after each question. The perceived workload values of a participant converted to a certain average after the first couple of questions, so this could cause the lack of differences. If we, however, look at the amount of answers, significant perceived cognitive workload differences between the amounts are found. On the other hand, participants indicated in a questionnaire to perceive the lowest workload in condition T2.

The automation of the trust estimation could be a first step to decrease cognitive workload. Another option is providing a training with the system, because the perceived cognitive workload was lowest in the last condition.

8.4 Further Research

As stated in the previous sections, T-QAS and results of the experiment are very promising. T-QAS can help intelligence analysts with the task of decision making with information from possibly unreliable sources and with the task of estimating the reliability of the sources. In the future, a first step would be to implement the whole system and test the system with professional intelligence analysts. It would be nice to compare the results with other proposed solutions such as solutions from G. M. Powell and Broome (2002), Jenkins and Bisantz (2011), Patterson et al. (2001) and Newcomb and Hammell II (2012). Furthermore, a lot of research can be done in each of the processes. In the process *Question Delivery* performance of systems from McGrath et al. (2000) and Van Diggelen et al. (2012) can be compared. For the implementation of those systems into the support model more research should be done on the factors and criteria that are used to decide to who, when and in what format the question is delivered. In this support model, the first step is to detect when an agent is 'appropriate' and what tags should be used in the smart question. For the appropriateness of the agent not only agent information but also information from the QA database could be used. Similar questions could identify appropriate agents. Furthermore, it is important to investigate the implications of changing the collection process in the military domain.

In the process *Trust Management* it is important to do more research on meta-information. The trust models proposed in this models work well, but a good value for λ and good function for the calculation of the experience are required. It would, furthermore, be interesting to see what the effect is of a distinction of trust in different domains. It is possible that a source is very reliable in one domain (his domain of expertise) and less reliable in another domain.

In the process *Selection of Answer(s)* a lot of further research can be done. It would be interesting to see what the effect is of tiling hypernyms, hyponyms and synonyms instead of only perfect matches, what the effect is of adding these relations to create a better estimation of the value of the answer or adding information about previous questions such as the ranked answers and agent that gave that answer. In the process of *Answer Ranking* the method from Newcomb and Hammell II (2012) can be compared to the method used in this thesis. More research can be done to see what the optimal formula is, what the influence is of asking for ratings both on the system and on the source and what the (theoretical and practical) optimal performance of T-QAS is. For the response formulation the best way of formulating and presenting the answers should be examined. The response formulation could, for example, be linked to the proposed knowledge generation methods of Biermann et al. (2004) and G. M. Powell and Broome (2002). If the response is directly sent to the question agent, research on the user interface and usability has to be done. For the usability, a first step would be to meet the standards for question answering systems as proposed by Burger et al. (2001) (see section 2.2.4).

T-QAS could not only be used in the military domain, but also in other organizations in which information has to be gathered and analyzed. It would be interesting to see the effects of T-QAS in these organizations.

9 Conclusions

In this thesis a support model for intelligent analysts is proposed. This support model uses a trust-based question answering system (T-QAS). This model cannot only use documents as source, as question answering systems do, but can also be used for information from other sources such as humans. The differencing part of this support model is the use of trust models which keep track of trust in each of the sources.

T-QAS is evaluated in an experiment in which participants had to perform a decision making task with unreliable sources in order to answer the research question. The research question can be answered positively, because T-QAS helps participants to execute their task. The results show that task performance of a human supported by T-QAS is higher than task performance of a poor performer alone and task performance of the system (T-QAS) alone. Even if the system performs significantly worse than the human, the human will not perform significantly worse. T-QAS is thus suited for the job of helping humans to execute a decision-making task, because support from T-QAS has no negative effect on performance. In the task of estimating trust in agents the human-system team performs better than the human alone. T-QAS can thus contribute to the task of estimating trust in agents. The system performs even better than the human-system team. This indicates that the estimation of trust can be automated in the future. Workload is, however, not lowered.

Concluding, we can state that organizations where information has to be analyzed, including defense organizations, could support their (intelligence) analysts with a support model such as T-QAS.

References

Abney, S., Collins, M., & Singhal, A. (2000). Answer extraction. In *Proceedings of the sixth conference on Applied natural language processing* (pp. 296–301).

Aliod, D. M., Berri, J., & Hess, M. (1998). A real world implementation of answer extraction. In *Database and Expert Systems Applications, 1998. Proceedings. Ninth International Workshop on* (pp. 143–148).

Army, U. (2006). *Field manual 2-22.3: Human intelligence collector operations.* Washington DC: Headquarters: Department of the Army.

Azari, D., Horvitz, E., Dumais, S., & Brill, E. (2002). Web-based question answering: A decision-making perspective. In *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence* (pp. 11–19).

Badalamente, R. V., & Greitzer, F. L. (2005). Top ten needs for intelligence analysis tool development. In *proceedings of the 2005 international conference on intelligence analysis.*

Baeza-Yates, R., Ribeiro-Neto, B., et al. (1999). *Modern information retrieval* (Vol. 463). ACM press New York.

Bahrami, A., Yuan, J., Smart, P. R., & Shadbolt, N. R. (2007). Context aware information retrieval for enhanced situation awareness. In *Military Communications Conference, 2007. MILCOM 2007. IEEE* (pp. 1–6).

Biermann, J. (2006). Remarks on resource management in intelligence. In *Information Fusion, 2006 9th International Conference on* (pp. 1–3).

Biermann, J., Chantal, L., Korsnes, R., Rohmer, J., & Ündeger, C. (2004). *From unstructured to structured information in military intelligence-some steps to improve information fusion* (Tech. Rep.). DTIC Document.

Boury-Brisset, A.-C., Frini, A., & Lebrun, R. (2011, June). *All-source Information Management and Integration for Improved Collective Intelligence Production* (Tech. Rep.). DEFENCE R&D CANADA.

Breck, E., Burger, J., Ferro, L., Hirschman, L., House, D., Light, M., & Mani, I. (2000). How to evaluate your question answering system every day ... and still get real work done. In *Proceedings 2nd International Conference on Language Resources and Evaluation (LREC-2000)* (p. 1495-1500).

Buchholz, S., & Daelemans, W. (2001). Complex answers: a case study using a www question answering system. *Natural language engineering*, *7*(4), 301–323.

Burger, J., Cardie, C., Chaudhri, V., Gaizauskas, R., Harabagiu, S., Israel, D., ... others (2001). Issues, tasks and program structures to roadmap research in question & answering (Q&A). In *Document Understanding Conferences Roadmapping Documents.*

Chen, Q., & Deng, Q.-n. (2009). Cloud computing and its key techniques. *Journal of Computer Applications*, *29*(9), 2565.

Coles, L. S., & Center, A. I. (1972). *Techniques for information retrieval using an inferential question-answering system with natural-language input.* Artificial Intelligence Center.

Cook, M. B., & Smallman, H. S. (2008). Human factors of the confirmation bias in intel-

ligence analysis: Decision support from graphical evidence landscapes. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *50*(5), 745–754.

Dumais, S., Banko, M., Brill, E., Lin, J., & Ng, A. (2002). Web question answering: Is more always better? In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 291–298).

Eachus, P., & Short, B. (2011). Decision support system for Intelligence Analysts. In *Intelligence and Security Informatics Conference (EISIC), 2011 European* (pp. 291–291).

Eppler, M. J., & Mengis, J. (2003). A framework for information overload research in organizations. *Insights from organization science, accounting, marketing, MIS, and related disciplines. paper*, *1*, 2003.

Gonsalves, P., & Cunningham, R. (2001). Automated ISR Collection Management System. In *Proc. Fusion 2001 Conference.*

Green, B., Wolf, A., Chomsky, C., & Laughery, K. (1961). Baseball: An automatic question answerer. *Proceedings Western Computing Conference*, *19*, 219-224.

Harabagiu, S., Moldovan, D., Pasca, M., Mihalcea, R., Surdeanu, M., Bunescu, R., ... Morarescu, P. (2000). Falcon: Boosting knowledge for answer engines. In *Proceedings of TREC* (Vol. 9).

Hirschman, L., & Gaizauskas, R. (2001). Natural language question answering: The view from here. *Natural Language Engineering*, *7(4)*, 275-300.

Hovy, E., Gerber, L., Hermjakob, U., Junk, M., & Lin, C.-Y. (2000). Question answering in webclopedia. In *Proceedings of the TREC-9 Conference.*

Hsu, J. (2011, September). *Military faces info overload from robot swarms.* Retrieved from `http://www.nbcnews.com/id/44430826/ns/technology_and_science-innovation/t/military-faces-info-overload-robot-swarms/`

Hutchins, S., Pirolli, P., & Card, S. (2004). *A new perspective on use of the critical decision method with intelligence analysts* (Tech. Rep.). DTIC Document.

Intelligence cycle. (n.d.). Retrieved from `http://www.fbi.gov/about-us/intelligence/intelligence-cycle`

The intelligence cycle. (2007, April). Retrieved from `https://www.cia.gov/kids-page/6-12th-grade/who-we-are-what-we-do/the-intelligence-cycle.html`

Jenkins, M. P., & Bisantz, A. M. (2011). Identification of human-interaction touch points for intelligence analysis information fusion systems. In *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on* (pp. 1–8).

Jijkoun, V., & De Rijke, M. (2004). Answer selection in a multi-stream open domain question answering system. *Advances in Information Retrieval*, 99–111.

Katz, B. (1997). From sentence processing to information access on the world wide web. In *AAAI Spring Symposium on Natural Language Processing for the World Wide Web* (Vol. 1, p. 997).

Kerbusch, P., Paulissen, R., van Trijp, S., Gaddur, F., & van Diggelen, J. (2012). *Mutual Empowerment 2012 Smart Questions CD&E* (Tech. Rep.). TNO.

Kim, S., Oh, J., & Oh, S. (2007). Best-answer selection criteria in a social Q&A site from the

user-oriented relevance perspective. *Proceedings of the American Society for Information Science and Technology*, *44*(1), 1–15.

Kleinberg, J. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, *46*(5), 604–632.

Kupiec, J. (1993). MURAX: A robust linguistic approach for question answering using an on-line encyclopedia. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and development in information retrieval* (p. 181-190).

Li, P., Wang, X., Guan, Y., & Xu, Y. (2006). Answer extraction based on system similarity model and stratified sampling logistic regression in rare date. *IJCSNS*, *6*(3), 1.

Lin, J., Quan, D., Sinha, V., Bakshi, K., Huynh, D., Katz, B., & Karger, D. (2003). What makes a good answer? The role of context in question answering. In *Proceedings of the Ninth IFIP TC13 International Conference on Human-Computer Interaction (INTERACT 2003)* (pp. 25–32).

Liu, D., Chen, Y., Kao, W., & Wang, H. (2012). Integrating expert profile, reputation and link analysis for expert finding in question-answering websites. *Information Processing & Management*.

Lopez, V., Nikolov, A., Fernandez, M., Sabou, M., Uren, V., & Motta, E. (2009). Merging and ranking answers in the semantic web: The wisdom of crowds. *The semantic web*, 135–152.

Lusher, R., & Stone III, G. (1997). A prototype report filter to reduce information overload in combat simulations. In *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on* (Vol. 3, pp. 2788–2793).

Matthews, W. (2012, January). *Data surge and automated analysis: The latest isr challenge.* Retrieved from `http://www.emc.com/collateral/analyst-reports/the-latest-isr-challenge-insights-gbc.pdf` (A Briefing from GBC: Industry Insights)

McGlaun, S. (2009, July). *Report: Information overload is a major issue for military.* Retrieved from `http://www.dailytech.com/Report+Information+Overload+is+a+Major+Issue+Military/article15653.htm`

McGrath, S., Chacón, D., & Whitebread, K. (2000). Intelligent mobile agents in the military domain. In *Fourth International Conference on Autonomous Agents.*

McSkimin, J., & Minker, J. (1977). *The use of a semantic network in a deductive question-answering system.* Citeseer.

Milward, D., & Thomas, J. (2000). From information retrieval to information extraction. In *Proceedings of the ACL-2000 workshop on Recent advances in natural language processing and information retrieval: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 11* (pp. 85–97).

Moldovan, D., Harabagiu, S., Pasca, M., Mihalcea, R., Goodrum, R., Girju, R., & Rus, V. (1999). Lasso: A tool for surfing the answer net. In *Proceedings of the eighth text retrieval conference (TREC-8).*

Moll, D., & Vicedo, J. (2007). Question answering in restricted domains: An overview. *Computational Linguistics*, *33(1)*, 41-61.

Nagao, M., & Tsujii, J.-i. (1973). Mechanism of Deduction in a Question Answering System

with Natural LanguageInput. In *International Joint Council on Artificial Intelligence. International Joint Conference, 3rd, Stanford, CA* (pp. 20–23).

Newcomb, E. A., & Hammell II, R. J. (2012). Examining the Effects of the Value of Information on Intelligence Analyst Performance. In *Proceedings of the Conference on Information Systems Applied Research ISSN* (Vol. 2167, p. 1508).

Nrl flight-tests autonomous multi-target, multi-user tracking capability (Vol. Spring). (2012).

Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). The PageRank citation ranking: bringing order to the web.

Pasça, A., & Adviser-Harabagiu, S. (2001). *High-performance, open-domain question answering from large text collections.* Southern Methodist University.

Patterson, E., Woods, D., Tinapple, D., Roth, E., Finley, J., & Kuperman, G. (2001). Aiding the intelligence analyst in situations of data overload: From problem definition to design concept exploration. *Institute for Ergonomics/Cognitive Systems Engineering Laboratory Report, ERGO-CSEL.*

Pfautz, J., Roth, E., Bisantz, A., Thomas-Meyers, G., Llinas, J., & Fouse, A. (2006). *The role of meta-information in C2 decision-support systems* (Tech. Rep.). DTIC Document.

Powell, G. M., & Broome, B. (2002). *Fusion-based knowledge for the objective force* (Tech. Rep.). DTIC Document.

Powell, M. (2012, December). *Intelligence analyst.* Retrieved from `http://www.prospects.ac.uk/intelligence_analyst_job_description.htm`

Powers, R. (n.d.). *35f – intelligence analyst.* Retrieved from `http://usmilitary.about.com/od/enlistedjobs/a/96b.htm` (Derived from Army Pamplet 611-21)

Prager, J., Brown, E., Radev, D. R., & Czuba, K. (2001). One search engine or two for question-answering. *Ann Arbor*, *1001*, 48109.

Radev, D., Fan, W., Qi, H., Wu, H., & Grewal, A. (2005). Probabilistic question answering on the web. *Journal of the American Society for Information Science and Technology*, *56(6)*, 571-583.

Ramprasath, M., & Hariharan, S. (2012). A survey on Question Answering System. *International Journal of Research and Reviews in Information Sciences (IJRRIS)*, *2*(1), 171-179.

Roth, E. M., Pfautz, J. D., Mahoney, S. M., Powell, G. M., Carlson, E. C., Guarino, S. L., ... Potter, S. S. (2010). Framing and Contextualizing Information Requests: Problem Formulation as Part of the Intelligence Analysis Process. *Journal of Cognitive Engineering and Decision Making*, *4*(3), 210–239.

Saggion, H., Gaizauskas, R., Hepple, M., Roberts, I., & Greenwood, M. (2004). Exploring the performance of boolean retrieval strategies for open domain question answering. In *Proc. of the IR4QA Workshop at SIGIR.*

Scott, S., & Gaizauskas, R. (2000). University of sheffield trec-9 q & a system. In *In Proceedings of the 9th Text REtrieval Conference.*

Shanker, T., & Richtel, M. (2011, January 16). *In new military, data overload can be deadly.* Retrieved from `http://www.nytimes.com/2011/01/17/technology/17brain.html?pagewanted=all&_r=0` (In the New York Times)

Simmons, R. (1965). Answering english questions by computer: A survey. *Communications of the ACM*, *8(1)*, 53-70.

Staff, T. I. O. S. (1996, May). *Intelligence threat handbook.* Retrieved from `http://www.fas.org/irp/nsa/ioss/threat96/index.html` (Operations Security)

Suzuki, J., Sasaki, Y., & Maeda, E. (2002). SVM answer selection for open-domain question answering. In *Proceedings of the 19th international conference on Computational Linguistics-Volume 1* (pp. 1–7).

Van Diggelen, J., Grootjen, M., Ubink, E., van Zomeren, M., & Smets, N. (2012). *Content-based design and implementation of ambient intelligence applications.*

Van Maanen, P.-P. (2010). *Adaptive support for human-computer teams: Exploring the use of cognitive models of trust and attention.* Unpublished doctoral dissertation, Vrije Universiteit Amsterdam. (p. 72 - 99)

Van Diggelen, J., van Drimmelen, K., Heuvelink, A., Kerbusch, P. J., Neerincx, M. A., van Trijp, S., ... van der Vecht, B. (2012). Mutual empowerment in mobile soldier support. *Journal of Battlefield Technology*, *15*(11).

Vector 21 - a strategic plan for the Defense Intelligence Agency. (n.d.). Retrieved from `http://www.fas.org/irp/dia/vector21/`

Voorhees, E. M. (1999). The TREC question answering track report. In *Proceedings of TREC-8* (p. 77-82). Gathersburg, MD.

What we do. (n.d.). Retrieved from `http://www.nrojr.gov/teamrecon/res_nro-whatwedo.html`

Wright, E., Mahoney, S., Laskey, K., Takikawa, M., & Levitt, T. (2002). Multi-entity Bayesian networks for situation assessment. In *Information Fusion, 2002. Proceedings of the Fifth International Conference on* (Vol. 2, pp. 804–811).

Zadeh, L. A. (2006). From search engines to question answering systems - The problems of world knowledge, relevance, deduction and precisiation. *Capturing Intelligence*, *1*, 163 - 210.

Zajac, R. (2001). Towards ontological question answering. In *Proceedings of the workshop on Open-domain question answering-Volume 12* (pp. 1–7).

A Questionnaires (Dutch)

Vragenlijst M

Geef aan in welke mate u het eens bent met deze stellingen **over de afgelopen conditie**.

	Vragen over eigen prestatie	← Oneens			Neutraal			Eens→
1.	Ik vond de taak moeilijk om uit te voeren.	O	O	O	O	O	O	O
2.	Ik vond het moeilijk om geconcentreerd te blijven tijdens de taak.	O	O	O	O	O	O	O
3.	Ik heb het gevoel dat mijn inschatting van het vertrouwen in de bronnen altijd goed was.	O	O	O	O	O	O	O
4.	Ik twijfelde altijd aan mijn ordening van de antwoorden.	O	O	O	O	O	O	O
5.	Ik heb altijd goede ordeningen van de antwoorden kunnen geven.	O	O	O	O	O	O	O

Vragenlijst S

Geef aan in welke mate u het eens bent met deze stellingen **over de afgelopen conditie**.

	Vragen over eigen prestatie	← Oneens			Neutraal			Eens→
1.	Ik vond de taak moeilijk om uit te voeren.	O	O	O	O	O	O	O
2.	Ik vond het moeilijk om geconcentreerd te blijven tijdens de taak.	O	O	O	O	O	O	O
3.	Ik heb het gevoel dat mijn inschatting van het vertrouwen in de bronnen altijd goed was.	O	O	O	O	O	O	O
4.	Ik twijfelde altijd aan mijn ordening van de antwoorden.	O	O	O	O	O	O	O
5.	Ik heb altijd goede ordeningen van de antwoorden kunnen geven.	O	O	O	O	O	O	O

	Vragen over het advies over het vertrouwen in de bronnen	← Oneens			Neutraal			Eens→
6.	Ik heb het advies over het vertrouwen in de bronnen altijd opgevolgd.	O	O	O	O	O	O	O
7.	Het advies over het vertrouwen in de bronnen kwam altijd overeen met mijn inschatting van het vertrouwen in de bronnen.	O	O	O	O	O	O	O
8.	Zonder het advies over het vertrouwen in de bronnen zal ik slechter zijn in het bepalen van het vertrouwen in de bronnen.	O	O	O	O	O	O	O
9.	Zonder het advies over het vertrouwen in de bronnen zal ik slechter zijn in het ordenen van de antwoorden.	O	O	O	O	O	O	O
10.	De manier waarop het systeem tot zijn advies over het vertrouwen in de bronnen kwam was voor mij duidelijk.	O	O	O	O	O	O	O

Vragenlijst A

Geef aan in welke mate u het eens bent met deze stellingen **over de afgelopen conditie**.

	Vragen over eigen prestatie	← Oneens			Neutraal			Eens→
1.	Ik vond de taak moeilijk om uit te voeren.	O	O	O	O	O	O	O
2.	Ik vond het moeilijk om geconcentreerd te blijven tijdens de taak.	O	O	O	O	O	O	O
3.	Ik heb het gevoel dat mijn inschatting van het vertrouwen in de bronnen altijd goed was.	O	O	O	O	O	O	O
4.	Ik twijfelde altijd aan mijn ordening van de antwoorden.	O	O	O	O	O	O	O
5.	Ik heb altijd goede ordeningen van de antwoorden kunnen geven.	O	O	O	O	O	O	O

	Vragen over het advies over het vertrouwen in de bronnen	← Oneens			Neutraal			Eens→
6.	Ik heb het advies over het vertrouwen in de bronnen altijd opgevolgd.	O	O	O	O	O	O	O
7.	Het advies over het vertrouwen in de bronnen kwam altijd overeen met mijn inschatting van het vertrouwen in de bronnen.	O	O	O	O	O	O	O
8.	Zonder het advies over het vertrouwen in de bronnen zal ik slechter zijn in het ordenen van de antwoorden.	O	O	O	O	O	O	O
9.	Zonder het advies over het vertrouwen in de bronnen zal ik slechter zijn in het bepalen van het vertrouwen in de bronnen.	O	O	O	O	O	O	O
10.	De manier waarop het systeem over het vertrouwen in de bronnen tot zijn advies kwam was voor mij duidelijk.	O	O	O	O	O	O	O

	Vragen over het advies over de ordening van de antwoorden	← Oneens			Neutraal			Eens→
11.	Ik heb het advies over de ordening van de antwoorden altijd opgevolgd.	O	O	O	O	O	O	O
12.	Het advies over de ordening van de antwoorden kwam altijd overeen met mijn inschatting van het ordening van de antwoorden.	O	O	O	O	O	O	O
13.	Zonder het advies de ordening van de antwoorden zal ik slechter zijn in het bepalen van de ordening van de antwoorden.	O	O	O	O	O	O	O
14.	De manier waarop het systeem tot zijn advies over de ordening van de antwoorden kwam was voor mij duidelijk.	O	O	O	O	O	O	O

Vragenlijst

Vul deze vragen in aan het eind van het experiment
De vragen gaan over het gehele experiment.

De taak was

	← Oninteressant			Neutraal			Interessant→
1.	O	O	O	O	O	O	O

	← Moeilijk			Neutraal			Makkelijk→
2.	O	O	O	O	O	O	O

Vragen advies

		← Oneens			Neutraal			Eens→
3.	Ik vertrouwde erop dat het systeem goed advies gaf over het vertrouwen in de bronnen.	O	O	O	O	O	O	O
4.	Ik vond het advies over het vertrouwen in de bronnen nuttig.	O	O	O	O	O	O	O
5.	Ik vertrouwde erop dat het systeem goed advies gaf over de ordening van de antwoorden.	O	O	O	O	O	O	O
6.	Ik vond het advies over de ordening van de antwoorden nuttig.	O	O	O	O	O	O	O

U heeft nu drie condities meegemaakt:

M (geen advies)
S (advies over het vertrouwen)
A (advies over het vertrouwen en de ordening)

In welke van deze drie condities was uw prestatie het hoogst, denkt u? Kunt u dit verklaren?

In welke van de drie condities was de taak het makkelijkst? Wat is hiervan de reden?

Z.O.Z.

In welke van de drie condities was uw cognitieve werklast het laagst? Waarom?

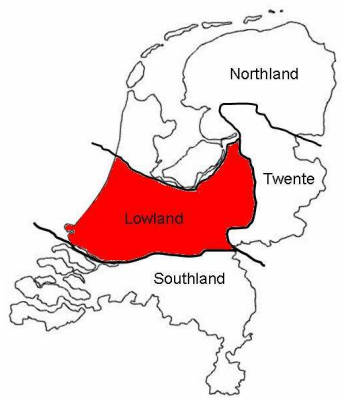
Maakte u gebruik van een bepaalde tactiek bij het ordenen van de antwoorden? Zo ja, welke?

Heeft u nog opmerkingen op de taak of in het algemeen op het experiment?

B Instructions (Dutch)

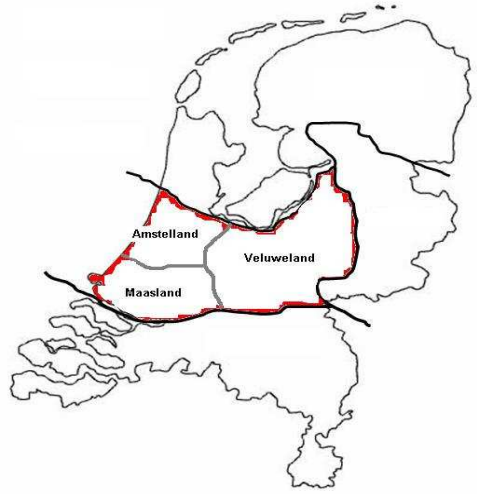
LOWLAND

Lowland is een fictief klein en zelfstandig democratisch land in het noordwesten van Europa. Het land wordt in het noorden begrensd door het land *Northland*, in het oosten door de staat *Twente* en in het zuiden door *Southland*. In het westen wordt het land begrensd door de Noordzee (zie figuur 1).



Figuur 1. Lowland met omringende landen

Lowland is ingedeeld in drie provincies: Amstelland, Maasland en Veluweland (zie figuur 2).



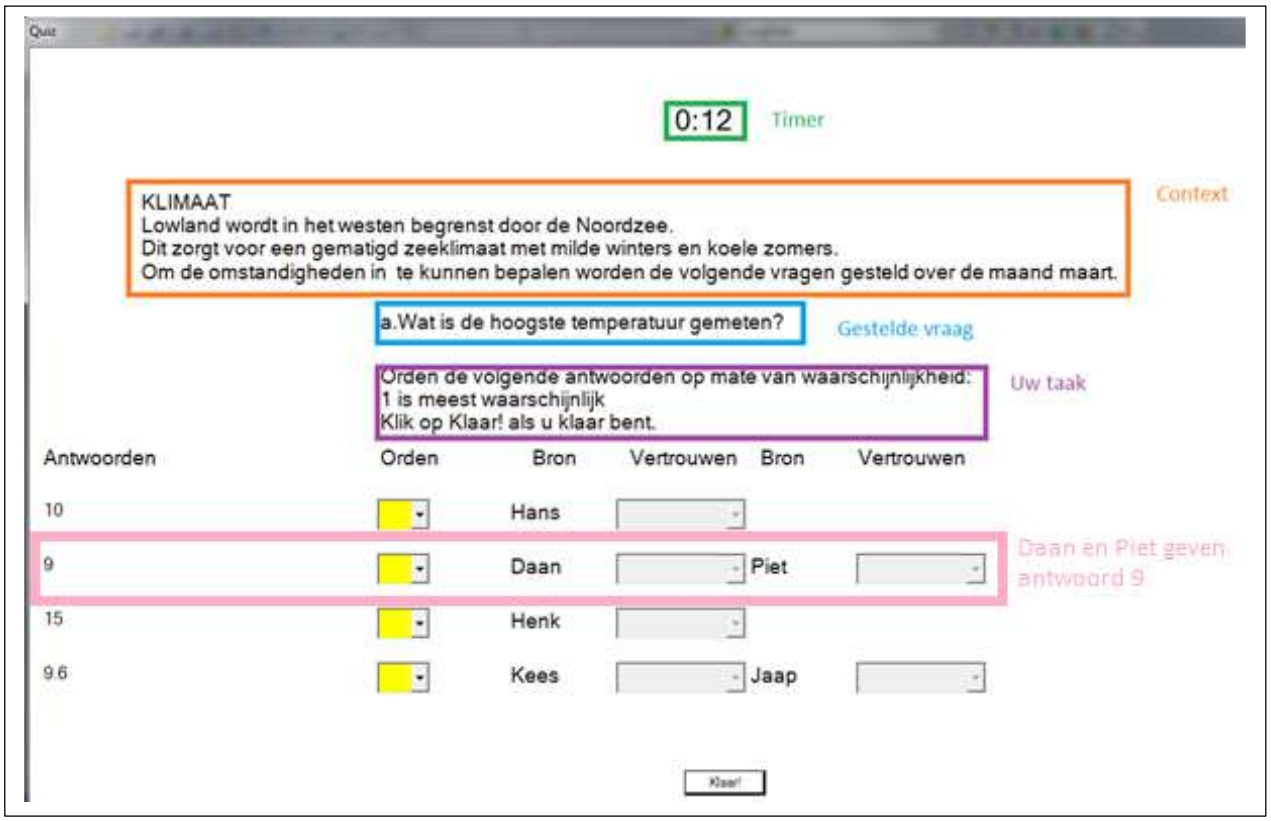
Figuur 2. Provincies van Lowland

In de periode van november 2012 tot en met februari 2013 hebben zich een groot aantal incidenten voorgedaan in de staat. Een groepering met de naam Streng Gereformeerde Eenheids Partij (SGEP) tracht met geweld een eigen territorium af te bakenen in de provincie Veluweland. De regering van Lowland weet niet op effectieve wijze het hoofd te bieden tegen het gewelddadige optreden en heeft hulp ingeroepen van de internationale gemeenschap (IG). De IG heeft ingestemd met ingrijpen en heeft diverse landen gevraagd om een bijdrage te leveren. Diverse landen hebben reeds een bijdrage toegezegd. De Nederlandse besluitvorming is gaande.

Voor de besluitvorming van een eventuele deelname van Nederlandse eenheden is informatie over het gebied nodig. De benodigde informatie wordt omgezet in een vraag die aan militairen wordt meegegeven tijdens een militaire operatie in dat gebied.
De antwoorden op de vragen worden verzameld en naar een informatieanalist gestuurd. Het is de bedoeling dat de informatieanalist met behulp van deze antwoorden een beeld van het gebied

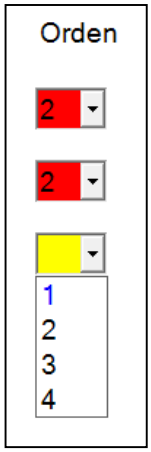
opbouwt om zo te helpen met de besluitvorming. In dit experiment neemt u een deel van de taken van de informatieanalist op zich. Van drie verschillende plaatsen uit de provincie Veluweland zal een beeld worden opgebouwd. In elk van de plaatsen zijn verschillende bronnen geraadpleegd om antwoorden op vragen uit elf verschillende categorieën te krijgen. Deze bronnen geven echter soms verschillende antwoorden. Aan u de taak om de antwoorden te ordenen naar mate van juistheid. Om deze ordening zo goed mogelijk te kunnen maken, zal voor elke vraag de context van de vraag gepresenteerd worden. Deze context kan ook (impliciet) informatie geven over het beste antwoord.

In figuur 3 kunt u zien hoe de taakomgeving is opgezet.



Figuur 3. Taakomgeving

Bovenaan wordt een timer gepresenteerd. U dient voordat de tijd is afgelopen uw taak te hebben afgerond en op de knop **Klaar!** onderaan te klikken. U wordt niet beoordeeld op hoe snel u klaar bent met de taak (binnen de tijd die ervoor staat), maar op hoe goed u de taak uitvoert. U krijgt per vraag vier verschillende schermen te zien. In de eerste drie schermen wordt zowel door middel van gele vlakken als door middel van de tekst (uw taak) aangegeven wat u moet doen. In het eerste scherm is het de bedoeling dat u de antwoorden ordent op waarschijnlijkheid dat het antwoord juist is door middel van het invullen van de boxen onder het woord **ORDEN**. U krijgt hiervoor **30 seconden** de tijd. In figuur 4 ziet u hoe een uitgeklapte box eruit ziet. U dient in elk vak een uniek getal in te vullen dat uw ordening representeert, waarbij het getal 1 staat voor het beste antwoord. Mocht u toch twee keer hetzelfde getal invullen, worden beide vlakken rood gekleurd (zie figuur 4). **Als u niet op tijd klaar bent of u twee dezelfde getallen heeft ingevuld, wordt de volledige ordening fout gerekend**.



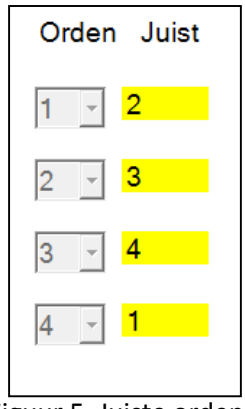
Figuur 4. Ordenen

Voor deze ordening krijgt u informatie over de context, de gestelde vraag, de antwoorden op de vraag en welke bron of bronnen het antwoord hebben gegeven (zie figuur 3). Achter de naam van de bron is te zien in welke vertrouwensschaal u de bron na de vorige vraag heeft geplaatst. Hoe deze vertrouwensschaal is opgebouwd kunt u zien in tabel 1. **LET OP**: de percentages zijn niet uniform verdeeld.

Rang	Percentages	Omschrijving
A	95 – 100%	Volledig te vertrouwen
B	75 – 94%	Meestal te vertrouwen
C	50 – 74%	Redelijk te vertrouwen
D	5 – 49%	Meestal niet te vertrouwen
E	0 – 4%	Niet te vertrouwen
F	-	Onbekend

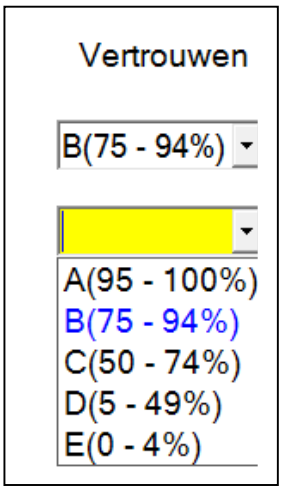
Tabel 1. Vertrouwensschaal

Als de tijd is verlopen of u op **Klaar!** heeft geklikt, volgt het tweede scherm.
In dit scherm wordt de juiste ordening van de antwoorden naast de door u ingevulde ordening weergegeven (zie figuur 5). U krijgt **drie seconden** om deze ordening te bestuderen.



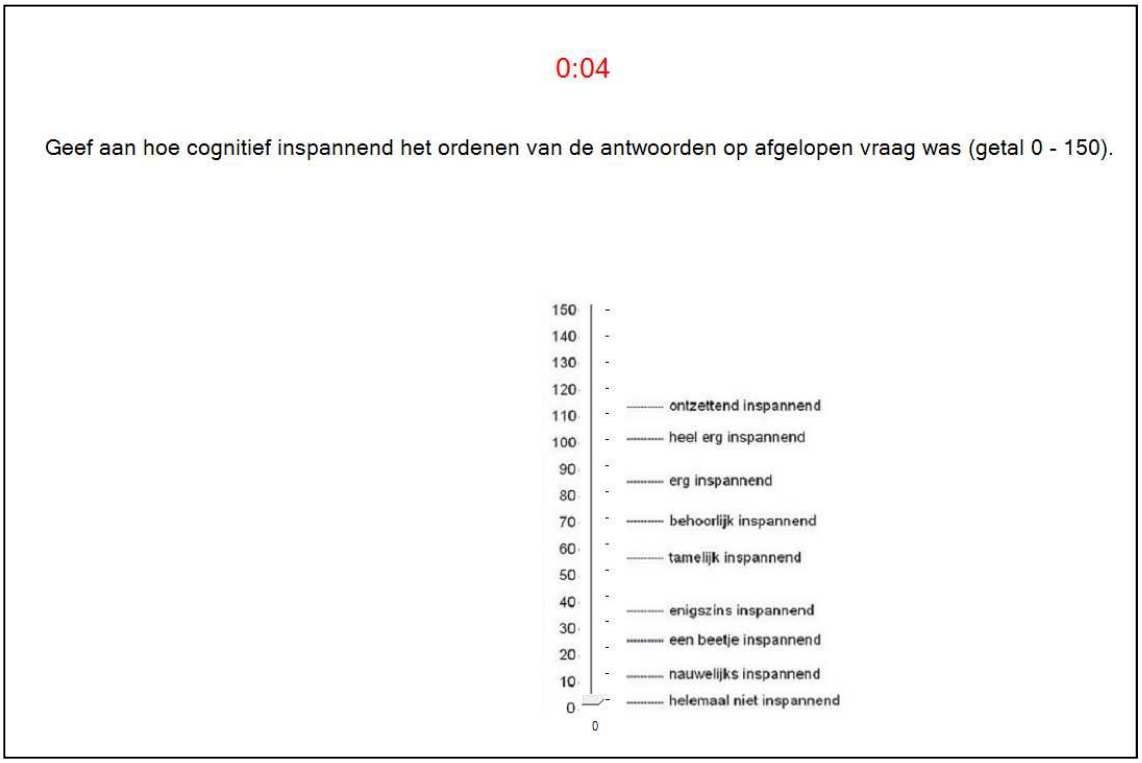
Figuur 5. Juiste ordening

In het derde scherm krijgt u de kans uw vertrouwen in elk bron aan te passen door middel van de box onder het woord **VERTROUWEN** (zie figuur 6). Hiervoor krijgt u **25 seconden** de tijd. Als u eerder klaar bent, kunt u op de knop **Klaar!** onderaan klikken.



Figuur 6. Vertrouwen

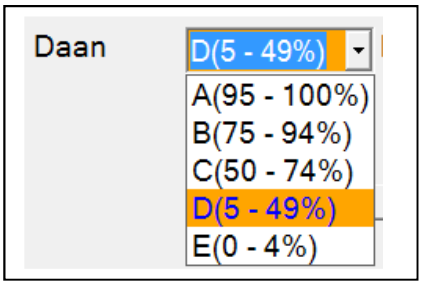
In het vierde scherm dient u aan te geven hoe cognitief inspannend het ordenen van de antwoorden was door middel van het verschuiven van de slider (zie figuur 7). **Het is de bedoeling dat u dit serieus invult en blijft invullen**. U krijgt hiervoor slechts **5 seconden** de tijd.



Figuur 7. Inspanning

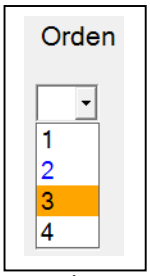
Zoals eerder is vermeld zijn er elf categorieën van vragen. Aan het einde van elke categorie zal u een conclusie en een context voor de nieuwe categorie te lezen krijgen. Voor het lezen van de conclusie krijgt u **10 seconden** de tijd en voor het lezen van de nieuwe context krijgt u **30 seconden** de tijd.

Het experiment bestaat uit drie condities. In alle condities zal u eerst één vraag oefenen met de proefleider. Vervolgens zal u tien oefenvragen (2 categorieën) krijgen, gevolgd door 40 testvragen (9 categorieën). In twee van de condities wordt het vertrouwen berekend door een systeem dat gebruik maakt van een lerend algoritme. Dit wordt weergegeven door middel van het oranje kleuren van het geadviseerde vertrouwen (zie figuur 8).



Figuur 8. Geadviseerd vertrouwen

In één van de twee condities met een systeem, zal het systeem ook advies geven over de ordening van de antwoorden. Dit wordt ook weergegeven door middel van het oranje kleuren van het geadviseerde cijfer (zie figuur 9).



Figuur 9. Geadviseerd cijfer

U kunt zelf beslissen wat u met de geadviseerde informatie doet. U wordt voor het starten van een scenario geïnformeerd over de conditie. Na elk scenario zal u worden gevraagd een vragenlijst in te vullen. Vragen stellen is alleen toegestaan voor de start van een scenario en na het maken van de oefenvragen. U dient het gehele scenario verder zelf te doorlopen.