Design and Construction of Semantic Document Networks Using Concept Extraction

Simon Boese, Satzmedia GmbH, Hamburg, Germany
Torsten Reiners, Curtin University, Bentely, Australia, *t.reiners@curtin.edu.au*
Lincoln C. Wood, Curtin University, Bentely, Australia, *l.wood@curtin.edu.au*

Abstract

Processing of unstructured documents according to their content is required in many disciplines; e.g., machine translation, text analysis and mining, and information extraction and retrieval. Whilst research in fields like text analysis, conceptualisation, or design of semantic networks progressed crucially over the last years, we still observe gaps between state-of-the-art algorithms to extract concepts from documents and how these concepts are linked effective and efficiently. This paper proposes a framework to store processed documents in a specialised semantic network database to enhance retrieval and analysis of common concepts in documents. We apply *natural language reduction* to calculate semantic cores for the concept-based indexing of stored documents. The developed prototype demonstrates an advanced document storage as well as a fast (semantical) retrieval of documents based on given key concepts.

1 Introduction

One key aspect in document management systems is the storage and retrieval of (un-)structured electronic documents. In general, meta-information as well as key terms associated with the document are used for indexing and classification. Due to the user generated content paradigm of the Web 2.0 and the continuous progress of document digitalisation, we encounter an exponential increase of mainly unstructured documents and, as a consequence, advanced challenges to increase (and even maintain) the retrieval quality. Improved search engines consider stem forms, synonyms or translations to the document language [18], a match to search request generally requires the words to be part of the document while the meaning and context is ignored. Semantic analysis can support the search by including additional associated terms (i.e. determining the concept of terms) and scenarios; e.g. the term “trunk” is used in subjects like car repair, travel accessories or safari reports. With the Web 2.0 smoothly shifting to the next version, we expect computers to process information on a higher level and grasp the meaning of words. Instead of handling documents, so-called (intelligent) agents are supposed to, for example, extract information and especially concepts according to our individual preferences, grade (free-text) exams, summarise correspondence or translate documents. In all cases, the understanding of natural language without any structure is essential to guarantee robustness and reliability with respect to quality.

Our research is centered on advanced document storage and fast queries to retrieve documents with the same concepts [9]. The preprocessing, i.e., semantic analysis, is still crucial as the outcome influences the storage quality significantly. The applied and later described algorithm was chosen as it demonstrated a good quality-perfomance ratio; nevertheless, other algorithms and the state-of-the-art literature were evaluated and considered for inclusion; see [8] for more details. Here, with respect to our focus, we refer to [31, 39, 27, 32, 6] for an overview of well-grounded and elaborated methods for information extraction and concentrate on the semantic document network.

In this contribution, we focus on the aspect of storing documents in a semantic document network after we extracted meta, structure and content information using semantic analysis. Subsequent to a brief introduction to conceptualisation, we introduce the semantic document network in Section 3; i.e. how information can be stored in an interlinked network. Using a short sample, we visualise in Section 2.4 the calculation of the semantic core using *conceptual indexing* as well as the embedding in an existing semantic document network. Our proposed framework avoids the transformation of the network as we use a semantic network database. Its advantage is summarised in Section 4.

2 Conceptualisation

A concept is described by one or multiple words and associated with a (semantic) category. These categories represent the meanings or senses of the containing words, whereas the interpretation might differ when the domain, context, language, or person changes [7]. That is, the word *trunk* in the context of motor vehicles refers to the storage room, while the same word in the context of traveling might be about an object to keep your clothes; see Figure 1. Here, the second story (solid arrows) uses all concepts (trunk, jeep, safari) and refers to trunk as the elephants' body part, while the first story is about a suitcase. The kind of arrow represents parameters associated with the relations. Note that the terms bear even further meanings, which are ignored here: Safari is an Internet browser and trunk is also the stem of a tree or torso.

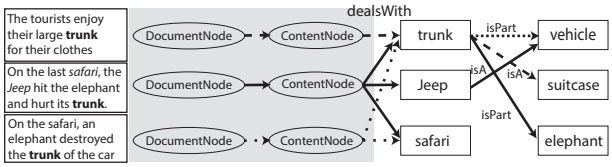

Figure 1: Different document networks using the same context. For clarity, the relations are reduced to the necessary ones to demonstrate the example.

Conceptualisation is the process of detecting a text's meaning; given as a set of concepts. The goal is to identify descriptive generic terms that characterise the entire text. Such concepts basically reduce the text to its most relevant elements with respect to content and can be regarded as a footprint. Two different texts could be seen as identical (telling the same story) if their footprint matches.

Humans intuitively employ their cognitive abilities to do conceptualisation. It facilitates them to generalise or abstract from the full text. Under the assumption that the reader knows the used vocabulary, one will be able to *understand* the text and possibly deduce knowledge from it [30]. The challenge here is to develop a software that is capable to reproduce this process. As to the example above, the conceptualisation must also cover scenarios, where people talk about *trunks* (of elephants they saw last holiday) while being in the garage with the jeep. The concept for *rear storage area in a vehicle* results in an erroneous footprint and, therefore, later misinterpretation.

Conceptualisation is used for manifold applications. Their common denominator is the deduction of a concept hierarchy, leading from a general to a more specific concept and vice versa. Examples are **information retrieval** (meaningful retrieval of stored information from a data repository with digital artefacts [25]), **automated essay grading** (comparing essays using their feature space – relevant concepts as well as their relations – against a model answers [38]), **plagiarism** (detection of reproduced ideas instead of just word-by-word-copied paragraphs [13]) and **building ontologies** (an ontology is formed by *Is-A* relations (generalisation and specialisation of concepts) [10], whereas the whole interplay of these relations construct ontology schema [7].

2.1 Fields of Research

The identification of concepts in documents requires different technologies with significant importance for the overall quality. Here, we briefly highlight four different fields of research; see also [1]: Tagging part of speech (POS), named entity recognition (NER), entity tracking (ET) and relation extraction (RE):

POS classifies every string sequence (separated by white spaces) and annotates it with the determined part of speech; e.g., noun, verb or adjective. Punctuation marks are used to interpret the structure which helps to identify subordinate clauses [1].

NER filters expressions that either stand for (known) people, organisations or places [36]. In general, the detection of named entities (proper nouns) depends on the coverage of the underlying entity lists.

ET identifies individual objects that are referenced differently throughout a text [24]. Such occurences are called co-references [1]. They appear as named entities (either abreviations like "USA" for "United States" or synonyms like "the state of liberty and freedom") and pronouns ("Mr. X vanished. **He** hasn't been seen ever since.") [16].

RE exploits a sentence's structure. It focuses on verbs but also respects involved objects to infer a relation between them. Therefore, relation extraction assigns active or passive roles to participating objects. Hence, relations are often expressed as predicates [17, 19].

2.2 Methodologies

Different techniques have been utilised to retrieve the most relevant concepts from a text; here under the assumption that the text is unstructured or semi-structured.

Formal Concept Analysis (FCA) is a pure mathematical approach to deduce knowledge from texts [37]. It draws upon the lattice theory, which utilises partially ordered sets. The key element of the FCA is the definition of contexts. A context is expressed by a tuple: a set

of entities, a set of attributes and a binary relation between them [15, 37]. Thus, a subset of a context defines a concept and concept hierarchies can be described and processed using set notation [11].

Latent Semantic Analysis (LSA) infers contextual relations between terms [21, 23]. The LSA assumes that there are hidden, i.e. latent, coherences throughout the analysed passages [20]. The starting point of the LSA is a correlation matrix over terms and their occurrences in the passages. This matrix states in which contexts a term does or does not occur. The individual rows are considered as independent term vectors.

After transforming the matrix values with a weight function the correlation matrix is split to three different matrices using Singular Value Decomposition (SVD). Next, a dimension reduction is executed upon the result of the SVD with the help of the original correlation matrix' rank. Afterwards, the SVD is reverted. The result is a matrix, reduced in dimension and reduced from noise which represents the most important terms or concepts respectively.

Concept-based Indexing (CBI) assembles a *semantic core* in form of a semantic network [5]. In such networks, nodes stand for the most important concepts which are identified based on the semantic relatedness among each other. The result describes the content of a document, which further contributes directly to the assemblance of a the semantic document network (see Section 3). In combination with the succeeding storage of the document network in the semantic database this procedure forms the *conceptualised document saving* (CDS). The following section describes this technique in more details as we adopted it for our work.

2.3 Concept-based Indexing

First, appropriate **concept candidates** are identified. Candidates might be adjacent (compouned terms) or single words. For so-called multiword concepts, the synsets – a set of semantically equivalent terms – are requested from WordNet (note that candidates are disposable, if they do not exist in WordNet).[1] Next, the concept candidates are weighted by $w(c_i) = cf(c_i) \cdot \ln(N/df(c_i))$ with N being the total number of documents that are analysed at once and df being the frequency of the i^{th} concept c accross all these documents. $cf(c_i)$ denotes the concept frequency in the current document and is calculated $cf(c_i) = \text{count}(c_i) + \sum_{sc \in \mathscr{S}_{c_i}} (\text{length}(sc)/\text{length}(c_i) \cdot \text{count}(sc))$

The functions `count` and `length` describe the number of times that c_i occurs throughout the text and the number of words it comprises. The set $\mathscr{S}_{c_i}$ holds all subconcepts of c_i. Its elements are all partially orded subsets of words of c_i; e.g., $c_i =$ "*House of Commons*" incorporate $\mathscr{S}_{c_i} = \{$ "*House of*", "*of Commons*", "*House*", "*of*", "*Commons*" $\}$. Only concept candidates whose weight exceed a certain threshold are retained.

Secondly, the **semantic relatedness** of concept candidates is calculated using an adapted Lesk Algorithm [22]. This algorithm resolves ambiguities of words with the help of machine readable dictionaries. It detects overlaps in the words' definitions [3, 22]. We use multiple WordNet relations accross synsets to compute the overlap $\mathscr{O}$ of concept candidates that are subsumed in the set $\mathscr{C}$; given in Equation (1). The relations are comprised in the set $\mathfrak{R} = \{\mathfrak{R}_1, \mathfrak{R}_2, \ldots, \mathfrak{R}_r\}$.[2]

This overlap is computed for all v meanings of a candidate c_k ($\mathscr{M}_v^k$) in combination with all meanings of the other concept candidates.

$$\mathscr{O}(\mathscr{M}_v^k, \mathscr{M}_w^l) = \sum_{i,j \in \{1,\ldots,card(\mathfrak{R})\}} \mathfrak{R}_i(\mathscr{M}_v^k) \cap \mathfrak{R}_j(\mathscr{M}_w^l) \tag{1}$$

$$\forall_{k \neq l}\ k, l \in \mathscr{C}\ \wedge\ \forall\ v, w \tag{2}$$

The intersection of two relations $\mathfrak{R}_i(\mathscr{M}_v^k)$ and $\mathfrak{R}_j(\mathscr{M}_w^l)$, called Lesk Overlap, is defined as sum of conjoint words whereas the number of adjacent conjoint words is squarred [4, 22].

In the final step, the **semantic core** is determined out of all possible combinations of concept meanings. For each of the x_k meanings of a concept candidate the relatedness to all other candidates' meanings is aggregated and forms the meaning's score:

$$\text{score}(\mathscr{M}_v^k) = \sum_{l \in \{1,\ldots,n\}, w \in \{1,\ldots,x_l\}, l \neq k} \mathscr{O}(\mathscr{M}_v^k, \mathscr{M}_w^l) \tag{3}$$

The best scored meaning of each concept candidate is finally selected to compose the semantic core.

$$\mathscr{SC} = (\text{best}(c_1), \ldots, \text{best}(c_n)) \tag{4}$$

$$\text{best}(c_k) = \max_{v \in \{1,\ldots,x_k\}} \text{score}(\mathscr{M}_v^k) \tag{5}$$

[1] This approach assumes that single word concepts come along with at least one meaning; i.e. are in multiple synsets. WordNet is a lexical database developed at Princeton University (`http://wordnet.princeton.edu/`). WordNet stores concepts in *synsets*. A synset comprises synonym words as well as a gloss definition for the altogether meaning [14]. A single word might belong to multiple synsets, which reflects its ambiguity. Among synsets different kinds of relations exists.

[2] These relations include the gloss, hyponymy, hypernymy, meronymy, holonymy, category, usage and region. Note: They are deduced in WordNet using FCA (see Section 2.2) [26].

Thus, the selected concept meanings represent the nodes of the semantic core. Edges between these nodes contain the semantic relatedness of the corresponding meanings; i.e. the value of $\mathcal{O}(\text{best}(c_k), \text{best}(c_l))$.

2.4 Example for Concept-based Indexing

The exemplary snippet to demonstrate the algorithm is reduced to the text shown below, where the identified concept candidates are already highlighted; see also [8].

> "The ***House of Commons*** is the ***name*** of the elected ***lower house*** of the ***bicameral parliaments*** of the ***United Kingdom*** and ***Canada***. Historically it was the ***name*** of the ***lower houses*** of ***Ireland*** and ***North Carolina***."[3]

The candidates are selected by checking each single word whether it belongs to a defined stop word category[4]. Exceptions are for named entities where stop words are part of the candidates; here, for example, "House ***of*** Commons". The identified terms in all sentences are retained as concept candidates.

Next, the weights of the candidates are computed. The following example shows the calculation for the concept candidate "House of Commons". The term $\ln(\frac{N}{df(c_i)})$ equals 1 due to the small size of the sample we use here.

$$
\begin{aligned}
&w(\text{House of Commons}) = count(\text{House of Commons}) \\
&+ \frac{length(\text{House of})}{length(\text{House of Commons})} \cdot count(\text{House of}) + \frac{length(\text{of Commons})}{length(\text{House of Commons})} \cdot count(\text{of Commons}) \\
&+ \frac{length(\text{House})}{length(\text{House of Commons})} \cdot count(\text{House}) + \frac{length(\text{of})}{length(\text{House of Commons})} \cdot count(\text{of}) \\
&+ \frac{length(\text{Commons})}{length(\text{House of Commons})} \cdot count(\text{Commons}) \\
&= 1 + \tfrac{2}{3} \cdot 1 + \tfrac{2}{3} \cdot 1 + \tfrac{1}{3} \cdot 2 + \tfrac{1}{3} \cdot 6 + \tfrac{1}{3} \cdot 1 \\
&w(\text{House of Commons}) = 5.\overline{3}
\end{aligned}
$$

The weights of the remaining concept candidates are summarised in Table 1 (left side) together with the number of meanings of each concept candidate i in WordNet (denoted as x_i).[5]

The threshold to accept candidates for further processing is set to 2.

Next, the semantic relatedness for all retained concept candidates' meanings ($\mathscr{M}^i_{x_i}$) are calculated as defined in Equation (1). In contrast to the approach from [5] we disrespect words belonging to a subset of stop word categories as mentioned above. Here, we demonstrate the procedure for the candidate "House of Commons" ($\mathscr{M}^1_1$), for which the gloss of its meaning is compared with the gloss of the second meaning of "North Carolina" ($\mathscr{M}^9_2$).[6] The results of all comparisons are shown in Table 1 with the relatednesses of all candidates' meanings.

i	Concept candidate	Weight	x_i	Retain
1	House of Commons	$5.\overline{3}$	1	×
2	name	2	6	×
3	lower house	3	0	
4	bicameral parliaments	2	0	
5	United Kingdom	2	1	×
6	Canada	1	1	
7	lower houses	2.5	0	
8	Ireland	1	2	
9	North Carolina	2	2	×

	$\mathscr{M}^1_1$	$\mathscr{M}^2_1$	$\mathscr{M}^2_2$	$\mathscr{M}^2_3$	$\mathscr{M}^2_4$	$\mathscr{M}^2_5$	$\mathscr{M}^2_6$	$\mathscr{M}^5_1$	$\mathscr{M}^9_1$	$\mathscr{M}^9_2$	score($\mathscr{M}^i_{x_i}$)
$\mathscr{M}^1_1$	0	1	0	0	0	1	0	0	3	6	11
$\mathscr{M}^2_1$	1	0	0	0	0	0	0	1	1	4	7
$\mathscr{M}^2_2$	0	0	0	0	0	0	0	0	0	0	0
$\mathscr{M}^2_3$	0	0	0	0	0	0	0	1	0	0	1
$\mathscr{M}^2_4$	0	0	0	0	0	0	0	0	0	2	2
$\mathscr{M}^2_5$	1	0	0	0	0	0	0	1	1	2	5
$\mathscr{M}^2_6$	0	0	0	0	0	0	0	0	0	0	0
$\mathscr{M}^5_1$	0	1	0	1	0	1	0	0	0	14	17
$\mathscr{M}^9_1$	3	1	0	0	0	1	0	0	0	4	9
$\mathscr{M}^9_2$	6	4	0	0	2	2	0	18	4	0	36

Table 1: **Left:** Different document networks using the same context. **Right:** Matrix of the semantic relatedness $\mathcal{O}(\mathscr{M}^i_{x_i}, \mathscr{M}^j_{x_j})$ of all meanings of all retained concept candidates together with their resulting score.

$\mathfrak{R}_{\textbf{gloss}}(\mathscr{M}^1_1)$ = "The lower house of the British parliament."

$\mathfrak{R}_{\textbf{gloss}}(\mathscr{M}^9_2)$ = "One of the British colonies that formed the United States."

$$
\begin{aligned}
&\mathfrak{R}_{\text{gloss}}(\mathscr{M}^1_1) \cap \mathfrak{R}_{\text{gloss}}(\mathscr{M}^9_2) = \\
&\begin{array}{cccccccc}
\text{The} & \text{lower} & \text{house} & \text{of} & \text{the} & \text{British} & \text{parliament} & \\
\times & 0 & 0 & \times & \times & 1 & 0 & = 1
\end{array}
\end{aligned}
$$

Note that this value differs from the one in the table as it is only a partial result as the example here calculated only one overlapping computation but all from WordNet relations. In total the semantic relatedness of both concept meanings $\mathcal{O}(\mathscr{M}^1_1, \mathscr{M}^9_2)$ is 6.

The column score ($\mathscr{M}^i_{x_i}$) in Table 1 (right side) shows the overall score for each concept meaning,

[3] From `http://en.wikipedia.org/wiki/House_of_Commons`

[4] In this list are articles, prepositions, verb forms and types, pronouns, adverbs, symbols, list markers and conjunctional words.

[5] Note that "name" occured twice and is merged into one candidate. "lower house", "lower houses" and "bicameral parliaments" are dropped because WordNet does not know these expressions; even though the weight is above 2 for all cases.

[6] For the sake of completeness: The first gloss ($\mathscr{M}^9_1$) is "Old North State, Tar Heel State, NC – (a state in southeastern United States; one of the original 13 colonies"

which is calculated by summing up all columns of the according row. For the semantic core, the best rated concept meaning of each candidate c_i is selected ($\max_{x_i} \text{score}(\mathscr{M}^i_{x_i})$). The resulting semantic core is:

$$\mathscr{SC} = (\text{best}(c_1), \text{best}(c_2), \text{best}(c_5), \text{best}(c_9)) = (\mathscr{M}^1_1, \mathscr{M}^2_1, \mathscr{M}^5_1, \mathscr{M}^9_2)$$

Figure 2 visualises the semantic core, whereas the concept nodes are already connected to the central node of a partial document network (ContentNode).

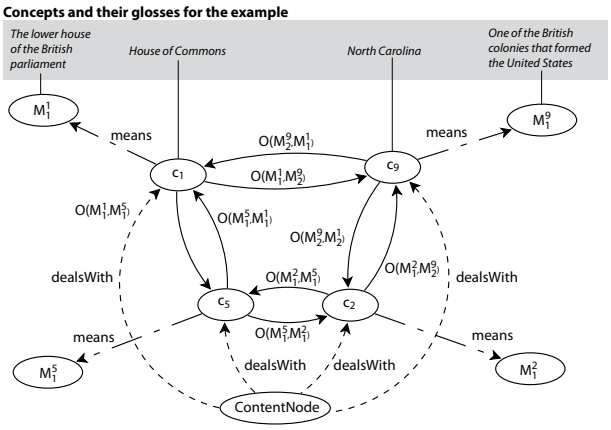

Figure 2: Semantic core of the exemplary text

3 Semantic Document Networks

The preparation of documents for automated processing is relevant; i.e. deriving implications (based on already existing knowledge) or reasoning by given rules. Nevertheless, the transition of documents to semantic networks is most of the time restricted to the meaning of the document rather than including the structure itself [28].

3.1 Partial Networks

Content representation is only one part of the story. To preserve documents in a semantic network, we also have to encode and store the document's structure and its corresponding meta-information. Petfli [28] suggests individual networks for each dimension being connected to one root node (DOCUMENTNODE); see Figure 3. The partial networks are disjunct to permit independent and parallel construction; nevertheless, individual nodes of one network can be connected to one of the others to link, e.g., a concept to a specific section. Note that we distinguish between *information* nodes (structure; oval elements) and *content* nodes (literals; square elements).

The network *meta-information* (starting at METANODE) contains all information to describe the document

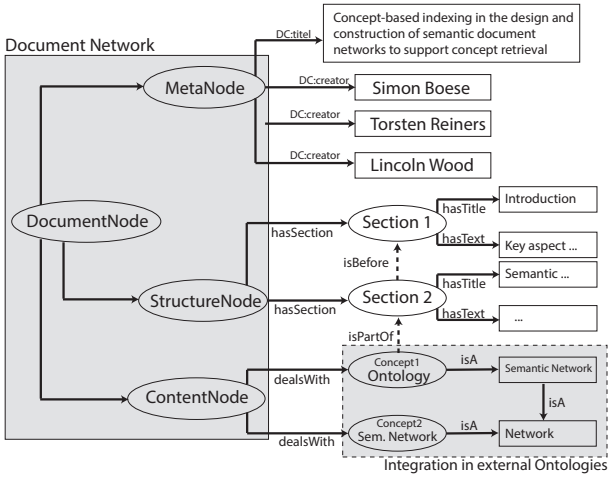

Figure 3: An exemplary semantic document network

itself; e.g. authors, format and publishing date. One commonly used standard is *Dublin Core*[7], which defines labels (prefix DC:) to annotate the relation between two elements in documents or digital artefacts. For example, *ANONYMOUS1* and *ANONYMOUS2* are the authors of the document with the title *Conceptualised Document Saving in a Semantic Database*.

The network *structure* (starting at STRUCTURENODE) defines the logical structure of the document. It describes the order of text units (being chapter, section, subsection, paragraph, ...) and their content, which can be anything from pure text to images or tables. In Figure 3 this document is (partly) shown with two sections, which are defined by their title as well as content. Links between nodes for the strucutre describe dependencies like order or nested units.

The last network *concept* (starting at CONTENTNODE) represents the content of the document by concepts. The extracted concepts (see Section 2 and 2.4) are connected among one another, but also linked into externally defined ontologies to provide additional information about superior and subordinate concepts and, therewith, allow for a comprehensive understanding of the document [33, 34]. We have to point out that concepts can have three states: fully specified (the concept is known; e.g., part of an ontology), partly specified (further information (e.g., other concepts in the partial network) is required to evaluate the concept) and underspecified (unknown concept in this context).

Even though the partial networks are constructed independently from each other, links inbetween the structure and content network can encode further valuable information. That is, concepts are allocated to specific

[7] See `http://dublincore.org/documents/dces`

structural units like sections or tables and define, therefore, an order within the concepts.

3.2 Consistency of Networks

Document networks have to be validated against consistency constraints before being added to the repository to guarantee an overall quality without incomplete document information. Here, we look at three constraints: 1) existence of partial networks, 2) minimum number of information in each partial network and 3) valid implications for relations and inheritance of literals.

1. A document network is complete, if it has the two partial networks *content* and *meta-information*. Both networks are mutually dependent as concepts within the content require a context, which is specified by the meta-information. On the other hand, the meta-information is obsolete, if there is no further information about the content itself. The network about the structure is optional as documents not necessarily have a recognisable structure; i.e. short or flat documents.

2. In addition to the existence of partial networks, it is also required to have at least the elements in the network to identify the document and allow adding it to the Semantic Document Network. For the meta-information, this might be the title and author and at least one concept being part of an ontology for classification. Note that this depends on the kind of document. If the structural network is given, it must contain enough information to reconstruct the structure of the document.

3. The (semantic) network defines hierarchical relations; e.g., all elements associated with *Section 1.1* are also part of *Section 1*. Same has to be given, if two concepts are associated to an ontology; the relation between the two concepts must be valid taking the ontology hierarchical structure into account. The system has to perform semantic verification and detect inconsistencies while processing the document.

After a document network is created from the document, it is stored in a repository (here, a semantic network database) together with other documents. Documents share in this repository common nodes, i.e. to expose similarity and closeness. However, to prevent association of terms to the wrong document, we use a disambiguation method: all relations (edges connecting the nodes) are parameterised with the document identification allowing a later reconstruction of individual document networks.

4 Experimental Prototype

In this section, we describe the developed prototype, point out its dependencies to other frameworks and how we employed them for our research. The software, developed in Java, is separated into three components that do the preprocessing, execute the actual conceptualisation and manage the database communication. A more fine-grained explanation of this prototype can be found in [8].

4.1 Preprocessing

The preprocessing is performed in two subsequent steps: 1) transforming the received document into an internal model (with information already being associated with the corresponding partial network) and 2) annotating the content of the document.

JDOM: Due to the XML character of DocBook we utilise JDOM to access the body of the document. JDOM creates a Document Object Model (DOM) in form of a tree by parsing the document's elements.[8] Thereby, we can distinguish between structural, meta and content information and assign them accordingly to the partial networks.

ANNIE: The *General Architecture for Text Engineering framework* (GATE) offers a wide range of functionality required for natural language processing. GATE is offered as a stand-alone application as well as a library to be included in other applications [12].[9] Within GATE, there is a module called ANNIE ("A nearly new information extraction system") that encapsulates standardised Natural Language Processing functions similar to those illustrated in Section 2.1. This module is applied to annotate the content information extracted from the JDOM document parser.

4.2 Conceptualiser

As visualised in Section 2.4, the conceptualisation relies on the annotated text. With respect to such a prepared document, the first task of the conceptualiser is to retrieve appropriate terms; i.e. concept candidates.

Candidate Extraction: The annotations are accessable in sets of a given annotation type. These sets can be considered as layers in the annotation hierarchy. For our purpose we extracte one set containing all annotated sentences and one set containing every single token from the text. Additionally, we retrieve a set of recognised named entities possibly containing persons, locations or organisation.

[8] For details visit `http://www.jdom.org`.

[9] More information can be obtained at `http://gate.ac.uk`.

We process all sentences and iterate over the tokens in chronological order to sort out text expressions as a pre-stage of concept candidates. We ignore all tokens that belong to a named entity by comparing the annotation IDs. To extract text expressions we concatenate token values until a token either is a punctation mark or belongs to one of the stop words (see Section 2.4 for remarks on the stop word category). This information can be obtained from an annotation's feature map.

After all sentences are processed, the *Conceptualsier* evaluates the generated list of text expressions and unifies all that have an identical (case insensitive) string. Afterwards, these expressions are declared as concept candidates, which are used as queries in WordNet.

Accessing WordNet WordNet offers several APIs for access from external applications and programming languages.[10] We picked the Java WordNet Interface (JWI) from the MIT Computer Science and Artificial Intelligence Laboratory.[11]

The key element of the JWI is the `Dictionary` class. It communicates with a local WordNet installation and handles all requests; either for synsets or for relations among synsets.

Determining the Semantic Core With the WordNet connection at hand the Conceptualiser queries the meanings, i.e. synsets, for all concept candidates. Next, the declared relation definitions from WordNet are read for all synsets, from which the Lesk Overlap is later calculated.

According to the retrieved relations a `ConceptMeaningMatrix` is generated by the Conceptualiser that is used to determine a concept meaning's score. Finally, the semantic core is built from those concept candidates meanings that have the highest assigned aggregated semantic relatedness for each candidate.

Assembling Document Networks Based on the semantic network implemenation from the database the semantic core is consolidated with the other two partial networks. These are created separately by filling as many fields as possible concerning the Dublin Core element set (for the metadata network) and getting the structure information from the internal document data model. Figure 4 depicts the correlation of the partial document networks; see also Section 3.

[10] `http://wordnet.princeton.edu/wordnet/related-projects`.

[11] Visit `http://projects.csail.mit.edu/jwi` for more information.

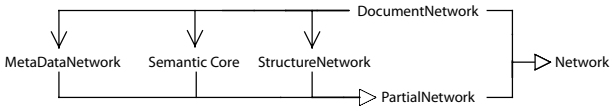

Figure 4: Document network class hierarchy

4.3 Semantic Database

After determining the semantic core and building the document network, it is stored in a semantic network database. This allow a storage of the semantic document networks without further transformation into an relational database model and improves the retrieval as well as the overlap with further documents in the database. We continue with an introduction in the principle of the semantic network database and afterwards describe how the database is integrated in our prototype.

4.3.1 Mode of Operation

The data model of the semantic network database does not use a defined and static schema, but simply represent information with two elements: nodes and edges. Nodes constitute atomic (unique) values, which implies that each node exists only once in the whole database. Edges are associated with exactly two nodes to embody a direct relation between the nodes. Furthermore, the edges are labeled.

With respect to performance and requirements regarding security and assurance of consistency, we decided to use the semantic network database developed (SND)) by [2]; but others like Neo4J or OWLIM could be used as well. SND is reduced to basic operations but therefore is fast in storing, extracting and searching content. SND allows only addition and deletion of nodes, but no updates. This is done to assure consistency as one has to remove all connections explicitly (assuming to imply up-to-dateness). Modification of notes and edges is done by deleting and adding an element. Both network elements offer further design variants meaning that nodes and edges may be under, partly or fully specified. These modelling options are useful when

1. the exact value is unknown but there is certainty about the existence of *a* value (under specified).

2. the exact value fulfils a *pattern* that can be expressed as logical expression (partly specified).

In addition to the degree of specification, when querying the database, edges may be defined, besides the standard search options, as optional or negated. Thereby, whole parts of the database's network can

be excluded from the search or are only respected if present.

To characterise information a data type is generally assigned. Within the semantic network database values cannot be assigned directly to a certain type. However, typification here always requires additional information; i.e. at least one node and a connceting edge (a descriptive relation). Thus, nodes implicitly may be an instance of different types (there is no mechanism to restrict the number of edges on a single node) which necessitates a way to guarantee particular properties. This can be done using *constraints*. These are specified as semantic networks themselves and utilise different specified network elements to ascertain the existence of other nodes and edges. Therefore, one or more edges are declared as *trigger* and define the responsibilty of a constraint network. When the trigger fires, the constraints claim the other elements and reject the network; in case that those are not available.

4.3.2 Database Layer

The assembled semantic document network is validated against a constraint network that matches the consistency conditions as established in Section 3.2. If not all required network elements are present, the document network will be rejected and the conceptualised document saving failed.

Otherwise, the document network is stored in the database by sending its entire set of edges to the semantic database. By definition, an edge is aware of its start and end node. Hence, it is ensured that no node will be omitted.

5 Conclusions

One of the key information and communication technologies in the future will be the interpretion of unstructured documents; that is the understanding of the meaning. The unlimited growth of the Web 2.0, where anyone can contribute, caused a proliferation of documents we are hardly able to deal with. Therefore, it is important to have *intelligent* systems being able to automate the processing, extraction of information and provision to users and also other systems. With respect to the manifold use of semantically processed documents, we decided not to focus on one application domain but rather develop a framework with focus on the storage of processed documents

- → to determine the best concept candidates using concept-based indexing;
- → to define a semantic core for the documents;
- → that generates a semantic document network for each document comprised of three partial networks: meta-information, structure and concept;
- → to store the processed documents in a semantic network database.

Where to go from here? On the one hand our ongoing research includes the improvement of the semantic document storage, but also the integration into domains such as automated essay grading and evaluation of machine translation. The *proof of concept* presented promising results as well as opened further opportunities for extensions and improvements. In this paper, we limited ourselves to a small example for demonstrative purposes, an extensive benchmark of the concept extraction and the semantic network database can be found in [8].

- → the exchangeability of the thesaurus,
- → supporting other languages than English,
- → enhancement of the semantic database,
- → anchoring concepts in the document's structure to preserve storylines.

The outcome of this study indicates that the semantic document network is already elegible to be incorporated in other domains and take advantage of the preprocessing. Our first field of application was the evaluation of machine translation. Instead of checking on n-grams, we determine the concepts within the documents and verify that both, the original document and the translation cover the same story. A practical use of this approach is the improvement of bots to facilitate greater interaction with users in virtual training environments [29] by understanding the meaning of user interactions and being able to respond appropriately.

References

[1] J. F. Allen. *Natural Language Understanding*. Benjamin/Cummings, Redwood City (CA, USA), 1995.

[2] T. Aust and M. Sarnow. Entwurf und Implementierung einer Medien-Datenbank-Middleware mit integrierten Semantischen Netzen [engl.: Design and implementation of a Media Database Middleware using Semantic Networks]. Master's thesis, Universitt Hamburg, 2009.

[3] S. Banerjee and T. Pedersen. *An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet*, volume 2276 of *Lecture Notes in Computer Science*, pages 136–145. Springer, Berlin, Heidelberg, 2010. In *Computational Linguistics and Intelligent Text Processing*.

[4] M. Baziz, M. Boughanem, N. Aussenac-Gilles, and C. Chrisment. Semantic cores for representing documents in ir. In 20^{th} *ACM Symposium on Applied Computing, SAC'2005*, pages 1011–1017, New York (NY, USA), 2005. ACM Press.

[5] M. Baziz, M. Boughanem, and S. Traboulsi. A concept-based approach for indexing documents in IR. In *Actes du XXIIIème Congrès INFORSID*, pages 489–504, 2005. Grenoble, Frankreich.

[6] S. Blohm, P. Cimiano, and E. Stemle. Harvesting relations from the web-quantiying the impact of filtering functions. In *Proc. 22nd Conference on Artificial Intelligence (AAAI)*, pages 1316–1323, 2007.

[7] F. Bodendorf. *Daten- und Wissenmanagement*. Springer, Berlin, Heidelberg, 1. edition, 2003.

[8] S. Boese. Entwicklung und Umsetzung einer konzeptualisierten Speicherung von Dokumenten in einer semantischen Datenbank [engl.: Design and Implementation of Conceptualized Document Storage in a Semantic Database]. Master's thesis, Universitt Hamburg, 2010.

[9] S. Boese, T. Reiners, and L. C. Wood. *Concept-based indexing in the design and construction of semantic document networks to support concept retrieval*. IGI Global, Hershey, U.S., 2013, under review.

[10] P. Buitelaar, P. Cimiano, and B. Magnini. *Ontology Learning from Text: An Overview*, pages 1–10. IOS Press, Amsterdam, 1. edition, 2005.

[11] C. Carpineto and G. Romano. *Concept Data Analysis*. Wiley, Sussex, England, 1. edition, 2004.

[12] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*, Philadelphia (PA, USA), 2002.

[13] H. Dreher. Automatic conceptual analysis for plagiarism detection. *Journal of Issues in Informing Science and Information Technology*, 4:601–614, 2007.

[14] C. Fellbaum. *WordNet – An Electronical Lexical Database*. MIT Press, Cambridge, MA, 1998.

[15] S. Ferré and S. Rudolph, editors. *Formal Concept Analysis*, volume 5548 of *Lecture Notes in Artificial Intelligence*, Berlin, Heidelberg, 2009. Springer.

[16] R. Florian, H. Hassan, A. Ittycheriah, H. Jing, N. Kambhatla, X. Luo, N. Nicolov, and S. Roukos. A Statistical Model for Multilingual Entity Detection and Tracking. Technical report, I.B.M. T.J. Watson Research Center, 2004. http://acl.ldc.upenn.edu/hlt-naacl2004/main/pdf/128_Paper.pdf.

[17] A. Harabagiu, C. A. Bejan, and P. Morărescu. Shallow semantics for relation extraction. In *Proceedings of the* 19^{th} *International Joint Conference on Artificial Intelligence (IJCAI '05)*, pages 1061–1066, 2005.

[18] D. He and J. Wang. *Cross-Language Information Retrieval*, pages 234–254. Wiley, Chichester, 2009.

[19] J.-P. Koenig, G. Mauner, B. Bienvenue, and K. Conklin. What with? The anatomy of a (proto)-role. *Journal of Semantics*, 25(2):175–220, 2008.

[20] T. K. Landauer, P. W. Foltz, and D. Laham. An introduction to latent semantic analysis. *Discourse Processes*, 25:259–284, 1998.

[21] T. K. Landauer, D. S. McNamara, S. Dennis, and W. Kintsch. *Handbook of Latent Semantic Analysis*. Lawrence Erlbaum Associates, Mahwah (NJ, USA), 1. edition, 2007.

[22] M. Lesk. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *SIGDOC '86: Proceedings of the* 5^{th} *annual international conference on Systems documentation*, pages 24–26, New York (NY, USA), 1986. ACM Press.

[23] D. I. Martin and M. W. Berry. *Mathematical Foundations Behind Latent Semantic Analysis*, pages 35–55. In [21], 1. edition, 2007.

[24] D. Maynard, K. Bontcheva, and H. Cunningham. Towards a semantic extraction of named entities. In *Proceedings of Recent Advances in Natural Language Processing (RANLP 2003)*. John Benjamins Publishing Company, 2003.

[25] A. Micarelli, F. Sciarrone, and M. Marinilli. *Web Document Modeling*, volume 4321 of *Lecture Notes in Computer Science*, pages 155–192. Springer, Berlin, 2007. In *The Adaptive Web: Methods and Strategies of Web Personalization*.

[26] G. A. Miller. *Nouns in WordNet*, pages 25–46. In [14], 1998.

[27] P. Pantel and M. Pennacchiotti. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 21st Inter. Conf. on Computational Linguistics (COLING) and the 44th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 113–120, 2006.

[28] J. S. Petöfi. Text representation and lexicon as semantic network. In *Logiche, calcoli, formalizzazioni e lingue storico-naturali*, Catania, Italien, 1976.

[29] T. Reiners and L. C. Wood. Immersive virtual environments to facilitate authentic education in logistics and supply chain management. In Y. Kats, editor, *Learning Management Systems and Instructional Design: Metrics, Standards, and Applications*. IGI Global, 2013.

[30] E. Reiterer, H. Dreher, and C. Guetl. Automatic concept retrieval with rubrico. In M. Schumann, L. M. Kolbe, M. H. Bretiner, and A. Frerichs, editors, *Anwendung der Konzeptanalyse und ontologische Modellierung in der Wirtschaftsinformatik, (MKWI 2010)*, pages 3–14, 2010.

[31] E. Riloff. Information extraction as a stepping stone toward story understanding. In A. Ram and K. Moorman, editors, *Understanding Language Understanding: Computational Models of Reading*. MIT Pr, Cambridge, MA, 1999.

[32] B. Rosenfeld and R. Feldman. Ures : an unsupervised web relation extraction system. In *Proceedings of the 21st International Conference on Computational Linguistics (COLING) and the 44th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2006.

[33] L. K. Schubert. *Semantic Nets Are in the Eye of the Beholder*, pages 95–108. In *The Morgan Kaufmann Series in representation and reasoning* [35], 1991.

[34] L. Shastri. *Why Semantic Networks?*, pages 109–136. In *The Morgan Kaufmann Series in representation and reasoning* [35], 1991.

[35] J. F. Sowa. *Principles of Semantic Networks – Explorations in the Representation of Knowledge*. The Morgan Kaufmann Series in representation and reasoning. Kaufmann, San Mateo (CA, USA), 1991.

[36] E. F. Tjong Kim Sang and F. De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In M. Daelemans, Walter; Osborne, editor, *Proceedings of the 7^{th} conference on Natural language learning at HLT-NAACL 2003*, pages 142–147, Morristown (NJ, USA), 2003. Association for Computational Linguistics.

[37] R. Wille. Restructuring lattice theory: An approach based on hierarchies of concepts. In Ferré and Rudolph [15], pages 314–339.

[38] R. Williams. The power of normalized word vectors for automatically grading essays. *Journal of Issues in Informing Science and Information Technology*, 3:721–728, 2006.

[39] D. Wimalasuriya and D. Dou. Ontology-based information extraction: An introduction and a survey of current approaches. *Journal of Information Science*, 36(3):306–323, 2010.