On the relationship between ontology-based and holistic representations in a knowledge management system

Sylvia Melzer
Hamburg University of Technology, Germany

ABSTRACT

This chapter presents a way for systematically combining ontology-based and holistic-based content descriptions in context of knowledge management in order to increase recall while at least maintaining precision.

INTRODUCTION

Usually, Knowledge Management (KM) systems are based on Content Management (CM) systems. In CM systems, content is stored, organized, and supplemented by metadata. Among simple descriptions for authors, characters, publishers, and so on, nowadays, metadata contain feature-based as well as ontology-based content descriptions (see Figure 1). For representing ontology-based content descriptions formal languages are used (Baader, Calvanese, McGuinness, Nardi, and Patel-Schneider, 2003). For instance, ontology-based content descriptions can be represented via logic-based techniques (Kaya, 2011; Espinosa, 2011).

Applications exploit ontology-based content descriptions in various ways, for example, in the semantic web content descriptions are used for finding documents, images, videos, or persons. Search requests are specified by posing queries in formal languages such as string patterns, logic-based queries, and so on. For matching queries with context, each query language has its pros and cons (Melzer, 2006). For most purposes, string patterns have a high recall but do not lead to high precision. Recall is defined as the number of relevant items retrieved divided by the number of relevant items in the repository. Precision is defined as the number of relevant items retrieved divided by the overall number of retrieved items. In practice it is difficult to maximize precision and recall simultaneously.

Until now, information retrieval (IR) processes are rarely based on ontology-based content descriptions for matching queries with content. Usually, so called holistic content descriptions and corresponding similarity measures are used for information retrieval (Manning, Raghavan, and Schütze, 2008). Matches on holistic content descriptions can be realized, for example, using nearest-neighbor algorithms (Manning et al., 2008, pp. 403–419).

Insert Figure 1 Here

Figure 1. Content representation types.

In short and slightly exaggerating, holistic representations lead to high recall and low precision, and ontology-based representations tend to be characterized by low recall and high precision (Blank, Meeden, and Marshall, 1992; Espinosa, Kaya, Melzer, Möller, and Wessel, 2007a; Espinosa et al., 2007b). In this context, it is desirable to increase recall while at least maintaining precision. This kind of improvement could be achieved by systematically combining ontology-based and holistic content descriptions. To the best of our knowledge, a combination of both kinds of content descriptions has not yet been investigated in a methodological way. It is a central idea of this chapter to suggest a way for systematically combining ontology-based and holistic content descriptions in order to increase recall while at least maintaining precision. In the following, retrieved documents with high precision will be called high-quality documents.

Retrieval of high-quality documents is a frequent task in KM contexts, in the sense that the documents themselves or, in some applications, their authors are subjects of further steps in KM processes. However, finding documents might be a problem in case that there is no direct match with simple queries. Consequently, queries need to reformulated, which usually is a rather difficult task for users. This is true for pattern-based as well as logic-based queries (Melzer, 2006). Indeed, if there are at least some query results, we argue that these results can be analyzed and exploited for detecting relevant additional material in order to find high-quality documents.

Let us assume there are some matches w.r.t. holistic content descriptions. The holistic representations are exploited for similarity searches, which increases the recall. Then our idea is to switch from holistic representations to ontology-based representations (see Figure 1). Then, ontology-based representations

are giving more precise matches. Accordingly, the holistic representations of these matches being found are then used as a filter in order to increase recall and precision.
This chapter has the objective to increase recall while at least maintaining precision by suggesting a way for the combination of holistic and ontology-based content descriptions in a knowledge management environment. In this context, it should be noted that this chapter has not the objective to present absolute numbers or performance measurements.

Related Work

In the past, logic-based techniques such as the non-standard annotation inference service Abox abduction have been studied and developed for representing ontology-based content descriptions (Kaya, 2011; Espinosa, 2011). Ontology-based representations (annotations) describe documents, images, videos, or persons and can be seen as an interpretation of content. The combination of several interpretation results is called fusion. First investigations on a fusion process of multimedia interpretation results were done by Kaya (2011) in order to combine ontology-based representations with the result to increase precision. In this process, annotated entities of a multimedia document, also called individuals, will be fused with other one's if the individuals describe the same real-world entity.

Holistic content descriptions are much better established (Manning et al., 2008). An investigation in this thesis will be based on the Latent Semantic Indexing (LSI) approach as defined in (Manning et al., 2008, pp. 403–419) because the LSI technique has a substantially high recall (Gee, 2003).

First investigations on the combination of different content descriptions were done in the past (Touretzky, 1990; Sun and Peterson, 1998) whereby content descriptions are called paradigms. The researchers distinguish between symbolic (ontology-based) and subsymbolic paradigms. In this context, holistic representations are a part of the particular subsymbolic representation. For the representation of content, researchers and developers used Artificial Intelligence (AI) technologies to represent ontology-based content descriptions, and cognitive techniques to represent subsymbolic (holistic) content descriptions. However, hybrid architectures or models which are based on both paradigms are not completely characterized (Touretzky, 1990), but most experts agree that a combination of ontology-based and holistic content descriptions is necessary for the development of efficient hybrid architectures (Sun and Peterson, 1998; Reuters, 2010; W3C, 2011}. In 2004, Birbeck proposed a new syntax, RDFa, for the combination of ontology-based and holistic representations. RDFa is syntax for embedding RDF within HTML. Combining content into a conceptual paradigm seems to be a good solution but, in this case, content cannot be seen as an abstraction. Additionally, absolute numbers and performance measurements are not known yet. In the following, it can be said in summary that ontology-based and holistic content descriptions should be seen both as abstractions, and it make sense to follow the idea to combine different content descriptions with the result to find high-quality documents in all heterogeneous sources.

The innovative and significant approaches for representing content, pose exceptional technological challenges in order to increase precision on the one hand, and recall on the other. We argue that the combination of both approaches in a systematically way achieves better matching results. In other words, IR processes which are based on systematically combined holistic and ontology-based content descriptions for matching queries with content results in high recall without an associated decrease of precision.

Symbolic and holistic content descriptions of a document represent knowledge. In most companies a lot of knowledge is available but the employees often do not know how to utilize all available knowledge. In 2008, Nonaka showed how employees in a company can exploit their knowledge for innovation and how they reinvent themselves continually in the face of unremitting change. Nonaka defined models for the management of knowledge which support the achievement of productivity advantages. In this chapter,

these models are the basis for the development and management of ontology-based and holistic content descriptions in the context of KM.

CONTENT AND KNOWLEDGE MANAGEMENT CHARACTERISTICS

Conceptual content and knowledge management systems might have the same purpose, namely the management of content. The way in which content is handled depends on the data model used but data models are limited by technical constrains of the target system, for example databases. A conceptual model can avoid such technical constraints as shown in (Sehring and Schmidt, 2004). Schmidt and Sehring introduced assets in order to specify such a model (Schmidt and Sehring, 2003).

The main purpose of this section is to give an overview about the characteristics of conceptual content and knowledge management, and the analysis of KM and CCM notions which are essential for the formalization of KM notions via semantic assets.

Conceptual Content Management

Conceptual Content Management (CCM) systems were especially developed for the administration of multimedia content. CCM overlaps with traditional content management in separating the description of entities from their presentation. Schmidt and Sehring defined assets which are used to provide descriptions of real-world entities through pairs of media content and conceptual abstractions in a CCM system (see Figure 2).

Insert Figure 2 Here

Figure 2. Dualistic description of entities by assets. An asset describes a real-world entity and is based on a content-concept pair. The content part supports a media and the concept part a model view of the entity. Adapted from (Sehring, 2004).

The terms content and concepts described in the works of Schmidt and Sehring are not the same term definitions as used in this chapter. Schmidt and Sehring defined concept as a description for concrete and abstract entities. Just like a piece of a code can only be used properly if its signature is known to the user. Content descriptions such as pixels of an image have to be paired with a conceptual understanding of the entities' nature, for example, the tuple (Kajsa, Bergqvist, 2.06) represents the athlete Kajsa Bergqvist whose performance is 2.06. The tuple (Snow White and the Seven Dwarfs, Fairy tale) represents the fairy tale "Snow White and the Seven Dwarfs" only if the conceptual model of the defined tuples is clear to the user. Therefore, entity descriptions in general consist of content paired with a conceptual model of the kind of entity it refers to. For such *[content, concept]* pairs the notion of an asset could be used as an atomic union of a perceivable content and a set of abstractly described expressions. However, systems cannot be based on a data model alone. Data models are limited by technical constraints of the target system (a database in most cases). To avoid such technical aspects in the domain model, a conceptual model is required. The asset language specifies such a model. More details of the approach are described in (Sehring and Schmidt, 2004). Variations of CCM systems are presented in (Bossung, Sehring, and Schmidt, 2005; Bossung, Sehring, Skusa, and Schmidt, 2005; Schmidt, Sehring, and Bossung, 2005; Schmidt, Sehring, and Warnke, 2001).

Knowledge Management

In 1986, Wiig defined the term Knowledge Management (KM). In 1997, Wiig claims that the main objectives of a Knowledge Management System (KMS) are: to make the enterprise act as intelligently as possible to secure its viability and overall success, and otherwise realize the best value of its knowledge assets. There are two fundamental aspects to KM: The first aspect involves knowledge being considered

as an asset that is capable of being shared within a wider community. Many early knowledge management projects involved intranet solutions to keep and distribute a form of ``knowledge" among companies. The visibility of the asset model enabled its use to justify significant levels of investment, principally in technology-based solutions. The second aspect is that the balance of explicit and tacit knowledge is not given (Cortada and Woods, 1999). In 2005, De Brun defined KM as follows: KM is based on the idea that an organization's most valuable resource is the knowledge of its people. Therefore, the extent to which an organization performs well will depend, among other things, on how effectively its people can create new knowledge, share this knowledge within the organization, and use that knowledge most effectively.

The Knowledge-Creation Process

In 2003, Nonaka, Toyama, and Byosière proposed a multilayered model of knowledge creation in order to understand how companies create knowledge dynamically. The process of knowledge creation is based on the SECI process, a platform for knowledge creation, called ba, and knowledge assets which are the inputs and outputs of the knowledge-creation process.

SECI Process

Nonaka defines knowledge creation as a spiraling process of interactions between explicit and tacit knowledge. The interactions between the explicit and tacit knowledge lead to the creation of new knowledge. The combination of both categories enabled him to conceptualize four conversion patterns. There are four conversion patterns of knowledge: tacit to tacit: socialization, tacit to explicit: externalization, explicit to explicit: combination, and explicit to tacit: internalization. The process begins with socialization and continuous on with externalization, combination, and internalization. Then the process continues at a higher level after one cycle. Subsequently, the metaphor of a spiral for the knowledge creation is often referred to the SECI model (Nonaka and Takeuchi, 1995). The SECI model describes a dynamic process in which explicit and tacit knowledge are exchanged. The four conversion patterns make it possible to conceptualize knowledge.

Extended SECI Model

In 2001 and 1998, Nonaka and Konno extended the SECI model with four types of *ba* because ba is the foundation for knowledge creation. *Ba* sets binding conditions for the user by limiting the way in which they see the world. The four types correspond to the four stages of the SECI model:

- socialization: originating *ba*
- externalization: dialoguing *ba*
- combination: systemizing *ba*, and
- internalization: exercising *ba*.

Originating *ba* is a place where individuals share feelings, emotions, experiences, and mental models. Dialoguing *ba* is a place where selected people with a specific knowledge interact with other people with a similar specific knowledge. The mental models etc. from the originating *ba* are shared through concepts, articulation, and so on. Systemizing *ba* is a place where the combination phase of new explicit knowledge with existing explicit knowledge is represented. Exercising *ba* is a place where the exercising *ba* facilitates the conversion of explicit knowledge to tacit knowledge.

Knowledge Assets

In the context of KM, knowledge assets are the basis of knowledge creation. Knowledge assets are inputs and outputs of the knowledge-creation process. In 2001, Nonaka divided knowledge assets into four types:

- experiential (tacit knowledge shared through common experiences),
- conceptual (explicit knowledge articulated through images, symbols and language),

- systemic (systematized and packaged explicit knowledge), and
- routine (tacit knowledge embedded in actions and practices).

An experiential asset (E-asset) contains unstructured data which represents tacit knowledge such as an image. A conceptual asset (C-asset) contains ontology-based content descriptions, holistic content descriptions, and conceptual descriptions which presents the E-asset. The tacit knowledge of an E-asset will be made explicit. A systemic asset (S-asset) contains all assets which results from the combination of concept and content descriptions. A routine asset (R-asset) contains all explicit knowledge which is embedded in actions and practices.

Knowledge-Creation Model

The four asset types are the basis for the knowledge acquisition process but not enough to manage knowledge assets. Knowledge assets cannot be managed in the traditional way of management because they are dynamic. In 2003, Dierkes, Antal, Child, and Nonaka presented a new approach how knowledge assets can be managed in a knowledge creating process. The knowledge creating process contains the three main elements: SECI, *ba* and knowledge assets. A company uses existing knowledge assets and creates new knowledge through the SECI process which takes place in *ba*. The newly created asset will be added to the existing knowledge assets of the company. The management of these assets can be done as described in the following: All users can take part in the definition of the knowledge vision by developing, promoting the sharing of knowledge assets, creating, and energizing *ba*. The definition of a knowledge vision supports the realization of a dynamic knowledge management because it gives a direction (vision) where the company is going and how knowledge can be managed over a long-term. Additionally, this definition sets the quality of knowledge.

A Knowledge Management Scenario

There are Information Retrieval (IR) processes which are based on either symbolic content descriptions or on holistic content descriptions. For this reason, users have to be experts into different data representation formats, and query languages in order to find high-quality documents in the semantic web.

The following example shows the problems to find high-quality documents in the semantic web. Assume in the semantic web are multimedia documents which represent athletics news, and there are holistic and symbolic content descriptions for these multimedia documents. Further assume, a sports reporter is interested into documents, and images about ``Kajsa Bergqvist". He knows that Kajsa Bergqvist is an athlete but he does not know that she is a high jump athlete. In this example, the user has the problem to find high-quality documents because he does not know to look for the term ``high jump".

If the sports reporter send the query ``Kajsa Bergqvist" to a search engine which is based on holistic content descriptions, then the query is matched with the holistic content descriptions. The holistic IR process usually delivers the multimedia documents which include the term ``Kajsa Bergqvist".
But if there is a multimedia document which contains an image which represents an pole vault event and some textual descriptions about the pole vault athletics news with the terms ``Kajsa Bergqvist" and ``Yelena Isinbayeva", then this multimedia document as part of the search result is a false positive document. That means the sports reporter receives this document which describes a pole vault event and maybe thinks that Kajsa Bergqvist is a pole vaulter.

Symbolic IR processes can reduce the false positive rate because they tend to high precision. Nowadays, it is possible to extract the term ``Pole Vault" from the image itself automatically (Espinosa et al., 2007). Otherwise it is also possible to extract the term ``High Jump" from an image which represents a high jump event. Indeed, it is known that symbolic IR processes tend to low recall. The automatic symbolic IR process is the modern way to ensure high-quality search results. To this end, content descriptions do not

have to link manually from ``Kajsa Bergqvist" to ``high jump" and users do not have to establish contact with the authors of the multimedia document or photographers of the image.

The systematic combination of both IR processes might lead to high recall and at least maintaining precision. Thus, the sports reporter do not have to be expert into different data representation formats, and query languages and receives high-quality document, in this example, he receives all images about ``Kajsa Bergqvist" and ``High Jump" documents with high precision.

Summary

Managing the wealth of content and content descriptions (feature-based, holistic, and ontology-based) contained in the semantic web and companies, particular data representation formats and processes for managing content descriptions are required. Especially for the representation of multimedia content, Schmidt and Sehring defined assets as management units in the context of conceptual content management (Schmidt and Sehring, 2003). In the context of knowledge management, Nonaka, Toyama, and Byosière defined assets as management units, whereby these assets are called knowledge assets (Nonaka, Toyama, and Byosière, 2003). Assets or knowledge assets represent the content and the knowledge of an entity such as a multimedia object. The management and sharing of knowledge assets are defined by specific processes such as the knowledge-creation process defined by Nonaka et al. (2003). The assets approaches and the processes for representing content, content descriptions and knowledge are the basis for the management and retrieval of high-quality documents. The knowledge management scenario showed that for an automatic information retrieval process it is necessary to formalize the management units and the processes.

FOUNDATIONS FOR ONTOLOGY-BASED AND HOLISTIC REPRESENTATIONS

The section before presented the need to formalize the notions (management units and processes) of conceptual content and knowledge management. Before we can start with the formalization of CCM and KM notions, we need the foundations for the management units and processes which are given in this section.

Foundations for Holistic Content Descriptions

An investigation for holistic content descriptions in this thesis will be based on the latent semantic indexing (LSI) which is described in the following. The aim of the LSI algorithm is to find the main concepts of a document in order to classify a document. The way how this works in general is described in the following.

Latent Semantic Indexing Algorithm

Latent Semantic Indexing (LSI) is an algorithm to indexing and retrieving documents (Manning et al., 2008, pp. 403-419). The terms of a document and the document are represented in a term-document matrix. Before this algorithm is described formally, a short introduction into linear algebra is given and a class of operations from linear algebra, known as matrix decomposition. Then a special form of matrix decomposition is presented which is used to construct a low-rank approximation to the term-document matrix. Then the LSI algorithm is described in the last part of this section.

A brief Introduction into Linear Algebra

Let C be a $M \times N$ matrix, where the $M \times N$ matrix is a term-by-document matrix with non-negative values. Each row corresponds to a unique term in the document corpus and each row corresponds to a document. The value at each row position is the number of the word in the columns' document. The $rank$ of a matrix is the number of linearly independent rows or columns in it. Thus,

$$rank(C) \leq min\{M, N\}.$$

A square matrix is a diagonal matrix ($diag(\sigma_1, \sigma_2 \ \ldots, \sigma_r)$ and 0 otherwise) with the dimension $r \times r$. If all diagonal entries of the diagonal matrix are 1, then this matrix is called the identity matrix I_r of dimension r.
For a square $M \times M$ matrix C and a vector $\vec{x}$, the values of λ satisfying the equation

$$C\vec{x} = \lambda\vec{x}, \vec{x} \neq 0$$

are eigenvalues of C. This equation is equal to the following equation

$$(C - \lambda I_M)\vec{x} = 0, \vec{x} = \vec{x} \neq 0.$$

That implies $(C - \lambda I_M)$ is singular and hence that

$$det(C - \lambda I_M) = 0.$$

Sometimes such eigenvalues are called right eigenvector. The left eigenvector are defined as

$$\vec{y}^T C = \lambda\vec{y}$$

Because of

$$\vec{y}^T C = (C^T \vec{y}\,)^T$$

the left eigenvectors of C are the right eigenvectors of C^T. The properties of eigenvalues and eigenvectors are described which was necessary to set up the central idea of the singular value decomposition.

Matrix Decomposition

In the linear algebra, matrix decomposition is a factorization of a matrix into some canonical form. In this section it is presented how a square matrix can be factored into the product of matrices derived from its eigenvectors where the matrices are square. Let $\lambda_1, \lambda_2, \ldots, \lambda_n$ be the eigenvalues of a square real-valued $M \times M$ matrix S, let $x_1, x_2, \ldots, x_n$ be a set of corresponding eigenvectors, let U denotes the $r \times r$ matrix whose jth columns is x_j, and let Λ is denoted by the $r \times r$ diagonal matrix with the λ_j on the diagonal, whereby the diagonal entries are in decreasing order. Then

$$SU = U\Lambda.$$

Giving the decomposition of S into a similarity transformation involving U and Λ, then

$$S = U\Lambda \mathrm{U}^{-1}$$

The fact that this decomposition is known as the eigendecomposition theorem. If T is a nonsingular matrix, then

$$S = TB\mathrm{T}^{-1}$$

is known as a similarity transformation and S and B are said to be similar. If $Cx = \lambda x$, and $x = Ty$, then $By = \lambda y$. In other words, a similarity transformation preserves eigenvalues. The eigenvalue decomposition is an attempt to find a similarity transformation to diagonal form.

Singular-Value Decomposition

Given C, let U be the $M \times M$ matrix whose columns are the orthogonal eigenvectors of CC^T, and V be the $N \times N$ matrix whose columns are the orthogonal eigenvectors of $C^T C$, where C^T is the transpose matrix of C. An orthogonal matrix C has the characteristics $C^T C = I$ where I is the identity matrix. The Singular-Value Decomposition (SVD) for C is defined as

$$C = U\Sigma V^T$$

where U is an orthogonal matrix ($U^T U = I_m$), V is an orthogonal matrix ($V^T V = I_n$), and Σ is a diagonal matrix ($\Sigma = diag(\sigma_1, \sigma_2, \dots, \sigma_n)$ and 0 otherwise). The eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_r$ of CC^T are the same eigenvalues of $C^T C$, and for $1 \leq i \leq r$, let the singular value of C and $\sigma_i = \sqrt{\lambda_i}$ with $\lambda_i \geq \lambda_{i+1}$. Then the $M \times N$ matrix Σ is composed by setting $\Sigma_{ii} = \sigma_i$ for $1 \leq i \leq r$, and 0 otherwise. By multiplying the Equation above by its transposed version, we have

$$CC^T = U\Sigma V^T V \Sigma U^T = U\Sigma^2 \mathrm{U}^{\mathrm{T}}.$$

The SVD for a rank-k-approximation of low error is defined as

$$C_k = U\Sigma_{\mathrm{k}} \mathrm{V}^{\mathrm{T}}.$$

The rank of C_k is at most k. This follows from the fact that Σ_{k} has a most k non-zero values. In LSI the SVD is used to construct a low-rank approximation C_k to the term-document matrix C. The value k is smaller than the original rank of C.

The matrix C_k is the best rank k approximation to the original matrix C because the distance between the two matrices is minimized. The Frobenius norm of the matrix difference $X = C - C_k$ is defines as

$$\|x\|_F = \sqrt{\sum_{i=1}^{M} \sum_{j=1}^{N} x_{ij}^2}\ .$$

If C has the rank r, then $C_r = C$ and the Frobenius norm is zero. If k is far smaller than r , then C_k is referred as a low-rank approximation. The minimized difference between the two matrices C and C_k is

$$\min_{Z|rank\,(Z)=k} \|C - Z\|_F \ \|C - C_k\|_F = \sigma_{k+1}.$$

Consider that the singular values are in decreasing order. This equation delivers that C_k is the best rank-k approximation to C, incurring an error equal to σ_{k+1}.These facts help us to generate a rank-k approximation of low error.

It is conventional to represent Σ as an $r \times r$ matrix with singular values on the diagonal and zero otherwise. The rightmost $M - r$ columns of U correspond to the omitted rows of Σ, and the $N - r$ columns of V are omitted since they correspond in V^T. This form of the SVD is known as the truncated SVD and is formalized as

$$C_k = U_k \Sigma_{\mathrm{k}} \mathrm{V}_{\mathrm{k}}^{\mathrm{T}}.$$

Insert Figure 3 Here

Figure 3: Diagram of the truncated SVD.

Figure 3 illustrates the truncated SVD. The usage of the truncated SVD has the advantage to eliminate a lot of redundant columns of zeros in U and V.

Latent Semantic Indexing

The low-rank approximation to C yields a new representation for each document in the semantic web. Queries can also be represented using the low-rank approximation. The process of computing query document similarity scores is known as latent semantic indexing. In the latent semantic indexing process the SVD is used to construct a low-rank approximation C_k to the term-document matrix. The value k is generally chosen in the low hundreds because k is far smaller than the original rank of C. Thus, each row and column is mapped into a k-dimensional space:

$$C_k = U\Sigma_k V^T$$
$$U^T C = U^T U\Sigma_k V^T$$
$$\Sigma_k^{-1} U^T C = V^T.$$

The query vector $\vec{q}$, is mapped into its representation in the LSI space by the transformation as stated above. Each column of V^T is represented as the query vector $\vec{q}$, thus, the following equation results

$$\vec{q}_k = \sigma_k^{-1}\, U_k^T \vec{q}.$$

The cosine similarity as described in the next section computes the similarity between a query and a document.

Cosine Similarity

The cosine similarity computes the similarity between a query and a document, between two documents, or between two terms. The cosine similarity between two documents d_1 and d_2 is defined as:

$$sim(d_1, d_2) = \frac{\vec{V}(d_1)\vec{V}(d_2)}{|\vec{V}(d_1)||\vec{V}(d_2)|}$$

where $\vec{V}(d)$ derived from document *d*, the numerator represents the dot product of the vectors $\vec{V}(d_1)\vec{V}(d_2)$, while the denominator is the product of their Euclidean lengths.

Term Frequency - Inverse Document Frequency

Term frequency (tf) and inverse document frequency (idf) are weight function pairs which support the improvement of LSA results. This weight is a statistical measure. It is used to find out the importance of a word in a document.

The *term frequency* (tf) counts the number of times that a term t_i appears in a corpus. If a word has *N* terms and n_j is the number if occurrences of the considered term t_i in the document, tf is defined formally as

$$tf(t_j) = \frac{n_j}{N}.$$

The *inverse document frequency* counts the number of words in the document that contain t_j. The total number of documents in the corpus is called d_j, and *W* is the number of documents where the term t_i appears. The idf is defined as

$$idf(t_j) = \log\frac{d_j}{W}.$$

Altogether, the $tf - idf$ *weight function* of term t_j is defined as

$$tf - idf(t_j) = tf(t_j) \times idf(t_j) = \frac{n_j}{N} \log\frac{d_j}{W}.$$

The $tf - idf$ *weight function* is often used with the one widely used method for defining similarity, the cosine similarity.

Foundation for Ontology-based Content Descriptions

For representing ontology-based content descriptions, in general, formal languages are used (Baader et al., 2003). In this thesis, we use descriptions logics (DLs) which are described in the following.

An Outline for Ontology-based Annotations and Multimedia Interpretation

Ontology-based annotations represent the content of documents, images, and videos. Nowadays, information extraction tools are able to annotate multimedia documents. That means they identify objects

and relations among them within multimedia documents. In 2009, Paliouras presents some annotation tools for video, and image, text, and HTML pages. Tools which annotate objects and relations are also called low-level Information Extraction (IE) tools.

VIA (Video and Image Annotation tool), and BTAT (BOEMIE Text Annotation Tool) are low-level IE tools. VIA is used for manual video and image annotations, whereby image annotations are done at low-level and at high-level. High-level annotation means that image regions and complete images are linked with concepts. Low-level annotation means that visual descriptors are extracted via annotated region, and associated with the corresponding concept. BTAT supports the annotations of named entities and the relations between those entities.

BIWS (BOEMIE Interpretation Web Services) is a high-level extraction tool which is offered by a semantic interpretation engine. The non-standard retrieval inference service Abox abduction is the basis of BIWS. BIWS generates interpretation data which are based on ontology-based annotations.

The ontology-based annotation process, also called analysis process, is a preliminary processing of multimedia interpretation. Both processes are outlined in the following by example in order to get an idea for describing multimedia documents via ontologies.

Insert Figure 4 Here

Figure 1: Image. The low-level IE tool extract the objects horizontal bar bar_1, body $body_1$, and face $face_1$. Original image adapted from (IAAF, 2009).

Ontology-based Annotations

Figure 4 shows an image from a sample web page taken from the website of the International Association of Athletics Federations (IAAF, 2009). The low-level IE tool VIA extract the objects image $image_1$, horizontal bar bar_1, body $body_1$, and face $face_1$, and the relations between these objects *isAdjacent*($body_1$, $face_1$), and *isAdjacent*($body_1$, bar_1).

Multimedia Interpretation

The annotated media objects are symbolic content descriptions, also called analysis results. The high-level extraction tool BIWS service generates interpretation data based on the annotated objects and relations given above. In this case, the BIWS generate the objects *person*, *high jump*, and *high jumper* as additional content descriptions.

An Introduction to Description Logics

Description Logics (DLs) (Baader et al., 2003) correspond to a large fragment of standard ontology languages such as OWL. They are used to model a certain application domain. It based on the idea of *atomic descriptions* for *concepts* and *roles* . From the first-order logic point of view, concepts are unary and roles are binary predicates to represent relations between concepts. For instance, *Person*, *Athlete* and *HighJump* are *atomic concept descriptions* and *hasParticipant* is an atomic role description in the *athletics* domain. And *Narrator* and *Title* are *atomic concept descriptions* and *hasAuthor* is an atomic role description in the *fairy_tales* domain.

There are some variations of DLs. In this work an introduction into *ALCQ* (Attributive Language with full *Complement* and *Qualified* number restrictions) is given. The letters are the operators which are used and the expressiveness is encoded in the label for logic. The prototypical Attribute concept Language with

Complements (*ALC*) is the basis of many more expressive DL and the letter *Q* means that *qualified* cardinality restrictions will be considered.

Syntax and Semantics

In *ALCQ* descriptions for *complex concepts C* or *D* can be inductively built using concept constructors shown in the following where *A* is an atomic concept description and *R* is a role description.

Syntax	Constructor
A	atomic concept
$C \sqcap D$	conjunction
$C \sqcup D$	disjunction
$-C$	negation
$\exists R.C$	existential restriction
$\forall R.C$	value restriction
$\exists_{\leq n} R.C$	qualified minimum restriction
$\exists_{\geq n} R.C$	qualified maximum restriction

For example, $JumpingEvent \sqcap \exists_{\leq 1} hasParticipant.Athlet$ is a complex concept description in the *athletics* domain. The highest concept in any given hierarchy is called *top concept* which contains all concept names. All other concepts are subsumed by the top concept. Analogue, the lowest concept in any given hierarchy is called *bottom concept* denoted $\perp$. This concept is equal to the empty set denoted {}. The semantics of concept and role descriptions are defined as *interpretations* I that consist of a non-empty set Δ_I, the domain, and an interpretation function $.^I$, which assigns to every atomic concept description *A* a set $A^I \subseteq \Delta_I$ and to every (atomic) role *R* a set $R^I \subseteq \Delta^I \times \Delta^I$. For complex concept descriptions, the interpretation function is extended as follows:

$$(C \sqcap D)^I = C^I \cap D^I$$
$$(C \sqcup D)^I = C^I \cup D^I$$
$$(-C)^I = \Delta^I \backslash C^I$$
$$(\exists R.C)^I = \{x | \exists y.(x,y) \in R^I and\ y \in C^I\}$$
$$(\forall R.C)^I = \{x | \forall y.if(x,y) \in R^I then\ y \in C^I\}$$
$$(\exists_{\leq n} R.C)^I = \{x | \#\{y | (x,y) \in R^I and\ y \in C^I\} \leq n\}$$
$$(\exists_{\geq n} R.C)^I = \{x | \#\{y | (x,y) \in R^I and\ y \in C^I\} \geq n\}$$

Model

The semantics of description logics is based on the notion of satisfiability. An interpretation $I = (\Delta^I, .^I)$ *satisfies* a concept description C if $C^I = \emptyset$. In this case, *I* is called a *model* for C.

Tbox

Concept definitions and inclusion axioms naturally form a hierarchy of concepts, for example *HighJump*$\sqsubseteq$*SportsTrial*, *ChildrensAndHouseholdTale*$\sqsubseteq$*FairyTale* is called *terminology* or *Tbox*.

GCIs

The elements of a Tbox are called *generalized concept inclusions* (*GCIs*).

Interpretation

An interpretation I *satisfies* a GCI $C \sqsubseteq D$ if $C^I \subseteq D^I$. An interpretation is a *model* of a Tbox if it satisfies all GCIs in the TBox. A concept description C *is subsumed by* a concept description D w.r.t. a Tbox if the GCI $C \sqsubseteq D$ is satisfied in all models of the Tbox. In this case, we also say that D *subsumes* C.

Abox

An *Abox* represents *assertional knowledge* and is a set of *assertions* of the form $i: C$ or $(i, j): R$ where C is a concept description, R is a role description, and $i; j$ are individuals. A concept assertion $i: C$ is satisfied w.r.t. a Tbox T if for all models I of T it holds that $i^I \in C^I$. A role assertion $(i, j): R$ is satisfied w.r.t. a Tbox T if $(i^I, j^I) \in R$ for all models I of T. An interpretation satisfying all assertions in an Abox A is called a model for A. An Abox A is called *consistent* if such a model exists, it is called *inconsistent* otherwise.

Ontology

An *ontology* Σ is a tuple (T, A) with Tbox T and Abox A. Let □ be conceptor role assertion. An ontology Σ *entails* an assertion α (α follows from Σ), denoted as $\Sigma \models \alpha$ if for all models I of □ it holds that I satisfies α. Let A be an Abox. An ontology Σ *entails* an Abox, denoted as $\Sigma \models$ A, if for all $\alpha \in$ A it holds that $\Sigma \models \alpha$. In the following sections we slightly misuse notation and assume that $(T, A) \cup A'$ means $(T, A \cup A')$.

Decision Problems and their Reductions

A decision problem is a question with a true or false answer, depending on the values of some input parameters. The definitions given in the previous subsection can be paraphrased as decision problems:

- If a model for a concept description exists check the *concept satisfiability* problem.
- If a model for the Tbox exists check the Tbox satisfiability problem.
- If $C \sqsubseteq D$ holds in all models of the Tbox check the *concept subsumption* problem.

Satisfiability checks of descriptions and consistency checks of sets of assertions are useful to determine whether a knowledge base, which comprises the Tbox and the Abox, is meaningful at all. An overview about the Tbox satisfiability problems is given in the following:

- The *Abox consistency problem* for an Abox A (w.r.t. a Tbox) is the problem of determining whether there exists a model of A (that is also a model of the Tbox).
- Another problem is to test whether an individual i is an instance of a concept description C w.r.t. a Tbox and an Abox (*instance test* or *instance problem*: $\models i: C$).
- The *instance retrieval* problem w.r.t. a query to the query concept *C* and the ontology Σ is to find all individuals i mentioned in the assertions of an Abox such that i is an instance of C.
- For roles and pairs of individuals, similar definitions can be given.

Following problems could be reduced by solving other problems:

- In theory, the *retrieval problem* can be reduced to several instance problems. In order to solve the instance problem for an individual i and a concept description C one can check if the Abox $\{i: (\neg C)\}$ is inconsistent (Baader and Nutt, 2003).
- The *satisfiability problem* for a concept description C can be reduced to the consistency problem for the Abox $\{i: C\}$.
- In theory, all problems introduced above can be reduced to the Abox consistency problem.
- In practical systems, specific optimization techniques are used to decide a certain decision problem.

Sequences, Variable Substitutions, Transformations

Some additional definitions for the interpretation algorithm are required and are given in this subsection. A *variable* is a name of the form ?NarratorName or ?PersonName where name is a string of characters from $\{a \ldots z\}$ and $\{A \ldots Z\}$.

Let V be a set of variables, and let $X, Y_1, \ldots, Y_n$ be sequences $\langle ?1 \ldots ?n \rangle$ of variables from V. $\underline{Z}$ denotes a sequence of individuals. Sequences of length one or two are only considered, if not indicated otherwise, and assume that $(\langle ?X \rangle)$ is to be read as (X) and $(\langle ?X, Y \rangle)$ is to be read as (X, Y) etc. Furthermore, it is assumed that sequences are automatically flattened. A function *as_set* turns a sequence into a set in the obvious way. A *variable substitution* has the form

$$\sigma = [X \leftarrow i, Y \leftarrow j, \ldots]$$

and is a mapping from variables to individuals. The application of a variable substitution σ to a sequence of variables $\langle ?X \rangle$ or $\langle X, Y \rangle$ is defined as

$$\langle \sigma(?X) \rangle := r$$
$$\langle \sigma(?X), \sigma(Y) \rangle$$

where $\sigma(X) = i$ and $\sigma(Y) = j$. In this case, a sequence of individuals is defined. If a substitution is applied to a variable X for which there exists no mapping $X \leftarrow k$ in σ then the result is undefined. A variable for which all required mappings are defined is called *admissible* (w.r.t. the context).

Grounded Conjunctive Queries

In *standard* conjunctive queries, variables (in the head and in query atoms in the body) are bound to (possibly anonymous) domain objects. A system supporting (unions of) standard conjunctive queries is QuOnto. In so-called *grounded* conjunctive queries $C(X)$, $R(X, Y)$ or $X = Y$ are true if, given some bindings α for mapping from variables to individuals mentioned in the Abox A, it holds that $(T, A) \models \alpha(X) : C$, $(T, A) \models (\alpha(X), \alpha(Y)) : R$, or $(T, A) \models \alpha(X) = \alpha(Y)$, respectively. In grounded conjunctive queries the standard semantics can be obtained for so-called tree-shaped queries by using corresponding existential restrictions in query atoms (Peraldi et al., 2007). The definition of grounded conjunctive queries is given in the following:

Let $\underline{X}, \underline{Y_1}, \ldots, \underline{Y_n}$ be sequences of variables, and let $Q_1, \ldots, Q_n$ denote atomic concept or role descriptions. A query definition has the syntax:

$$\{(\underline{X}) | Q_1(\underline{Y_1}), \ldots, Q_n(\underline{Y_n})\}.$$

The sequence $\underline{X}$ may be of arbitrary length but all variables mentioned in $\underline{X}$ must also appear in at least one of the $\underline{Y_1}, \ldots, \underline{Y_n}$: $as\_set(X) \subseteq as\_set(\underline{Y_1}) \cup \ldots \cup as\_set(\underline{Y_n})$.

$Q_1(\underline{Y_1}), \ldots, Q_n\left(\underline{Y_n}\right)$ defines a conjunction of *query atoms* $Q_i(\underline{Y_i})$. The list of variables to the left of the sign | is called the *head* and the atoms to the right of are called the *body* (query). The variables in the head are called distinguished variables. They define the query result. The variables that appear only in the body are called non-distinguished variables and are existentially quantified.
Answering a query with respect to an ontology Σ means finding admissible variable substitutions □ such that

$$\Sigma \models \{\left(\sigma\left(\underline{Y_1}\right)\right) : Q_1, \ldots, \left(\sigma\left(\underline{Y_n}\right)\right) : Q_n\}.$$

A variable substitution

$$\sigma = [X \leftarrow i, Y \leftarrow j, \ldots]$$

introduces *bindings* $i, j, \ldots$ for variables $X, Y, \ldots$. Given all possible variable substitutions σ, the *result* of a query is defined as

$$\{(\sigma\ (X))\}.$$

Note that the variable substitution σ is applied before checking whether

$$\Sigma \models \{\left(\sigma\left(\underline{Y_1}\right)\right): Q_1, \ldots, \left(\sigma\left(\underline{Y_n}\right)\right): Q_n\},$$

i.e., the query is *grounded* first.
For a query $\{(?\,x) \mid \text{Person}(?\ x), \text{hasParticipant}(?\,y, ?\,x)\}$ and the Abox
ΓFehler! Textmarke nicht definiert.$\{ind_1: HighJump,\ ind_2: Person,\ (ind_1, ind_2): hasParticipant\}$,
the substitution $[?\,x \leftarrow ind_2, ?\,y \leftarrow ind_1]$ allows for answering the query, and defines bindings for $?\,x$.
And for a query $\{?\,x | Narrator(?\,x), hasAutor(?\,y, ?\,x)\}$ the substitution $[?\,x \leftarrow ind_2, ?\,y \leftarrow ind_1]$ allows for answering the query, and defines bindings for $?\,x$.
A *boolean* query is a query with $\underline{X}$ being of length zero. If for a boolean query there exists a variable substitution σ such that

$$\Sigma \models \{\left(\sigma\left(\underline{Y_1}\right)\right): Q_1, \ldots, \left(\sigma\left(\underline{Y_n}\right)\right): Q_n\}$$

holds. The query is answered with *true*, otherwise the answer is *false*. Then the query atoms will be converted into Abox assertions with the function $transform$. The function *transform* applied to a set of query atoms is defined as

$$\{transform(\gamma_1, \sigma), \ldots transform(\gamma_n, \sigma)\},$$

where $transform(P(X), \sigma) := (\sigma\ (X)): P$.

Rules

Rules are used to derive new Abox assertions. A rule *r* has the following form

$$P(X) \leftarrow Q_1\left(\underline{Y_1}\right), \ldots, Q_n(\underline{Y_n})$$

where $P, Q_1, \ldots, Q_n$ denote atomic concept or role descriptions with the additional restriction that $as\_set(X) \subseteq as\_set(\underline{Y_1}) \cup \ldots \cup as\_set(\underline{Y_n})$. A rule r is *applied* to an Abox A. The function application

$$apply(\Sigma, P(X) \leftarrow Q_1\left(\underline{Y_1}\right), \ldots, Q_n\left(\underline{Y_n}\right), A)$$

returns a set of Abox assertions $\{(\sigma\ (X)): P\}$ if there exists an admissible variable substitution σ such that the answer to the query

$$\{()|Q_1\left(\underline{Y_1}\right), \ldots, Q_n(\underline{Y_n})\}$$

is *true* with respect to Σ ∪ A. If no such σ can be found, the result of the call to $apply(\Sigma, r, A)$ is the empty set. The application of a set of rules $R = \{r_1, \ldots, r_n\}$ to an Abox is defined as follows:

$$apply(\Sigma, \mathrm{R}, \mathrm{A}) = \bigcup_{r \in R} apply(\Sigma, \mathrm{r}, \mathrm{A})$$

If $apply(\Sigma, R, A) \cup A = A$, then the result of $forward\_chain(\Sigma, \mathrm{R}, \mathrm{A})$ is ∅. Otherwise the result is $apply(\Sigma, \mathrm{R}, \mathrm{A}) \cup forward\_chain(\Sigma, \mathrm{R}, \mathrm{A})$.

Computing Explanations via Abduction

Abduction can be considered as a new type of non-standard retrieval inference service. In this view, observations (or part of them) are utilized to constitute queries that have to be answered. Contrary to

existing retrieval inference services, answers to a given query cannot be found by simply exploiting the knowledge base. In fact, the abductive retrieval inference service has the task of acquiring what should be added to the knowledge base in order to positively answer a query.
More formally, for a given set of Abox assertions Γ (in form of a query) and a knowledge base $\mathcal{K} = (T, A)$, the abductive retrieval inference service aims to derive all sets of Abox assertions (explanations Δ) such that

$$\mathcal{K} \cup \Sigma \cup \Delta \models \Gamma$$

and the following conditions are satisfied:

- Σ ∪ Δ is satisfiable, and
- Δ is a minimal explanation for Γ, i.e., there exists no other explanation in the solution set that is not equivalent to Δ and it holds that $\Sigma \cup \Delta' \models \Gamma$.

The high-level interpretation techniques find abstract concepts (explanations Δ) in the background knowledge as aggregate concepts with constraints among its parts. In the following it is described how such abstract concept (explanations Δ) will be computed via abduction. A set of rules R is assumed. The definition of these rules is given above. The non-deterministic function *compute_explanation* is defined as follows:

$$compute\_explanation(\Sigma, \mathrm{R}, \mathrm{A}, (\underline{Z}): \mathrm{P}) = \mathrm{transform}(\Phi, \sigma)$$

if there exists a rule

$$r = \mathrm{P}(\underline{X}) \leftarrow Q_1\left(\underline{Y_1}\right), \ldots, Q_n\left(\underline{Y_n}\right) \epsilon R$$

such that a set of query atoms Φ and an admissible variable substitution σ with $\sigma\ (X) = Z$ can be found, and the query

$$Q{:=}\{()|expand(P(\underline{X}),r,R) \setminus \Phi\ \}$$

is answered with *true*. If no such rule r exists in R it holds that

$$compute\_explanation(\Sigma, \mathrm{R}, \mathrm{A}, (\underline{Z}): \mathrm{P}) = \emptyset.$$

The goal of the function *compute_explanation* is to determine what must be added (Φ) such that an entailment

$$\mathcal{K} \cup A \cup \Phi \models (\underline{Z}): P$$

holds. Hence, abductive reasoning is used for *compute_explanation*. The set of query atoms Φ defines what must be hypothesized in order to answer the query Q with *true* such that

$$\Phi \in expand(P(\underline{X}), r, R)$$

holds. The definition of *compute_explanation* is non-deterministic due to several possible choices for Φ. The function application

$$expand(\mathrm{P}(\underline{X}), \mathrm{P}(\underline{X}) \leftarrow Q_1\left(\underline{Y_1}\right), \ldots, Q_n\left(\underline{Y_n}\right), R)$$

is also defined in a non-deterministic way as

$$expand'(Q_1\left(\underline{Y_1}\right), R) \cup \ldots \cup expand'(Q_n\left(\underline{Y_n}\right), R)$$

with $expand'(P(\underline{X}), R)$ being $expand(P(\underline{X}), r, R)$ if there exist a rule $r = P(\underline{X}) \leftarrow \ldots \in R$ and $\langle P(\underline{X}) \rangle$ otherwise. We say the set of rules is backward-chained, and since there might be multiple rules in R, backward-chaining is non-deterministic.

Interpreting Aboxes in Terms of Rules

Interpretation of Abox assertions w.r.t. a set of rules is not the interpretation of concept descriptions. Interpretation of Abox assertions are the rules of a high-level explanation such that the Abox assertions are entailed. The interpretation of an Abox is again an Abox. For instance, the output Abox might represent results of a content interpretation process (see below for an example).

Let Γ be an Abox whose assertions are to be interpreted. The goal of the interpretation process is to use a set of rules *R* to derive explanations for elements in Γ. The definition of *requires_fiat* depends on the application context and has the following definition:

$$requires\_fiat((\underline{X}): \mathrm{P}) = \text{true iff } \mathrm{P} \epsilon \{\text{near}, \text{adjacent\_to}, \ldots\})$$

The interpretation algorithm implemented by the interpretation engine works on a set of (possible) interpretations I, for example, a set of Aboxes. Initially, I ⇐ {Γ}, for example , at this stage, the interpretation is just the input Abox Γ.[1] The function *interpret* is applied to an Abox Γ and applies a strategy function Ω in order to decide which assertion to interpret, and uses a termination function Ξ in order to check whether to terminate due to resource constraints. The functions Ω for the interpretation strategy and Ξ for the termination condition are used as an oracle and must be defined in an application-specific way. The function $interpretation\_step(\Sigma, \mathrm{R}, \mathrm{S}, \mathrm{A}, \alpha)$ is defined as

$$\bigcup_{\Delta \,\epsilon\, compute\_all\_explanations\ (\Sigma,\mathrm{R},\mathrm{S},\mathrm{A},\alpha)} \{\Delta \cup \mathrm{A} \cup forward\_chain(\Sigma, \mathrm{R}, \Delta \cup \mathrm{A})\}.$$

The function $compute\_all\_explantions(\Sigma, \mathrm{R}, \mathrm{S}, \mathrm{A}, \alpha)$ is defined as

$$maximize(\{\Delta | \Delta = compute\_explantions(\Sigma, \mathrm{R}, \mathrm{A}, \alpha), \mathrm{S}).$$

There are restrictions on the choice of the □ 's returned by*interpret*. In particular, a preference score of each explanation has the formula $S(\Delta) \coloneqq S_i(\Delta) - S_h(\Delta)$ where S_i and S_h are defined as follows:

$$S_i \coloneqq |\{i | i \in inds(\Delta) \text{ and } i \in inds(\Xi)\}|$$
$$S_h \coloneqq |\{i | i \in inds(\Delta) \text{ and } i \in newInds\}|.$$

The set $newInds$ contains all individuals that are hypothesized during the generation of an explanation (new individuals). The function $inds$ returns the set of all individuals found in a given Abox or a set. The preference score reflects the two criteria proposed by Thagard for selecting explanations, namely simplicity and consilience. That means, the preference score reflects the assumption that an explanation is preferable over others if it is more consilient and simpler. Consequently, an explanation with the highest score is preferred over others.

Abox Fusion

The *fusion algorithm* embedded in an interpretation engine works initially on an empty set of interpretations I. The function *fusion* computed the differences between two Aboxes w.r.t. the Tbox and the individual names. In this work, two Aboxes must contain different media types. The result of the

[1] ⇐ denotes the assignment operator

operation *compute_abox_differences* is a new Abox. Subsequently, the interpretation algorithm starts and computes a new Abox which includes fused assertions of two different media types.

The fusion algorithm is defined as follows:

$fusion(\Omega, \Xi, \Sigma, R, S, A, A'')$:
$I = \emptyset$
$\text{Candidates}_{A,A'} := \text{find\_same\_as\_candidates}(\Sigma, A, A')$
$I' \Leftarrow \text{interpret}(\Omega, \Xi, \Sigma, R_{specific}, S, \text{Candidates}_{A,A'})$
if(I' *is consitent*)
  $I := I \cup I'$
end if
return I

where Ω is a strategy function, Ξ is a termination function, Σ is the knowledge base, R is a set of rules $R = r_1, \ldots r_n$, S is a preference score, and A and A' are just the input Aboxes. $\text{Candidates}_{A,A'}$ is the Abox which contains the "same-as" (similar) candidates from the Aboxes A and A'. $R_{specific}$ is another set of rules, for example document specific rules. In the document specific approach is for example image, caption, or text. These specific rules contain the "same-as" relation which is necessary in order to fuse two candidates. Consider that the functions Ω and Ξ must be defined in an application-specific way.

COMBINING HOLISTIC AND ONTOLOGY-BASED CONTENT DESCRIPTIONS

The previous sections presented the CCM and KM characteristics which motivate the realization of semantic assets, and the foundations for the formalization of semantic assets. This section has the aim to formalize semantic assets by combining holistic and symbolic content descriptions in a knowledge management environment.

Semantic Assets

Semantic assets are management units in a KM area and will be generated in an automatic process. Semantic assets are represented as content-concept pairs. The content part is represented by feature-based metadata,
holistic content descriptions, and symbolic content descriptions. The concept part is the data model used w.r.t. the data representation formats. Consider that in this work the terms content and concept differ from the ones defined by Schmidt and Sehring in 2003.

Insert Figure 5 Here

Figure 5. Semantic asset representation. An asset is represented as a content-concept pair. Content is represented by feature-based metadata, holistic content descriptions, and ontology-based content descriptions. The concept part is the data model used w.r.t. the data representation formats.

Creation and Query Process of Holistic and Ontology-based Content Descriptions

This section describes the creation and querying processes of holistic content descriptions by an example. Let us assume, there are eight multimedia documents d_1 untild_8. The first five documents are from the *athletics* domain and the other three documents are from the *fairyTale* domain. LSI begins to generate a term-document matrix which is presented in **Fehler! Verweisquelle konnte nicht gefunden werden.**.

Insert Table 1 Here

Table 1. Term-document-matrix in the athletics and fairyTale domain.

The SVD for the term-document-Matrix C is defined as

$$C = U\Sigma V^T$$

where U is the term vector and V is the document vector, and Σ contains the singular values. In this case, the term-document-matrix *C* contains the values from **Fehler! Verweisquelle konnte nicht gefunden werden.**.

The SVD values of *C* are

$$C = \begin{pmatrix} 0.00 & 0.36 \\ 0.00 & 0.02 \\ 0.00 & 0.02 \\ 0.00 & 0.04 \\ 0.00 & 0.00 \\ 0.00 & 0.93 \\ 1.00 & 0.00 \\ 0.07 & 0.00 \\ 0.03 & 0.00 \\ 0.02 & 0.00 \\ 0.01 & 0.00 \\ 0.01 & 0.00 \end{pmatrix} \begin{pmatrix} 156.54 & 0 \\ 0 & 130.51 \end{pmatrix} \begin{pmatrix} 0.00 & 0.74 \\ 0.00 & 0.13 \\ 0.00 & 0.28 \\ 0.00 & 0.59 \\ 0.00 & 0.07 \\ 0.03 & 0.00 \\ 1.00 & 0.00 \\ 0.00 & 0.00 \end{pmatrix}^T$$

This is the result of the latent semantic indexing algorithm for the eight documents. The result represents that the dimension of the matrix is reduced from 12 x 8 to 2 x 8. That means the LSI algorithm has computed two main concepts. The first five documents were grouped together, and the other three documents were grouped together.

The search component of latent semantic indexing process is described next. Queries are computed by taking the centroid of the term vectors corresponding to the terms in the query. For example, the query q "athlete HighJumpCompetition" contains the terms "athlete" and "HighJumpCompetition". The centroid is computed point wise by adding the values in each dimension:

$$q = (\,0.00 \quad 0.38).$$

The term scores vs. the query are:

Document d_1	=	37.15
Document d_2	=	6.59
Document d_3	=	14.04
Document d_4	=	29.53
Document d_5	=	3.29
Document d_6	=	0.00
Document d_7	=	0.00
Document d_8	=	0.00

This result shows that the documents d_1, d_2, d_3, d_4, and d_5 contain the query terms. The documents from the *fairyTaile* domain do not contain the terms as expected.

The LSI example shows that the results have a high recall. Using LSI the recall is higher than in other term-matching techniques (Mugo, 2010; Velinska, 2010) whereby the precision is quite low. But precision is quite good in all systems at low recall. The largest benefit of the LSI approach is high recall (Deerwester, Dumais, Furnas, Landauer, and Harshman, 1990).

Creation and Query Process of Ontology-based Content Descriptions

There are three creation processes of ontology-based content descriptions: analysis process, interpretation process, and fusion process. The creation process of ontology-based content descriptions is illustrated in Figure 2.

Insert Figure 6 Here

Figure 3. Creation process of ontology-based content descriptions.

The multimedia repository in Figure 6 contains multimedia documents including all media objects such as images, texts, audio, and video clips. The low-level information extraction (IE) tool generates for each media object unstructured metadata in the analysis process. The high-level interpretation tool generates interpretation and fusion metadata using Abox abduction and Abox fusion. The low-level IE and high-level interpretation tools use the ontology repository to generate unstructured metadata for each media object and multimedia document. The ontology repository consists of several ontologies in OWL and rules. All generated datasets will be stored in a semantic asset repository. This repository contains the semantic assets. The Asset Manager supervises the processes of the data generation and storage, and manages semantic assets.

An Example for Multimedia Interpretation and Fusion as Abduction

The ontology-based content description of a multimedia document will be generated if the document runs through three processes: analysis, interpretation, and fusion process. This section shows how ontology-based content description could help to increase the precision. The multimedia document in Figure 7 contains an image and a caption text.

Insert Figure 7 Here

Figure 7. A sample multimedia document. It contains two multimedia objects: image and caption. Adapted from (IAAF, 2009).

The textual information in the caption supplements the visual information in the image by providing additional information such as the athlete's name, performance, and the city. It is assumed that the multimedia document in Figure 7 has successfully been partitioned into image and caption parts before the analysis process can start.

Analysis Process

Low-level IE tools such as OntoMat-Annotizer (Handschuh and Staab, 2002; Handschuh, Staab, and Volz, 2003) annotate multimedia objects and deliver analysis Aboxes as a result. Figure 8 and Figure 9 are analysis Aboxes for the image and caption parts in Figure 7.

Insert Figure 8 Here

Figure 8. The analysis Abox AAboxImage. This Abox represents the results of the image analysis for the image in Figure 7.

The instances $image_1, face_1, bar_1,$ and $body_1$ are instances of the concepts *Image*, *PersonFace*, *HorizontalBar*, and *PersonBody*.

Insert Figure 9 Here

Figure 9. The analysis Abox AAboxCaption. This Abox represents the results of caption analysis for the caption text in Figure 7.

The instances $caption_1, pname_1,$ and $perf_1$ are instances of the concepts *Caption*, *PersonName*, and *Performance*.

Interpretation Process

In order to obtain interpretation results for the image and caption text, a client has to use the particular interpretation service (i.e., BIWS) of the semantic interpretation engine. BIWS is based on the abduction approach. We remember, the *abduction* process is used to find explanations (causes) for observations (effects). Abduction is formalized as

$$\Sigma \cup \Delta \models \Gamma$$

where the background knowledge (Σ) and the observations (Γ) are given, and the explanations (Δ) are to be computed. DLs are used as the underlying knowledge representation formalism in these processes. The background knowledge $\Sigma = (T, A)$ is a knowledge base (KB) that consists of a Tbox T, DL-safe rules, and an Abox A. DL-safe rules (Studer, Motik, and Sattler, 2005) are a combination of OWL DL and rules. DL-safe rules require that each variable that appears in the head of a rule must occur in the body of the same rule. In this example, the Tbox is presented in Figure 10. The Tbox contains intentional knowledge in the form of a terminology and DL-safe rules.

Insert Figure 10 Here

Figure 4. An excerpt of Σ consisting of a Tbox T and DL-safe rules

The interpretation service BIWS consists of a modified abduction approach: the original abduction equation **Fehler! Verweisquelle konnte nicht gefunden werden.**is modified to

$$\Sigma \cup \Gamma_1 \cup \Delta \models \Gamma_2$$

by splitting the assertions in Γ into bona fide assertions Γ_1 and assertions requiring fiats Γ_2. Bona fide assertions are assumed to be true by default, whereas fiat assertions are aimed to be explained. The abduction retrieval inference service has the task of acquiring what should be added to the knowledge base in order to positively answer a query. The analysis Aboxes represented in Figure 8 and Figure 9 are assertions in Γ which have to be partitioned into bona fide and fiat assertions.

In this example, the bona fide assertions in Figure 8 are $image_1 : Image$, $bar_1 : HorizontalBar$, and $body_1 : PersonBody$, *and* $face_1 : PersonFace$. And Γ_2 contains the fiats ($body_1$, bar_1):*isAdjacent*, and ($body_1$, $face_1$):*isAdjacent* . Respectively, the bonafide assertions in Figure 9 are $caption_1 : Caption$, $pname_1 : PersonName$, and $perf_1 : Performance$ with the string values ($pname_1$:(*string=hasValue*"*KajsaBergqvist*")), and ($perf_1$:(*string=hasValue*"2.06")). And Γ_2 contains the fiats ($pname_1$,$perf_1 : personNameToPerformance$).

Insert Figure 11 Here

Figure 5. The interpretation Abox IAboxImage1. This Abox represents the first explanations Δ_1 for the analysis Abox presented in Figure 8.

Insert Figure 12 Here

Figure 6. The interpretation Abox IAboxImage2. This Abox represents the second explanations Δ_2 for the analysis Abox presented in Figure 8.

The interpretation service delivers for the analysis Abox *AAboxImage* two explanations Δ_1 and Δ_2 which are presented in Figure 11 and Figure 12. The interpretation service delivers for the analysis Abox *AAboxCaption* one explanation which is presented in Figure 13.

Insert Figure 13 Here

Figure 73. Interpretation Abox caption. This Abox represents the explanation Δ for the analysis Abox presented in Figure 9.

Fusion Process

The fusion process is a new service which completes the interpretation service. Fusion is based on the Abox abduction and Abox differences approaches which are used to fuse knowledge assets. The resulting fused knowledge assets which get associated are called *semantic assets*. In this example, each interpretation Abox contains an instance of the particular media type (i.e., $image_1$,, $caption_1$). Additionally, these media specific instances have a *depicts* relation to all other instances. Consequently the fiat assertions are generated. From the document structure, we can generate the following document specific rules for caption texts, images, and texts:

$$hasCaption(X, A) \leftarrow Image(X), depicts(X, Y), Caption(A), depicts(A, B), same\_as(Y, B)$$
$$hasImage(X, A) \leftarrow Text(X), depicts(X, Y), Image(A), depicts(A, B), same\_as(Y, B)$$
$$hasText(X, A) \leftarrow Caption(X), depicts(X, Y), Text(A), depicts(A, B), same\_as(Y, B)$$

In this example, image and caption will be fused because it makes sense that image and caption belongs together. In this case Γ_2 contains the fiat assertion $hasCaption(image_1, caption_1)$. The result of the fusion process is $same\_as(IND_3, IND_{11})$, that means the person in the image is the same person in the caption text.

The following two examples show the query process of symbolic content descriptions. The examples show the effects for the information retrieval for not fused (Example 1) and fused multimedia documents (Example 2).

Example 1:
Assume a sports reporter is searching for images showing high jump events.

Insert Figure 14 Here

Figure 8. A sample multimedia document with athletics news. Adapted from (IAAF, 2009).

The multimedia document in Figure 14 is in the multimedia repository as well as the related interpretation Abox presented in Figure 15.

Insert Figure 15 Here

Figure 9. Interpretation Abox IAboxImage2. This Abox represents the results after the analysis and the interpretation process for the image in Figure 14.

In this example, the fiat assertion is $\Gamma_2 = (Image, HighJump: depicts)$. The interpretation service gives the answer $\text{image}_1$and therefore the sports reporter retrieves the correct document.
If another sports reporter is interested in images with pole vault events, consequently the fiat assertion is $\Gamma_2 = (Image, PoleVault: depicts)$. The interpretation service delivers also the answer image_1. This image shows a high jump event and therefore the sports reporter retrieves the false (positive) document. This lies on the interpretation process for this image. The interpretation process generated two possible explanations for this image: PoleVault and HighJump. That is why the sports reporter receives this image. In this example the precision of the results is not increased.

Example 2:
This example shows that ontology-based content descriptions and Abox fusion provide an increase in precision. The multimedia document in Figure 14 has the interpretation Abox *Text* represented in Figure 16.

Insert Figure 16 Here

Figure 10. Interpretation Abox Text. This Abox represents the results after the analysis and the interpretation process for the text in Figure 14.

The fusion operator has the aim to fuse equal concept names. In this example, the fiat assertion is $\Gamma_2 = (Image, PoleVault: depicts)$. The interpretation service delivers no answers which is correct.
The results of the interpretation and Abox fusion process are sets of relational structures that represent aggregates that have parts of other aggregates. These configurations of such structures have the advantage to keep the information from aggregate to aggregate and therefore to increase the precision of the extracted metadata (Espinosa, Kaya, and Möller, 2009). In this example the precision increases.

Semantic Assets in a Knowledge Creating Process

In this example the semantic assets are assigned to four asset types. An E-asset contains tacit knowledge. In this example the document in Figure 7, the Aboxes in Figure 8 and Figure 9 are unstructured data which describe an image. They are stored in an E-asset. A C-asset contains ontology-based and holistic content descriptions, the concepts such as ontologies, DL-safe rules and the rule *hasCaption*. Aboxes such as given in Figure 8, Figure 9, Figure 10, Figure 11, Figure 13, Figure 14, and the results from the fusion process are ontology-based content descriptions. An S-asset contains the combination of ontology-based and holistic content representations for an entity. An R-asset contains explicit knowledge from the S-asset which will be shared to all people which were involved in the definition process of these S-assets.

A Way for the Combination of Ontology-based and Holistic Representations

A systematically combining of holistic and ontology-based content descriptions is given if holistic representations are exploited for similarity searches. This first step leads to high recall. Then, the similarity searches results will be represented via ontologies. These ontology-based representations are giving more precise matches. Accordingly, the holistic representations of these matches being found, are used as a filter in order to increase recall while at least maintaining precision.

Insert Figure 17 Here

Figure 11. Combination of ontology-based and holistic representations.

For example, the term-document-matrix in Table 1 is illustrated in Figure 17. The LSI approach distinguishes two groups: group *A* which represents the *athletics* domain and group *B* which represents the *fairyTale* domain. Consider, it seems that the multimedia document in Figure 7 does not contain any information about the type of the sports activity. Via LSI it is possible to categorize it into the athletics domain but not into the concrete category *HighJump*. This result can be achieved by the ontology-based approach.

Summary

A systematic combination of holistic and ontology-based content descriptions is presented. The first step for the combination of both representation types is the holistic representation of content. Then, holistic representations are exploited for similarity searches which lead to high recall. Further on, the similarity searches results will be represented via ontology-based content descriptions. These descriptions give more precise matches. Taken together, the systematic combination of ontology-based and holistic content descriptions ensures the retrieval of high-quality documents and implies an efficient knowledge management environment.

This is one way to define the relationship between holistic and ontology-based content descriptions. Techniques for holistic content descriptions are well established (Manning et al., 2008) such as the latent semantic indexing (LSI) approach (Manning et al., 2008, pp. 403–419). It is known that LSI has a substantially high recall (Gee, 2003). This is why we suggest combining LSI with ontology-based techniques, namely the multimedia interpretation and fusion approaches. It is also well known that ontology-based techniques lead to high precision (Blank et al., 1992; Espinosa et al., 2007a; Espinosa et al., 2007b). The suggested systematic combination of both representation leads to high recall while at least maintaining precision. This kind of improvement was done in the context of knowledge management.

CONCLUSION

In fact for companies, the practices of KM are necessary and essential because KM increases the quality of knowledge. The quality of knowledge is measured on a high recall and precision. This contribution suggests a way how to combine ontology-based and holistic content descriptions systematically in order to increase recall while at least maintaining precision. Ontology-based content descriptions were represented via logic-based techniques in order to increase precision. For the holistic representation the latent semantic indexing approach was used which leads to high recall. A way for the combination of holistic and ontology-based representations was presented. Additionally in this context, it was shown that the systematic combination leads to an effective knowledge management environment.

FUTURE RESEARCH DIRECTIONS

The non-standard information retrieval inference service Abox abduction and the multimedia interpretation process are used in order to generate ontology-based content descriptions. In the multimedia process, annotated entities of a multimedia document, also called individuals, will be fused with other one's if the individuals describe the same real-world entity. However, this fusion algorithm applies only to document specific individuals. A future research could be the extension of the fusion algorithm. Hence, the expanded fusion algorithm could be applied to all individuals in order to increase precision.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my supervisor, Professor Dr. rer. nat. habil. Möller, Head of the research group Intelligent Autonomous Systems, Institute for Software Systems, Hamburg University of Technology. His wide knowledge and his logical way of thinking were of great value for me and made it possible for me to write this contribution.

REFERENCES

Baader, F., Calvanese, D., McGuinness, D., Nardi, D. &, Patel-Schneider, P.F., (Eds.). (2003). *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press.

Baader, F. & Nutt, W. (2003). Basic description logics. In Baader et al. (Eds.), *Description Logic Handbook: Theory, Implementation, and Applications* (pp. 43–95). Cambridge University Press.

Birbeck, M. (2004). XHTML and RDF. Retrieved from http://www.w3.org/MarkUp/2004/02/xhtml-rdf

Blank, D. S., Meeden, L. A., & Marshall, J. B. (1992). Exploring the Symbolic/Subsymbolic Continuum: A Case Study of RAAM. In John Dinsmore (Ed.), *The Symbolic and Connectionist Paradigms: Closing the Gap* (pp. 113–148). Erlbaum, Hillsdale, NJ.

Bossung, S. (2008). *Conceptual Content Modeling - Languages, Applications, and Systems*. dissertation.de. Germany.

Bossung, S., Sehring, H.-W., & Schmidt, J. W. (2005). Conceptual Content Management for Enterprise Web Services. In *Perspectives in Conceptual Modeling: ER 2005 Workshops CAOIS, BP-UML, CoMoGIS, eCOMO, and QoIS*, volume 3770 / 2005 of *Lecture Notes in Computer Science* (pp. 343–353). Springer-Verlag.

Bossung, S., Sehring, H.-W. Skusa, M., & Schmidt, J. W. (2005). Conceptual Content Management for Software Engineering Processes. In *Advances in Databases and Information Systems: 9th East European Conference, ADBIS 2005*, volume 3631 / 2005 of *Lecture Notes in Computer Science* (p. 309). Springer-Verlag.

Cortada, J. W., & Woods, J. A. (Ed.). (1999). *The Knowledge Management Yearbook 1999-2000*. Butterworth-Heinemann.

De Brun, C. (2005). ABC of Knowledge Management. *NHS National Library for Health: Knowledge Management Specialist Library*.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the american society for information science*, 41(6), 391–407.

Dierkes, M., Antal, A. B., Child, J., & Nonaka, I. (Eds.). (2003). *Handbook of Organizational Learning and Knowledge*. Oxford University Press.

Espinosa Peraldi, I. S. (2011). *Content management and knowledge management: two faces of ontology-based deep-level text interpretation*. Pdh thesis. Retrieved from http://doku.b.tu-harburg.de/volltexte/2011/1124/pdf/PhDThesis_SofiaEspinosa.pdf

Espinosa Peraldi, I.S., Kaya, A., & Möller, R. (2009). The boemie semantic browser: A semantic application exploiting rich semantic metadata. In *Proceedings of the Applications of Semantic Technologies Workshop (AST-2009), Lübeck, Germany*.

Espinosa, S., Kaya, A., & Möller, R. (2009). Formalizing multimedia interpretation based on abduction over description logic aboxes. In *Proc. of the 2009 International Workshop on Description Logics DL- 2009.* Oxford, United Kingdom.

Espinosa Peraldi, I. S, Kaya, A., Melzer, S., Möller, R., & Wessel, M. (2007a). Multimedia Interpretation as Abduction. In *Proc. DL-2007: International Workshop on Description Logics*, 2007.

Espinosa Peraldi, I. S., Kaya, A., Melzer, S., Möller, R., & Wessel, M. (2007b). Towards a media interpretation framework for the semantic web. *The 2007 IEEE/WIC/ACM International Conference on Web Intelligence (WI'07)*. Washington, DC, USA

Epinosa Peraldi, I. S., Kaya, A., Melzer, S., & Möller, R. (2008). On ontology based abduction for text interpretation. In A. Gelbuhk (Ed.), *Proc. of 9th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2008)* (pp. 194–205). International Society of the Learning Sciences.

Gee, K. R. (2003). Using latent semantic indexing to filter spam. In *In Proceedings of the 2003 ACM symposium on Applied computing*, pp. 460–464. ACM Press.

Handschuh, S. & Staab, S. (2002). Authoring and annotation of web pages in cream. *Proceedings of the 11th International World Wide Web Conference*.

Handschuh, S., Staab, S., & Volz, R. (2003). On deep annotation. *Proceedings of the 12th International World Wide Web Conference*.

IAAF (International Association of Athletics Federations) (2009), Retrieved from http://www.iaaf.org.

Kaya, A. (2011). *A Logic-Based Approach to Multimedia*. Mensch & Buch Verlag. Berlin.

Melzer, S. (2006). *A content-based publish-subscribe architecture for individualized sensor data supply*. Master thesis, Hamburg University of Technology (TUHH).

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval* (1st ed.). Cambridge University Press.

Mugo, D. M. (2010). Connecting *People using Latent Semantic Analysis for Knowledge Sharing*. Master thesis, Hamburg University of Technology (TUHH), January 2010.

Nonaka, I. & Konno, N. (1998). The concept of 'ba': Building a foundation for knowledge creation. *California Management Review*, 40(3), 40–54.

Nonaka, I. (2008). *The Knowledge-Creating Company (Harvard Business Review Classics)*. Harvard Business School Press.

Nonaka, I., & Takeuchi, H. (1995). *The knowledge-creating company: how Japanese companies create the dynamics of innovation*. Oxford University Press, New York.

Nonaka, I., & Teece, D. J. (Eds.). (2001) *Managing Industrial Knowledge: Creation, Transfer and Utilization*. Sage Publications, Inc., Thousand Oaks, CA, USA.

Nonaka, I., Toyama, R. & Byosière, P. (2003). A Theory of Organizational Knowledge Creation: Understanding the Dynamic Process of Creating Knowledge. In Dierkes et al. (Eds.), *Handbook of Organizational Learning and Knowledge* (pp. 291-517). Oxford University Press.

Paliouras, G. (2009). *Boemie final report*. Technical report, Hamburg University of Technology. Final version 1.0.

Racer Systems GmbH & Co. KG (2007). *Racerpro manual 1.9*.

Reuters, T. (2010). OpenCalais. Retrieved from http://www.opencalais.com/.

Touretzky, D. S. (1990). Boltzcons: Dynamic symbol structures in a connectionist network. *Artificial Intelligence*, 46(1-2),5 - 46.

Sun, R & Peterson, T. (1998). A subsymbolic+symbolic model for learning sequential navigation. In *Proceedings of the fifth international conference on simulation of adaptive behavior on From animals to animats 5* (pp. 246–251), Cambridge, MA, USA. MIT Press.

Schmidt, J. W., Sehring, H.-W., Bossung, S. (2005). Active Learning By Personalization - Lessons Learnt from Research in Conceptual Content Management. In *Proceedings of the 1st International Conference on Web Information Systems and Technologies* (pp. 496–503). INSTICC Press Miami.

Schmidt, J. W., & Sehring, H.-W. (2003). Conceptual Content Modeling and Management: The Rationale of an Asset Language. In *Perspectives of System Informatics* (pp. 469–493). Springer Verlag.

Schmidt, J. W., Sehring, H.-W., & Warnke, M. (2001). Der Bildindex zur Politischen Ikonographie in der Warburg Electronic Library - Einsichten eines interdisziplinären Projektes. In *Archivprozesse* (pp. 238–268). Dumont.

Sehring, H.-W., & Schmidt, J. W. (2004). Beyond Databases: An Asset Language for Conceptual Content Management. In *Proceedings of the 8th East European Conference on Advances in Databases and Information Systems* (pp. 99–112). Springer-Verlag.

Sehring, H.-W. (2004). *Konzeptorientiertes Content Management: Modell, Systemarchitektur und Prototypen*. dissertation.de. Germany

Studer, R., Motik, B., & Sattler, U. (2005). Query Answering for OWL-DL with Rules. *Journal of Web Semantics, 3 (1), 41-60*.

Velinska, A. (2010). *Tag Cloud Control by Latent Semantic Analysis (LSA)*. Master thesis, Hamburg University of Technology (TUHH).

W3C (2011). HTML+RDFa 1.1. *W3C Working Draft*. Retrieved from http://www.w3.org/TR/2011/WD-rdfa-in-html-20110525/.

Wiig, K. (1986). Knowledge Based Systems. In *Expert Systems: Impacts & Potentials - Proceedings of the International conference*. London.

Wiig, K. (1997). Knowledge management: Where did it come from and where will it go? *Expert Systems with Applications*, 13(1), 1–14.

KEY TERMS & DEFINITIONS

Asset: a real-world entity which is based on a content-concept pair.

Holistic representation: holistic content descriptions which will be created via holistic techniques.

Ontology-based representation: symbolic content descriptions which will be created via logic-based techniques.

Precision: the number of relevant items retrieved divided by the overall number of retrieved items.

Recall: as the number of relevant items retrieved divided by the number of relevant items in the repository.

Semantic asset: management units in a knowledge management area which will be generated in an automatic process.

Symbolic representation: ontology-based representation.

Systematic combination: the combination of two content representation formats with the result to increase recall while at least maintaining precision.

Tables

Document	d1	d2	d3	d4	d5	d6	d7	d8
Terms								
Athlete	12	1	45	43	3	0	0	0
HighJumpCompetition	2	0	0	2	1	0	0	0
LongJumpCompetition	2	0	0	2	0	0	0	0
PoleVaultCompetition	6	1	1	0	1	0	0	0
PoleVaulter	0	0	0	0	0	3	0	0
Person	99	18	22	66	8	0	0	0
Snow White	0	0	0	0	0	5	156	0
Snow White and the	0	0	0	0	0	0	11	0

Seven Dwarfs								
Snow White and Rose Red	0	0	0	0	0	0	4	0
Prince Charming	0	0	0	0	0	15	2	2
Sleeping Beauty	0	0	0	0	0	4	1	1
Rapunzel	0	0	0	0	0	1	0	1

Table 2. Term-document-matrix in the athletics and fairyTale domain.