FEATURE-TAK - Framework for Extraction, Analysis, and Transformation of Unstructured Textual Aircraft Knowledge

Pascal Reuss[1][2], Rotem Stram[1], Cedric Juckenack[1], Klaus-Dieter Althoff[1][2], Wolfram Henkel[3], Daniel Fischer[3], and Frieder Henning[4]

[1] German Research Center for Artificial Intelligence
Kaiserslautern, Germany
http://www.dfki.de
[2] Institute of Computer Science, Intelligent Information Systems Lab
University of Hildesheim, Hildesheim, Germany
http://www.uni-hildesheim.de
[3] Airbus Operations GmbH
Kreetslag 10, 21129 Hamburg, Germany
[4] Lufthansa Industry Solutions, Hamburg, Germany

Abstract. This paper describes a framework for semi-automatic knowledge extraction for case-based diagnosis in the aircraft domain. The available data on historical problems and their solutions contain structured and unstructured data. To transform these data into knowledge for CBR systems, methods and algorithms from natural language processing and case-based reasoning are required. Our framework integrates different algorithms and methods to transform the available data into knowledge for vocabulary, similarity measures, and cases. We describe the idea of the framework as well as the different tasks for knowledge analysis, extraction, and transformation. In addition, we give an overview of the current implementation, our evaluation in the application context, and future work.

1 Introduction

The amount of experience knowledge is huge in many companies. They store historical data about projects, incidents, occurred problems and their solutions, and mauch other information. All this can be used to gather experience to solve future problems. Because the aircraft domain is a technical domain, much information is clearly structured like attribute-value pairs, taxonomies, and ontologies. But there also exists information in form of free text written by cabin crew members, pilots, or maintenance technicians. Examples of free text are the cabin and pilot logbook, customer service reports, and maintenance reports. To use this information in the context of a case-based reasoning (CBR) system, they have to be analyzed and transformed into useful knowledge for vocabulary, similarity measures, and cases. While the structured information can be transformed with little to moderate effort, sometimes it can even be used without transformation, the unstructured information in free texts can only be analyzed and transformed with high effort from a knowledge engineer. To support a knowledge engineer at this task, we developed a framework that combines several methods from natural language processing (NLP) and CBR to automate the transformation process. The framework is called FEATURE-TAK, a **F**ramework for **E**xtraction, **A**nalysis, and **T**ransformation of **U**nstructu**RE**d **T**extual **A**ircraft **K**nowledge. While parts of the framework are not new to the research community, other parts were developed or improved in-house to bridge the gap to an automated knowledge transformation directly usable in CBR systems. In addition, the combination of the NLP and CBR tasks as well as the underlying methodology and the benefit for knowledge transformation for CBR systems is a new approach and will help to reduce the creation and maintenance effort of CBR systems. In the following section we describe the project context in which the use case of the framework occurred and several basics. Section 3 contains related work about the topic, while Section 4 gives an overview of the framework and detailed information about the individual NLP and CBR tasks. We also describe our current implementation status and evaluation in Section 4.4. At the end we summarize our paper and give an outlook on future work.

2 OMAHA project

The OMAHA project is supported by the Federal Ministry of Economy and Technology in the context of the fifth civilian aeronautics research program [9]. The high-level goal of the OMAHA project is to develop an integrated overall architecture for health management of civilian aircraft. The project covers several topics like diagnosis and prognosis of flight control systems, innovative maintenance concepts and effective methods of data processing and transmission. A special challenge of the OMAHA project is to outreach the aircraft and its subsystems and integrating systems and processes in the ground segment like manufacturers, maintenance facilities, and service partners. Several enterprises and academic and industrial research institutes take part in the OMAHA project: the aircraft manufacturer Airbus (Airbus Operations, Airbus Defense & Space, Airbus Group Innovations), the system and equipment manufacturers Diehl Aerospace and Nord-Micro, the aviation software solutions provider Linova and IT service provider Lufthansa Systems as well as the German Research Center for Artificial Intelligence and the German Center for Aviation and Space. In addition, several universities are included as subcontractors. The OMAHA project has several different sub-projects. Our work focuses on a subproject to develop a cross-system integrated system health monitoring (ISHM). The main goal is to improve the existing diagnostic approach with a multi-agent system (MAS) with several case-based agents to integrate experience into the diagnostic process and provide more precise diagnoses and maintenance suggestions. In this context we have to acquire cases from historical data, which contains a high number of free texts. Therefore, the development of an approach to analyze and transform this free text is required.

2.1 SEASALT

The SEASALT (Shared Experience using an Agent-based System Architecture Layout) architecture is a domain-independent architecture for extracting, analyzing, sharing, and providing experiences [6]. The architecture is based on the Collaborative Multi-Expert-System approach [3][4] and combines several software engineering and artificial intelligence technologies to identify relevant information, process the experience and provide them via an user interface. The knowledge modularization allows the compilation of comprehensive solutions and offers the ability of reusing partial case information in form of snippets. Figure 1 gives an overview over the SEASALT architecture.

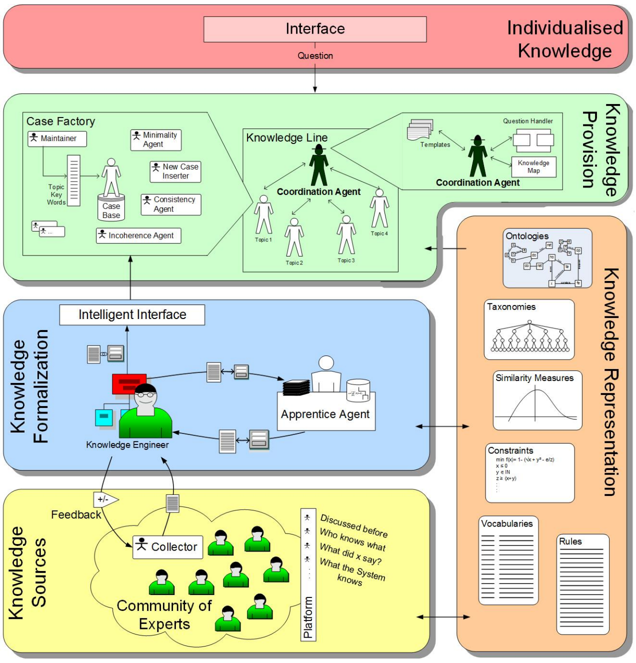

Fig. 1. Overview of the SEASALT architecture

The SEASALT architecture consists of five components: the *knowledge sources*, the *knowledge formalization*, the *knowledge provision*, the *knowledge representation*, and the *individualized knowledge*. The *knowledge sources* component is responsible for extracting knowledge from external knowledge sources like databases or web pages and especially Web 2.0 platforms, like forums and social media platforms. These knowledge sources are analyzed by so-called Collector Agents, which are assigned to specific Topic Agents. The Collector Agents collect all contributions that are relevant for the respective Topic Agent's topic. The *knowledge formalization* component is responsible for formalizing the extracted knowledge from the Collector Agents into a modular, structural representation. This formalization is done by a knowledge engineer with the help of a so-called Apprentice Agent. This agent is trained by the knowledge engineer and can reduce the workload for the knowledge engineer. The *knowledge provision* component contains the so called Knowledge Line. The basic idea is a modularization of knowledge analogous to the modularization of software in product lines. The modularization is done among the individual topics that are represented within the knowledge domain. In this component a Coordination Agent is responsible for dividing a given query into several sub queries and pass them to the according Topic Agent. The agent combines the individual solutions to an overall solution, which is presented to the user. The Topic Agents can be any kind of information system or service. If a Topic Agent has a CBR system as a knowledge source, the SEASALT architecture provides a so-called Case Factory for the individual case maintenance. Several Case Factories are supervised by a so-called Case Factory Organization to coordinate the maintenance of the overall multi-agent system. The *knowledge representation* component contains the underlying knowledge models of the different agents and knowledge sources. The synchronization and matching of the individualized knowledge models improves the knowledge maintenance and the interoperability between the components. The *individualized knowledge* component contains the web-based user interfaces to enter a query and present the solution to the user [6][5][19].

2.2 Application domain

The aircraft domain is a highly complex technical domain. An aircraft consists of hundreds of components, which consists of dozens of systems, which contains dozens of individual parts, called Line Replacement Units (LRU). These systems and LRUs are interacting with and rely on each other. Therefore, it is not easy to identify the root cause of an occurred fault, because the root cause can either be found within a single LRU, or within the interaction of several components of a system, or even within the interaction of LRUs of different systems. Finding cross-system root causes is a very difficult and resource expensive task. The existing diagnosis system onboard an aircraft can track root causes based on causal rules defined for the LRUs. These rules are not always unambiguous, because the diagnosis approach is effect-driven. Based on a comprehensible effect (visible, audible, or smellable) in the cockpit or the cabin, the diagnosis system tries to determine the system behavior that belongs to the effect and traces the root cause through the defined rules. The use of CBR for the diagnosis can help to clear ambiguous diagnosis situations with the help of experience knowledge from successfully solved problems, especially with cross-system root causes.

3 Related Work

Many systems with textual knowledge use the textual CBR approach, like [24], [21], and [11]. The data sources available for our project are mainly structured data, therefore we choose a structural CBR approach. But the most important information about an occurred fault can be found in fault descriptions and logbook entries, which are free text. We decided to use a hybrid approach with the combination of structural CBR and NLP techniques to integrate all available information. There also exist several frameworks and toolkits for natural language processing like Stanford CoreNLP[17], Apache Lucene[18], GATE[12], and SProUT[13]. All these frameworks provide several algorithms and methods for NLP tasks, but do not link them directly to be used for CBR systems. Several methods from these frameworks are used by our framework, too. But we also combine them with techniques from association rule mining, case-based reasoning, and techniques developed in-house to have a direct use for knowledge modeling in CBR systems. There is extensive research pertaining to adjustment of feature weights in the past years and it is still an important topic. Wettschereck and Aha compared different feature weighting methods and developed five dimensions to describe these methods: Model, weight space, representation, generality and knowledge.[26] According to their work, our approach uses a wrapper model to optimize the feature weights iteratively during the training phases. The weight space is continuous, because the features of our problem vary in their relevance for different diagnoses. Our knowledge representation is a structural case structure with attribute-value pairs and this given structure is used for feature weighting. We are using case specific weights to set the weights for each diagnosis individually. This way we are able gain more precise results during the retrieval. Our approach for feature weighting is knowledge intensive, because we are using domain-specific knowledge to differentiate between individual diagnoses and setting case specific weights. An approach that addresses the same problem as our approach is presented by Sizov, Ozturk and Styrak[22]. They analyze free text documents from aircraft incidents to identify reasoning knowledge that can be used to generate cases from these text documents. While we are using a structural approach with attribute-value pairs and try to classify and map the identified relevant knowledge to attributes, the approach from Sizov and his colleagues uses a so-called text reasoning graph to represent their cases. The approach uses the same NLP techniques like the Standford CoreNLP Pipeline as our approach to analyze and preprocess the text documents. While we are using Association Rule Mining algorithms like Apriori and FP-Grwoth to identify associations between collocations and keywords, their approach uses pattern recognition to identify causal relations.

4 FEATURE-TAK

This section describes FEATURE-TAK, an agent-based **F**ramework for **E**xtraction, **A**nalysis, and **T**ransformation of **U**nstructe**RE**d **T**extual **A**ircraft **K**nowledge. We will describe the idea, the agent-based architecture and the individual tasks of the framework. We will support the description with a running example.

4.1 Problem description and framework idea

Airbus databases contain a lot of data about historical maintenance problems and their solutions. These data sets are currently used for maintenance support in various situations. To use this information within our case-based diagnosis system, they have to be analyzed to find relevant pieces of information to be transformed into knowledge for CBR systems. The data sets from Airbus contain different data structures. Technical information can mostly be found in attribute-value pairs, while logbook entries, maintenance , and feedback are stored in form of free text articles. Based on a first data analysis, we choose a structural approach for our CBR systems. Information like fault codes, aircraft type and model, ATA (Air Transport Association) chapter and fault emitter are important information and can easily be used within a structural approach. Over time the main use case changed within the OMAHA project and the new data to be transformed has free text components. Therefore, we have to use the structured information as well as the free text. To transform the relevant information in the free texts into useful knowledge for our structural CBR system, we had to adapt and combine techniques from NLP and CBR. The idea is to develop a framework to combine several techniques and automatize the knowledge transformation. This framework could be used for knowledge acquisition and maintenance for CBR system in the development phase or for existing CBR systems.

4.2 Framework architecture

The framework consists of five components: data layer, agent layer, CBR layer, NLP layer and interface layer. The data layer is responsible for storing the raw data and the processed data for each task. In addition, domain specific information like abbreviations and technical phrases are stored in this layer to be accessible for the other components. The agent layer contains several software agents. For every task an individual agent is responsible. All task agents communicate with a central supervising agent. This supervising agent coordinates the workflow. For visualization and communication purposes for the user, this layer also contains an interface agent. For each task an agent is spawned when starting the framework, but during the workflow additional agents can be spawned to support the initial agents with huge data sets. The NLP layer contains algorithms and methods like part of speech tagging, lemmatization, abbreviation replacement and association rule mining. These algorithms are used by the agents to execute their assigned tasks. The algorithms could either be third party libraries or own implementations. The fourth layer is the CBR layer and is responsible for the communication with a CBR tool like myCBR or jColibri. It contains methods to add keywords to the vocabulary, extend similarity measures and generate cases from the input data sets. The last layer contains the graphical user interface of the framework. This user interface can be used to configure the workflow, select input data, and start the workflow. In addition, the user interface presents the results of each task to the user and shows the status of the software agents.

4.3 Framework tasks

In this section we will describe the eight tasks of the framework in more detail. These tasks and their interaction are defined based on the existing input data and the required data structure for our CBR systems. Based on our initial idea and the experience from the input data analysis, we had to regroup the tasks and their substeps. As input for the workflow a data set with free text components, for example a CSV file, or a pure free text document is possible. In addition to the data sets, a file with mapping information, an abbreviations file, and files with domain specific white and black lists are used. The data sets are first transformed into an internal representation of our case structure based on the information in the mapping file. It is not required to have information for every attribute in the data set or to use all information in the data set. The complete case structure for our use case consists of 72 attributes with different data types and value ranges and the mapping process adapts dynamically to the input information. The complete workflow with all tasks and possible parallelization is show in Figure 2. In the following, the individual tasks will be described. As an example to illustrate the tasks, a free text problem description will be used:

– 'One hour before departure, cabin crew informed maint that the FAP was frozen.'

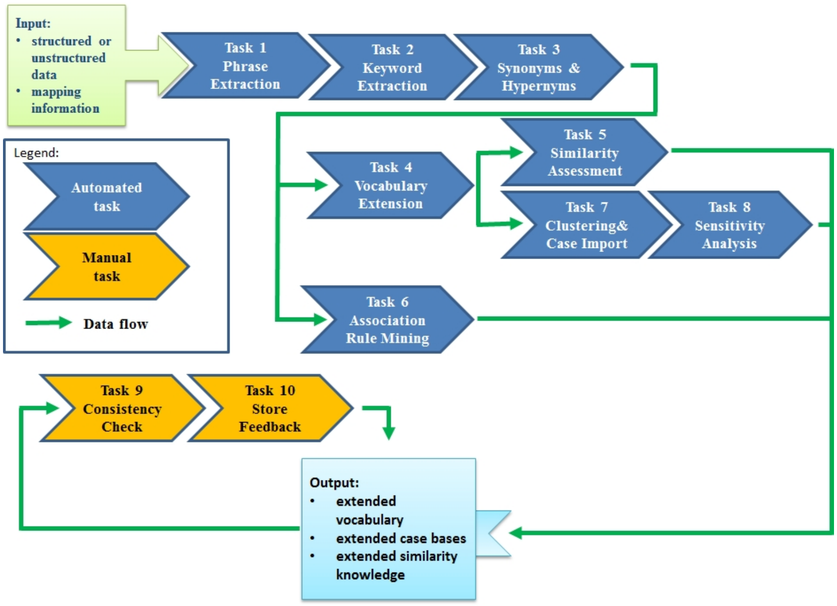
Input:
• structured or unstructured data
• mapping information
Task 1 Phrase Extraction
Task 2 Keyword Extraction
Task 3 Synonyms & Hypernyms
Legend:
Automated task
Manual task
Data flow
Task 4 Vocabulary Extension
Task 5 Similarity Assessment
Task 7 Clustering& Case Import
Task 8 Sensitivity Analysis
Task 6 Association Rule Mining
Task 9 Consistency Check
Task 10 Store Feedback
Output:
• extended vocabulary
• extended case bases
• extended similarity knowledge

Fig. 2. Workflow task and possible parallelization

Collocation extraction The first task is the identification and extraction of phrases from the free text components of the input data. The idea is to find recurring combinations of words based on standard english grammar and domain-specific terms. For the phrases an acceptable length between 2 and 6 words is chosen. This length is based on manual analysis of free texts and domain-specific phrases. There are no domain-specific phrases with more than 6 words and the correct identification of phrases with more than 6 words only reaches 30 percent and generates not much additional benefit. This task has three substeps: part of speech tagging, multi-word abbreviation identification, and phrase extraction. First, the free text is tagged to identify nouns, verbs, and adjectives. The next step is to identify multi-word abbreviations, because the longform of these abbreviations counts as phrases, too. The last step is to identify phrases based on the tagging information and the word position in a sentence. This task was set as the initial task, because for a successful phrase identification the whole free text is required. The result of this tasks is a modified free text, reduced by multi-word abbreviations and found phrases. For our example, the identified phrases are

- 'One hour', 'cabin crew', and 'flight attendant panel'.

Keyword extraction The second task is the extraction of keywords from the remaining text and consists also of three substeps: stopword elimination, lemmatization, and single-word abbreviation replacement. As input for this task, the modified text from task one is used. The stopword elimination is based on common english and a white list with words that should not be eliminated. The second substep identifies abbreviations in the remaining words and replaces them with their longform. For all words the lemmata are determined. We replaced the former stemming algorithm with a lemmatization algorithm, because the lemmatization algorithm is considering the context of a word in a sentence and therefore produces better results. The result of the complete task is a list of keywords, reduced to their base form. According to our example, the extracted keywords are

- 'before', 'departure', 'inform', 'maintenance', and 'freeze'.

Synonyms and hypernyms The third task is responsible for identifying synonyms and hypernyms for the extracted keywords and phrases. Therefore, the input for this task is a list of phrases from the first task and list of keywords from the second task. For every keyword the synonyms are identified, based on common english and domain-specific terms. One challenge in this task, is to consider the context and word sense of a keyword to identify the right synonyms. Therefore, we are using the part of speech information and a blacklist of words, that should not be used as synonyms. The second step is to identify the hypernyms for all keywords. There are two goals for this task. The first goal is to enrich the vocabulary of our CBR systems and the second goal is to use the synonyms and hypernyms to enhance our similarity measures by extending or generating taxonomies. The result of this task is a list of keywords and their synonyms and hypernyms. In our example, the result could be as follows:

- before: earlier, once
- departure: exit, movement, withdrawal
- inform: describe, make known
- maintenance: support, administration
- freeze: stop, paralyze, stuck, immobilize
- flight attendant panel: monitor, display

Vocabulary extension This task consists of adding the extracted keywords, phrases, synonyms, and hypernyms to the vocabulary of the CBR systems. The first step is to remove duplicate words and phrases to avoid redundant knowledge. The second step is to check the list of keywords against the list of phrases to identify keywords which occur as phrases. We want to slow down the growth of the vocabulary and therefore we identify keywords that are only occur as part of a collocation. These keywords are not added to the vocabulary. If a keyword occurs without the context of a collocation, it will be added.

Similarity measures When describing faults there are terms that are easily predictable and their similarity can be modeled by experts. However, when confronted with a large amount of manually inserted text from many sources it is virtually impossible to predict every concept that may appear, and how it stands in relation to other concepts. Therefore, this task is responsible for setting initial similarity values for newly discovered concepts and extends existing similarity measures. The first substep is to set similarity values between the newly added keywords and phrases and their synonyms. Therefore, the existing similarity matrices are extended and a symmetric similarity is proposed. The value itself could be configured, but we assume an initial similarity for synonyms of 0.8, based on the assumption that the similarity measures can take values from the [0;1] interval. The second step is to use the keywords, phrases, and hypernyms to extend or generate taxonomy similarity measures. The hypernyms serve as inner nodes, while the keywords and the synonyms are the leaf nodes. Keywords and their synonyms are sibling nodes if they have the same hypernym. This second step provides the possibility to model or extend similarity measures based on the layers of a taxonomy and therefore less similarity values have to be set. For values of keywords and phrases that could not be assigned to a taxonomy, no initial similarity value could be set, than 0. To overcome this hurdle, we employ social network analysis (SNA) methods to supplement the similarity between each two values of a given attribute. SNA is based on graph theory and utilizes the structure of the data and the relationships between the different items to reach conclusions about it, and has been used previously to measure the similarity of objects [2][15]. It is useful for our purposes since besides the structure of the data, which is readily available, no additional information is required. Our data consist of attribute-value pairs, representing different concepts of a fault.

We want to compute the similarity degree between the different values, for instance between two systems. In order to do so, we see our data as a weighted bipartite graph. On the left side are all the values of a given attribute, and on the right side the case diagnoses. Nodes A and B are then connected if a value represented by A appeared in a case that received the diagnosis represented by B. To eliminate multi-edges, edge weights represent the number of connections between each node pair. Since we are only interested in the similarity of nodes of type value, we perform weighted one-mode projection (WOMP)[16] on the left side of the graph. The resulting edge weights of the WOMP are the similarity degree between the nodes, and between the values they represent.

Association rule mining This task is used to analyze the keywords and phrases and find associations between the occurrence of these words within a data set as well as across data sets. Using association rule mining algorithms like the Apriori[1] or the FP-Growth[10] algorithm, we try to identify reoccurring associations to determine completion rules for our CBR systems to enrich the query. An association between keywords or phrases exists, when the combined occurrence exceeds a given threshold. For example, a combination between two keywords that occurs in more than 80 % of all analyzed documents, may be used as a completion rule with an appropriate certainty factor. To generate only completion rules with a high significance, a larger number of data sets have to be mined. Therefore, a minimum number of data sets has to be defined. Based on manual analysis of data sets in collaboration with aircraft experts, we assume a minimum of 10000 datasets and a confidence of 90 % will generate rules with the desired significance in the aircraft domain. In the aircraft domain many causal dependencies between systems, status, context and functions exist. Association rule mining can help identify this dependencies in an automated way to avoid the high effort from manually analyzing the data sets.

Clustering and case generation This task is responsible for generating a case from each input data set and storing it in a case base. To avoid a large case base with hundreds of thousands of cases, we cluster the incoming cases and distribute them to several smaller case bases. Generating an abstract case for each case base, a given query can be compared to the abstract cases and this way a preselection of the required case bases is possible. The first substep uses the mapping document to map the content of the document to a given case structure. The data from the documents are transformed into values for given attributes. In collaboration with experts from Airbus and Lufthansa we identified the aircraft type and the ATA chapter as the two most discriminating features of the cases. Therefore, the clustering algorithm uses these features to distribute the cases on the different case bases. For each aircraft type (A320, A330, A340, etc.) a set of case bases will be created and each set will be separated by the ATA chapter. The ATA chapter is a number with four or six digits and is used to identify a component of an aircraft. The cases are discriminated by the first two digits of the ATA chapter, which identify the component, while the other digits are used to specify a system of the component.

Sensitivity analysis In this task the feature weights for the problem description of the given case structure are determined. Not all attributes are equal. In retrieval tasks some attributes are more important to determine which objects are relevant, but how do we identify these attributes, and what is their degree of importance? Can some attributes be detrimental to retrieval? To answer these questions we used sensitivity analysis, and developed a method to calculate a relevance matrix of attributes. In our data each case has a diagnosis, and a diagnosis set consists of all the cases with the given diagnosis. While some attributes may be important to determine whether or not a case belongs to set A, other attributes might be more important for set B. This is why we have a relevance matrix, and not a vector. Our method is based on work done by [20] and [25], and includes three phases: 1. the static phase, where all attributes have the same weight for all diagnosis sets, and is used as a baseline to measure the contribution of the next two phases, 2. the initial phase, which includes a statistical analysis of the data set, and functions as the starting point of the next phase, 3. the training phase, where the values are optimized. The idea behind the training phase is that in a retrieval task there are two reasons for a false positive: first, the weights of attributes with a similar value are too high, and second, the weights of attributes with dissimilar values are too low. Much like the training phase of artificial neural networks, the contribution of each attribute to the error is calculated and propagated back through the weights, updating them accordingly. Within the OMAHA project, the analysis will be performed offline and the resulting relevance matrix will be embedded within the retrieval task. A more detailed description of the sensitivity analysis can be found in [23].

4.4 Current implementation

This section describes the current implementation status of our framework. FEATURE-TAK is an agent-based framework that uses the scalability of multi-agent systems and parallelization possibilities. In addition, an agent-based framework could easily be integrated into the multi-agent system for case-based diagnosis developed within the OMAHA project. For the implementation of the agents the JADE framework [8] was used. Currently, seven agents are implemented: supervising agent, gui agent, collocation agent, keyword agent, synonym agent, vocabulary agent and cluster agent. The supervising agent is the central coordinator of the framework and routes the communication between the other agents. The gui agent controls the user interface of the framework. He receives the input data and sends the information to the supervising agent. He also presents the interim results to the user and shows the status of the workflow. The collocation agent uses the Stanford CoreNLP library[17], a suite of NLP tools for part-of-speech tagging and collocation extraction. The keyword agent uses Apache Lucene[18] and the Stanford CoreNLP library for stopword elimination and lemmatization. The abbreviation replacement of both agents is an own implementation based on domain-specific abbreviations from Airbus and Lufthansa. For the synonym identification the synonym agent is using WordNet[14] and its databases of common english synonyms and hypernyms. The results of the first three tasks are passed to the vocabulary agent. This agent uses the myCBR[7] API to access the knowledge model of the CBR systems. The last implemented agent is the cluster agent, which generates the cases from the data sets and distributes them to the different case bases. The clustering algorithm is an own implementation and the myCBR API is used to pass the generated cases to the correct case base. The functionality for similarity assessments, association rule mining and sensitivity analysis are implemented, but not integrated into the framework yet. Different import mechanisms are implemented to process data from CSV files and text files like word documents or PDF files. Because of the different content and data structures of the documents, the data is processed differently for each document type. CSV files and result sets are processed row-wise, while text documents are processed in the whole. The mapping file is written in XML format and contains information about which column in a CSV file or result set should be mapped to which attribute in the case structure. For text documents the mapping is far more difficult and not completely implemented yet.

4.5 Evaluation

The current implementation of the framework was tested with a CSV file containing 300 real world data sets. On these data sets five tasks were computed: collocation extraction, keyword extraction, synonym search, vocabulary extension, and case generation and clustering. The following results were generated:

- Collocation extraction: 2465 phrases extracted, 2028 distinct phrases
- Keyword extraction: 8687 keywords extracted, 1464 distinct keywords
- Synonym search: 21285 synonyms identified, 3483 distinct synonyms
- Vocabulary extension: 4621 concepts added to the vocabulary
- Case generation and clustering: 300 cases distributed over 8 case bases

The results of the workflow were evaluated by experts from Airbus Operations GmbH and Lufthansa Industry Solutions. The extracted collocations and keywords were compared against the original fault description, while the synonyms were checked for adequate word sense. The added concepts were checked to identify duplicates or false entries. The following graphic illustrates the evaluation results.

While we have good results for the collocation and keyword extraction, we have poor results for the synonyms identification. The reason is that our word sense disambiguation is just based on black and white lists and therefore our synonym task identifies a great number of synonyms with inappropriate word sense. Therefore, the word sense disambiguation has to be improved with state of the art approaches. In addition, we conducted a performance evaluation of the

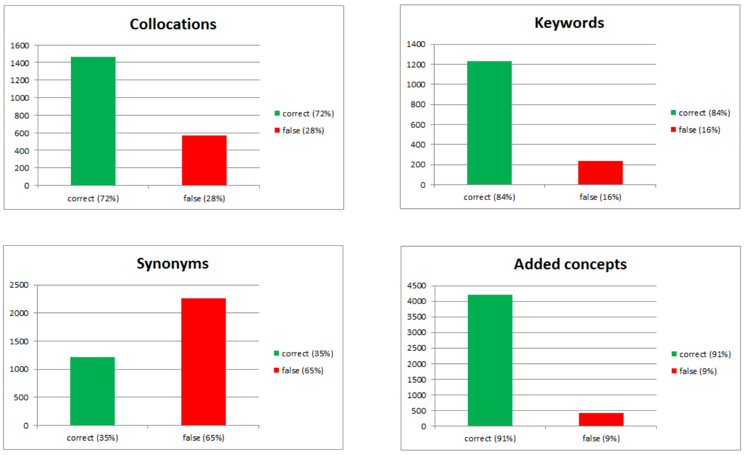

Fig. 3. Evaluation results

implemented tasks. Therefore, we run the workflow with different sized CSV files: 10, 20, 100, 150, and 300 data sets. Figure 4shows the results. The y axis contains the time in seconds and the x axis the number of data sets. With an increasing number of data sets, the computation time appeared to grow exponentially. We identified the myCBR tool as the main cause for this performance problem during the task of the vocabulary extension.

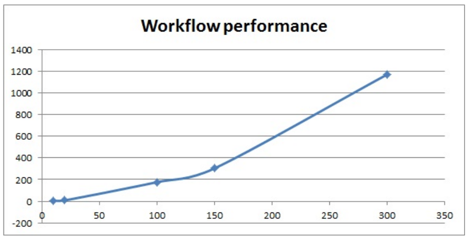

Fig. 4. Workflow performance

5 Summary and Outlook

In this paper we describe the concept and implementation of our framework FEATURE-TAK. The framework was developed to transform textual information in the aircraft domain into knowledge to be used by structural CBR systems. We give an overview of the framework architecture and describe the individual tasks in more detail. In addition, we describe the status of our current implementation. The newly improved version is still in an evaluation process and will be tested with a larger data set based on historical problem data from Airbus. We will test the framework with input data of more than 65.000 single data sets. Based on the evaluation results we will improve the framework methods. A specific challenge is the word sense disambiguation. We will address this challenge using pattern recognition and neural networks. In addition, we will integrate the remaining functionality into the framework and connect it with the corresponding agents. After the complete implementation of the framework, it will be integrated into the diagnosis system to provide the frameworks functionality for knowledge modeling and maintenance purposes. In addition to improvement on the semantic level, we also will improve the performance and scalability of the framework to support the computation of large data sets. For further development we plan to modularize and generalize the tasks and substeps to get a framework with domain-independent and domain-specific components, that could be configured for the use in different domains. We also want to support an interface for different additional NLP or CBR methods and tools, to provide the user with a greater variety on analysis, extraction and transformation possibilities.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Proceedings of the 20th International Conference on Very Large Data Bases. pp. 487–499. VLDB '94, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1994), http://dl.acm.org/citation.cfm?id=645920.672836
2. Ahn, Y., Ahnert, S., Bagrow, J., Barabasi, A.: Flavor network and the principles of food pairing. Scientific reports 1 (2011)
3. Althoff, K.D.: Collaborative multi-expert-systems. In: Proceedings of the 16th UK Workshop on Case-Based Reasoning (UKCBR-2012), located at SGAI International Conference on Artificial Intelligence, December 13, Cambride, United Kingdom. pp. 1–1 (2012)
4. Althoff, K.D., Bach, K., Deutsch, J.O., Hanft, A., Mänz, J., Müller, T., Newo, R., Reichle, M., Schaaf, M., Weis, K.H.: Collaborative multi-expert-systems – realizing knowledge-product-lines with case factories and distributed learning systems. In: Baumeister, J., Seipel, D. (eds.) KESE @ KI 2007. Osnabrück (Sep 2007)
5. Althoff, K.D., Reichle, M., Bach, K., Hanft, A., Newo, R.: Agent based maintenance for modularised case bases in collaborative mulit-expert systems. In: Proceedings of the AI2007, 12th UK Workshop on Case-Based Reasoning (2007)
6. Bach, K.: Knowledge Acquisition for Case-Based Reasoning Systems. Ph.D. thesis, University of Hildesheim (2013), dr. Hut Verlag Mnchen
7. Bach, K., Sauer, C.S., Althoff, K.D., Roth-Berghofer, T.: Knowledge modeling with the open source tool mycbr. In: Proceedings of the 10th Workshop on Knowledge Engineering and Software Engineering (2014)
8. Bellifemine, F., Caire, G., Greenwood, D.: Developing multi-agent systems with JADE. Jon Wiley & Sons, Ltd. (2007)
9. BMWI: Luftfahrtforschungsprogramms v (2013), http://www.bmwi.de/BMWi/Redaktion/PDF/B/bekanntmachung-luftfahrtforschungsprogramm-5,property=pdf,bereich=bmwi2012,sprache=de,rwb=true.pdf

10. Borgelt, C.: An implementation of the fp-growth algorithm. In: Proceedings of the 1st International Workshop on Open Source Data Mining: Frequent Pattern Mining Implementations. pp. 1–5. OSDM '05, ACM, New York, NY, USA (2005), http://doi.acm.org/10.1145/1133905.1133907
11. Ceausu, V., Despres, S.: A semantic case-based reasoning framework for text categorization. In: The Semantic Web, Lecture Notes in Computer Science. pp. 736–749 (2007)
12. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Aswani, N., Roberts, I., Gorrell, G., Funk, A., Roberts, A., Damljanovic, D., Heitz, T., Greenwood, M.A., Saggion, H., Petrak, J., Li, Y., Peters, W.: Text Processing with GATE (Version 6) (2011), http://tinyurl.com/gatebook
13. Drozdzynski, W., Krieger, H.U., Pisorski, J., Schfer, U.: Sprout a general-purpose nlp framework integrating finite-state and unification-based grammar formalisms. In: Finite-State Methods and Natural Language Processing. pp. 302–303 (2006)
14. Feinerer, I., Hornik, K.: wordnet: WordNet Interface (2016), https://CRAN.R-project.org/package=wordnet, r package version 0.1-11
15. Jeh, G., Widom, J.: Simrank: a measure of structural-context similarity. In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 538–543 (2002)
16. Lapaty, M., Magnien, C., Vecchio, N.D.: Basic notions for the analysis of large two-mode networks. Social Networks 30, 31–48 (2008)
17. Manning, C.D., Mihai, S., Bauer, J., Finkel, J., Bethard, S., McClosky, D.: The stanford corenlp natural language processing toolkit. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations. pp. 55–60 (2014)
18. McCandless, M., Hatcher, E., Gospodnetic, O.: Lucene in Action, Second Edition. Manning Publications Co., Greenwich, CT, USA (2010)
19. Reuss, P., Althoff, K.D.: Explanation-aware maintenance of distributed case-based reasoning systems. In: LWA 2013. Learning, Knowledge, Adaptation. Workshop Proceedings. pp. 231–325 (2013)
20. Richter, M.: Classification and learning of similarity measures. In: Concepts, Methods and Applications Proceedings of the 16th Annual Conference of the Gesellschaft fr Klassifikation e.V. pp. 323–334 (1993)
21. Rodrigues, L., Antunes, B., Gomes, P., Santos, A., Carvalho, R.: Using textual cbr for e-learning content categorization and retrieval. In: Proceedings of International Conference on Case-Based Reasoning (2007)
22. Sizov, G.V., Ozturk, P., Styrak, J.: Acquisition and reuse of reasoning knowledge from textual cases for automated analysis. In: Lecture Notes in Computer Science. pp. 465–479. Springer International Publishing (2009)
23. Stram, R., Reuss, P., Althoff, K.D., Henkel, W., Fischer, D.: Relevance matrix generation using sensitivity analysis in a case-based reasoning environment. In: Proceedings of the 25th International Conference on Case-based Reasoning, ICCBR 2016. Springer Verlag (2016)
24. Weber, R., Aha, D., Sandhu, N., Munoz-Avila, H.: A textual case-based reasoning framework for knowledge management applications. In: Proceedings of the ninth german Workshop on Case-Based Reasoning. pp. 244–253 (2001)
25. Wess, S.: Fallbasiertes Problemlsen in wissensbasierten Systemen zur Entscheidungsunterstttzung und Diagnostik. Ph.D. thesis, TU Kaiserslautern (1995)
26. Wettschereck, D., Aha, D.: Feature weights. In: Proceedings of the First International Conference on Case-Based Reasoning. pp. 347–358 (1995)