THE CHALLENGE OF AUTOMATICALLY ANNOTATING SOLUTION DOCUMENTS

Comparing Manual and Automatic Annotation of Solution Documents in the Field of Mechanical Engineering

Andreas Kohn, Udo Lindemann
Institute of product development, Technische Universität München, Boltzmannstraße 15, 85748 Garching, Germany

Gerhard Peter
Festo GmbH & Ko. KG, Plieninger Straße 50, 73760 Ostfidern-Schamhausen, Germany

Keywords: Solution knowledge, Manual annotation, Evaluation of annotation.

Abstract: This paper contains part of the actual research in the use case PROCESSUS of the German research program THESEUS. A case study about comparing manual and automatic annotation of solution documents in the field of mechanical engineering is described. A set of six solution documents was annotated manually by four users. Then, the same set of documents was annotated automatically by an ontology-based system. The two annotations are compared considering proposed ranking numbers. These ranking numbers give the weighting of annotations according to the overall and merged manual annotations. Therewith, they serve as a reference for the expected result of the automatic annotation. Comparing the automated and the manual annotation can not only reveal limitations of the automatic annotation process but also raise interesting questions to what extent domain specific knowledge has to be represented in the ontology.

1 INTRODUCTION

In product development, the access to existing knowledge about previous solutions may reduce the amount of development cycles and conception rework and therewith reduce the efforts of time and costs. Principally, various sources exist for supporting this knowledge. In a study in the German automation industry, sources for the search of existing solution knowledge were identified (Ponn et al., 2006). Besides direct personal communication, organisation-internal knowledge sources (e.g. project folders or databases), construction catalogues, internet portals, and publically available marketing documents were identified as mostly used.

However, an engineer who wants to retrieve existing solution knowledge may face several barriers (see Figure 1). First of all, solution knowledge is mostly unstructured and the access to unstructured data is often insufficient (Blumberg et al., 2003). Secondly, different wordings are used by the involved developers (Dylla, 1990). This different wording hinders the access via a normal full-text search (Pocsai, 2000). Furthermore, varying taxonomies and classifications due to different viewpoints in sales, marketing, and engineering (Hepp, 2003) contribute to the barrier that hinders the access to needed solutions.

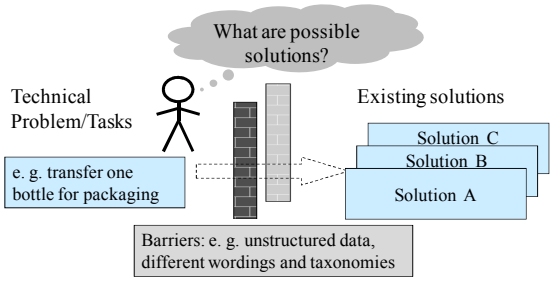

Figure 1: Barriers in the process of retrieving technical solutions.

Improving this process of retrieving existing solutions is one of the main goals of the use case PROCESSUS as one part of the German research

Kohn A., Lindemann U. and Peter G..
THE CHALLENGE OF AUTOMATICALLY ANNOTATING SOLUTION DOCUMENTS - Comparing Manual and Automatic Annotation of Solution Documents in the Field of Mechanical Engineering .
DOI: 10.5220/0003055201530158
In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development* (KEOD-2010), pages 153-158
ISBN: 978-989-8425-29-4

project THESEUS. Within PROCESSUS, an ontology has been developed, that is used for capturing the knowledge of technical solutions (Gaag et al., 2009). The instances of the ontology and the modelled relations can also be used as a vocabulary for automated annotation of solution documents. This annotation should help in the later retrieval of the documents.

This paper focuses on improving the process of annotating unstructured text data stored in publicly available solution documents of the automation industry. In these documents, companies provide information about previously installed solutions (e.g. a bottling and filling line for beverages). They are mostly used for marketing purposes to give references of previous work. Furthermore, they are useful in generating first ideas how to approach an engineering task.

In engineering design theory, technical solutions can be described by their functions - typically composed of an object and an operation performed on the object (Ponn et al., 2008). Given a solution document with a certain number of different functions, an annotation tool that identifies most of these functions but not the really important ones is surely not the best one. Due to these uncertainties, generally applied methods of ranking like term frequency or the evaluation of annotations with precision and recall can hardly be applied here.

To evaluate and improve annotations, it is necessary to get a deeper insight into the content of the existing solution documents. For this purpose, solution documents are analysed by comparing manual annotations made by different persons. These manual annotations are merged and by applying ranking numbers the most relevant content of the document concerning the technical functions of the solution is identified. Subsequently, these ranking can be used to evaluate the automated annotation. This procedure is exemplarily tested with six solution documents and applied on the developed annotation tool of our prototype.

The paper is organised as follows: First we will provide a short overview of the ontology (the main concepts and their relations) and its use in the developed prototype. The technical functionalities of the prototype will not be described in detail and only as far as it is relevant for this work. Second, we describe our methodology. Then, we will describe the case study and results in detail. This is followed by a review of related work. We will conclude the paper with a discussion and summary of our findings and provide an outlook on the next steps to take.

2 USAGE OF THE ONTOLOGY IN THE PROTOTYPE

A prototype was implemented that uses the developed ontology (as an OWL ontology) to support the automated annotation and the subsequent search for solution documents. For the automated annotation, the ontology serves as a vocabulary and provides the needed information about the existing relations of elements belonging to technical solutions. The base structure with the core concepts of this ontology is shown in Figure 2. The function has the central position. It is realised by a technical solution, used in a special industrial sector, executed by a function owner, and performs a certain operation on a decent object. Existing solutions can be described by instantiating these concepts with the appropriate instances.

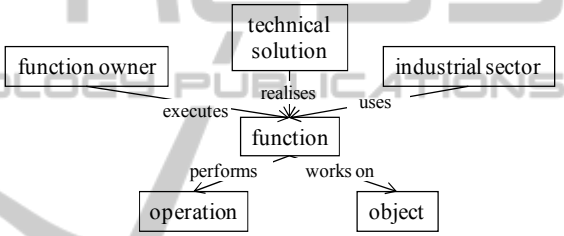

Figure 2: Base structure of the domain-specific ontology to support solution retrieval.

For the automated annotation, the prototype uses the label property of the instances in the ontology to recognize the appropriate words and attach the corresponding concept to the document. Linguistic features as word stemming and flexion of words are considered. Also, linguistic algorithms are supposed to analyze the syntax of a sentence and to determine relations between the words in a sentence. The annotation process will be illustrated by a simple exemplary sentence “The conveyor belt transports the boxes” taken from one of the solution documents. “Conveyor belt” is the function owner which performs the operation “transports” on the object “box”. If these instances are available in the ontology, the corresponding concepts are annotated. With the help of the linguistic algorithms, the combination of “transport” and “box” in one sentence leads to the annotation of the function “transport box”.

Figure 3 shows screenshot of this prototype with an exemplary result of an automated annotation. On the left side, the annotated instances of a document are listed according to the concepts chosen as annotation filter (property of a solution, industrial

sector, etc.). On the right side, a graph browser offers the possibility to navigate through the ontology and adding further annotations manually.

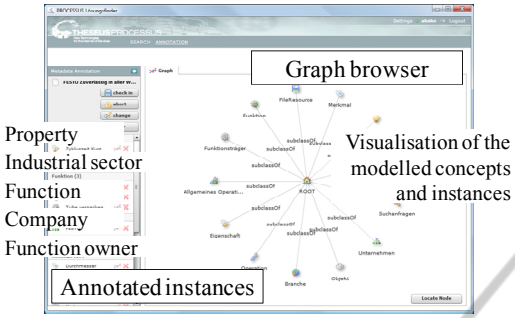

Figure 3: Screenshot of the prototype.

3 ANALYSING MANUAL ANNOTATIONS

This section shows the procedure of manually annotating the documents and merging these annotations. Afterwards, the applied ranking numbers for the annotated instances and their use for the evaluation of automatic annotation are explained.

3.1 Manual Annotation of Documents

To get a deeper insight into the content of the solution documents, they are analysed by a comparison of manual annotations. Test participants were asked to identify all function owners and the corresponding functions. There was no limitation of the number of maximum function owners or functions annotated in each document. It was also allowed to annotate only function owners without a corresponding function or vice versa.

As a result of the single manual annotation, a set of function owners and corresponding functions (operation and object) emerges. Additionally, the position of the source for the annotation in the document was marked in order to identify where the annotation stems from.

3.2 Merging of the Manual Annotations

Subsequently, the manual annotations of one document are merged to give an overview over the similarities and differences of the single manual annotations. The manual annotations are merged according to their appearance in the document. Figure 4 gives a short overview of the merging procedure.

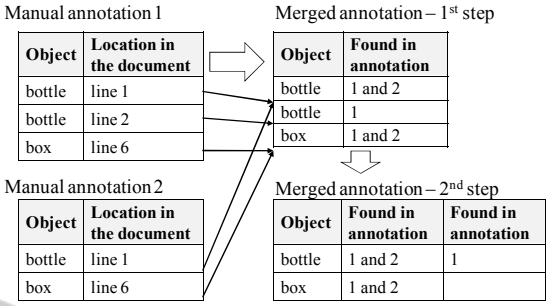

Figure 4: Merging of the manual annotations.

When, for example the same part of the document is annotated by more than one person (in this example the objects “bottle” and “box” in line 1 and 6 by person 1 and person 2), it is only added once to the merged annotation. While merging the documents, both the number of overall different annotations of one concept and the number of equal annotation between the single manual annotations becomes evident. In the example, the instance “box” was annotated once by two persons, while the instance “bottle” was annotated twice (one time by two persons, the other time by one person).

Therewith, the merged annotations can be interpreted as a very precise annotation as possibly missed annotations of one single annotation can be found in the annotation of another person.

3.3 Appliance of Ranking Numbers

The number of equal annotations within the single manual annotations gives a first impression about major or minor important instances. When an instance of a concept or two instances of two related concepts are annotated by a high number of people, they can be interpreted as important for the document. In the example presented above, the instance “bottle” is annotated by two people in line 1 but only once in line 2. This indicates that in the second line, the one person did not interpret the “bottle” in this sentence as an important object for this solution.

Additionally to this simple measurement, two more ranking numbers are proposed. These ranking numbers show similarities to the term frequency used in information retrieval (Salton et al., 1986). In contrast to the term frequency, not the importance of a word in a document, but the importance of an annotation according to all annotations of the respective document is focused here. Furthermore,

the ranking numbers combine the amount of overall annotations of an instance within all manual annotations with the number of annotations of an instance after the merging of the manual annotations. By this combination, the error rate of a single manual annotation is decreased while the "overall intelligence" of several manual annotations is increased.

The first ranking number *R(i)* considers the annotation of instances of single concepts. It is calculated by the multiplication of the overall number of similar annotated instances of a single concept *N(io)* with the number of similar annotated instances of a single concept after merging the manual annotations *N(im)*. To normalise the number, the product is divided by the product of the maximums of *N(io)* and *N(im)* over all instances (equation 1).

$$R(i) = \frac{N(io) * N(im)}{\max\big(N(io)\big) * \max\,(N(im))} \quad (1)$$

In the example in Figure 4, the ranking numbers are calculated as shown in Table 1.

Table 1: Calculation of the ranking numbers.

Object	N(io)	N(im)	R(i)
bottle	3	2	1
box	2	1	0,33

The instance "box" was annotated twice in line 1 and once in line 2, so the overall number of annotations *N(io)* is 3. It is annotated in line 1 and 2 which makes the *N(im)* equal to 2.

The second ranking number *R(r)* - and from the ontological point of view the more interesting one - considers the annotation of instances of related concepts. Similar to *R(i),* it is calculated by the multiplication of the number of overall annotations and the number of annotations after merging the manual annotations. This time, the numbers are only counted when the annotation contains a pair of instances belonging to concepts that are related in the ontology. Once again, it is normalised by the maximum of these numbers *N(ro)* and *N(rm)* as shown in equation 2.

$$R(r) = \frac{N(ro) * N(rm)}{\max\big(N(ro)\big) * \max\,(N(rm))} \quad (2)$$

Therewith, the ranking number R(r) provides information about the mutual annotation of instances that are related according to the ontology.

3.4 Interpretation

The ranking numbers take values between 0 and 1. These ranking numbers, applied to each instance or related instances, give the weighting according to the overall and merged manual annotations and therewith the reference for the expected result of the automatic annotation. The automatic annotation has to identify at least the highest ranked instances. Especially R(r) can be used for evaluating the quality of the annotation of related instances.
With the help of the ranking numbers, precision and recall measures for the evaluation of the automatic annotation can be calculated with a higher granularity. It is more important to find higher ranked instances than lower ranked ones.

4 CASE STUDY

This section shows the application of the above described steps of annotating and merging the documents. The annotated documents and the results of the annotations are presented and finally compared with the automatic annotation of the developed prototype. Four persons of different background (marketing, computer science and mechanical engineering) were asked to manually annotate six solution documents concerning the contained function owners and their corresponding functions.

The documents describe technical solutions in the field of automation technology (see Table 2 for a short overview of the content of the documents). Their length varies between 2 and 8 DIN A4 pages and their number of words lies between 343 and 912.

Table 2: Overview of the used documents.

Content	Pages	Words
Packaging of medical tablets	2	877
Separation of small components	4	750
Sorting of empty bottles	2	343
Bottling of bottles	2	741
Palletizing of bread	2	752
Packaging of drink crates	8	912

4.1 Merging of Manual Annotations

By the example of one document (packaging of medical tablets), the results of the manual annotation and the merging shall be explained. Table 3 shows the numbers of annotations of instances of the four concepts. The first column shows the number of

overall annotations after merging; the following columns show the number of annotations of the individual manual annotations.

Table 3: Number of annotations.

Concept	all	1	2	3	4
Operation	30	28	14	12	14
Object	27	24	14	9	14
Function owner	16	13	11	9	8
Function	27	24	14	9	14

The four participants differ in the number of annotations made. In subsequent interviews it was identified that this can be explained due to the different professional backgrounds. A mechanical engineer did not consider every function as "important". He focused on the core functions. In a subsequent search, he expects these functions to be ranked higher than other functions.

By merging these annotations the number of different annotations of instances of the four concepts can be identified. In this document, 26 different operations, 14 objects, 7 function owners, and 26 different functions were identified.

Table 4 gives an exemplary overview of the instances of the concept "function owner". The corresponding values of *N(io)* and *N(im)* are presented and the resulting *R(i)*-values shown.

Table 4: Annotations of the concept "function owner".

Function owner	*N(io)*	*N(im)*	*R(i)*
Robot	17	7	1,00
Machine	8	3	0,20
Conveyor belt	4	1	0,03
Barcode reader	4	1	0,03
Operator	2	2	0,03

The instance "robot" was annotated 7 times in the document and was mentioned 17 times altogether by the four annotators. This identifies this instance as most relevant for the annotation of function owners.

4.2 Evaluation of the Automatic Annotation

With the help of these ranking, numbers, the automatic annotation can be evaluated. Table 5 shows exemplary which terms have been annotated as functions owners in the document by the automated annotation process. As illustrated, the most important function owner (*R(i)* = 1) has been identified. Nevertheless, some instances have not been automatically annotated.

Table 5: Comparing with the automated annotation.

Function owner	*R(i)*	*Autom. annotation*
Robot	1,00	found
Machine	0,20	not found
Conveyor belt	0,03	found
Barcode reader	0,03	found
Operator	0,03	not found

The results of the evaluation of function owners, operation und object over the six documents were quite similar. Only the annotation of technical functions did not achieve the expected results. This result can be explained by the fact, that the linguistic algorithms do not properly recognise when an object and an operation constitute a technical function.

5 RELATED WORK AND DISCUSSION

An overview of general methods and tools for semantic annotation is given by Uren et al. (2006). Uren et al. proposed seven requirements for ontology-supported annotation and evaluated twenty-seven annotation tools. Especially automatic annotation was mentioned as an important field for further improvement. Corcho (2006) compared different annotation approaches (ontology, thesauri and controlled vocabulary) for supporting the process of creating metadata. He identified ontology-based annotation as the most powerful annotation approach concerning the annotation of relations between the instances of a document and also emphasized the meaning of improving automated annotation. A domain ontology as knowledge base for information retrieval is used to improve search over large document repositories by Vallet et al. (2005). In their approach, Vallet et al. also used a label property to identify potential occurrences of instances in the annotated documents.

The high amount of work for manually annotating and the following merging make this approach only limited applicable for a larger number of documents and questionable concerning its statistical validation. Furthermore, the influence of the personal background has to be considered when interpreting the results of the manual annotations. Nevertheless, in addition to the identification of ranked instances for the annotation, this approach is twofold useful: First of all, by analysing and verifying the manual annotations, linguistic and syntactic properties of the solution documents can be identified. In a next step, these can be used to deduce typical linguistic schemes (e.g. the syntax of sentences) of solution documents for improving the

automated annotation. Secondly, the merging of the manual annotation and its later validation is useful for obtaining a set of well-annotated documents for further evaluation of automatic annotations.

The findings of this work can be used in other domains of knowledge where unstructured data has to be annotated using a domain-specific ontology. In this context, it has to be considered, where the needed knowledge is stored. If using only instances for the annotation, the ontology could become huge. For example, if every function owner should be part of the ontology, huge classifications or standards have to be integrated. For instance, transferring the products and services categorization standards eCl@ss in OWL yielded 75,000 ontology classes plus more than 5,000 properties (Hepp, 2006). Alternatively, you may use a combination of ontological knowledge and linguistic patterns (or rules) for annotation. For example, modelling only on the (technical) operations in the ontology and defining patterns to annotate a technical function in combination with an identified noun in the sentence would decrease the size of the ontology, as the number of technical operations is limited. However, the number of rules to be defined will increase. What works best has to be judged considering the relevant domain and the complexity of the modelled knowledge.

6 CONCLUSIONS AND OUTLOOK

The analysis of solution documents done in this research permits an insight into the content of solution documents in the field of automation technology. With the help of the proposed ranking numbers, important instances can be identified according to the manual annotations made by different persons. This ranking numbers can be subsequently used for the evaluation of an automated annotation. The evaluation of the used prototype showed need for improvement concerning the annotation of related instances in the ontology.

To improve this annotation, further work will focus on the interpretation of the made analyses for identifying patterns in the syntax or layout of solution documents. Furthermore, the personal background of the manual annotations will be considered for the purpose of identify individual requirements on the annotation. This will improve the automatic annotation and may also be instrumental to identifying the “core functions” of a technical solution.

ACKNOWLEDGEMENTS

This work has been funded by the German Federal Ministry of Economy and Technology (BMWi) through THESEUS. The authors wish to acknowledge gratitude and appreciation to all the PROCESSUS project partners for their contribution during the development of various ideas and concepts presented in this paper.

REFERENCES

Blumberg, R., and Atre, S. (2003). The Problem with Unstructured Data. In *DM Review,* 13(2).

Corcho, O. (2006). Ontology based document annotation: trends and open research problems. *Int. J. of Metadata, Semantics and Ontologies,* 1(1), 47–57.

Dylla, N. (1990). *Denk- und Handlungsabläufe beim Konstruieren*, PhD thesis, Technische Universität München.

Gaag, A., Kohn, A., and Lindemann, U., (2009). Function-based Solution Retrieval and Semantic Search in Mechanical Engineering. In *ICED’09, 17th International Conference on Engineering Design,* Stanford, California, USA.

Hepp, M. (2003). *Güterklassifikation als semantisches Standardisierungsproblem*. Wiesbaden: Deutscher Universitäts-Verlag.

Hepp, M. (2006). Products and Services Ontologies: A Methodology for Deriving OWL Ontologies from Industrial Categorization Standards. *Int. J. on Semantic Web & Information Systems*, 2 (1), 72-99.

Ponn, J., and Lindemann, U. (2008). *Konzeptentwicklung und Gestaltung technischer Produkte*. Berlin: Springer.

Pocsai, Z. (2000). *Ontologiebasiertes Wissensmanagement für die Produktentwicklung*. PhD thesis, Technische Universität Karlsruhe.

Ponn, J., Deubzer, F., and Lindemann, U., (2006). Intelligent Search for Product Development Information - an Ontology-based Approach. In: *DESIGN’06, 9th International Design Conference*, Dubrovnik, Croatia.

Salton, G. and McGill, M. J. (1986). *Introduction to Modern Information Retrieval*. New York: McGraw-Hill.

Uren, V., Cimiano, P., Iria, J. e., Handschuh, S., Vargas-Vera, M., Motta, E., and Ciravegna, F. (2006). Semantic annotation for knowledge management: Requirements and a survey of the state of the art. *Journal of Web Semantics,* 4(1), 14-28.

Vallet, D., Fernández, M., and Castells, P., (2005). An Ontology-Based Information Retrieval Model. In *ESWC’05, The Semantic Web: Research and Applications, Second European Semantic Web Conference.* Heraklion, Crete, Greece.