Text Condensation as Knowledge Base Abstraction*

Ulrich Reimer

University of Constance
Information Science Group
P.O. Box 5560
D-7750 Konstanz
F.R. Germany

Udo Hahn

University of Passau
Dept. of Math. & Computer Science
P.O. Box 2540
D-8390 Passau
F.R. Germany

ABSTRACT

A model of knowledge-based text condensation is presented which has been implemented as part of the text information system TOPIC. Two major processes are considered in detail, text parsing and text condensation with emphasis on the latter. Based on principles of semantic parsing, the text parser serves the purpose to augment the initial domain knowledge base with the knowledge encoded in a text, thus generating a specific text knowledge base. A condensation process then transforms these text representation structures into a more abstract thematic description of what the text is about, filtering out irrelevant knowledge structures and preserving only the most significant concepts. Finally, a hierarchical representation of the thematic units of the text is generated in terms of a text graph which supports variable degrees of abstraction for text summarization as well as content-oriented retrieval of text knowledge.

1. Introduction

The rapid advance of electronic text production and distribution technology has created a lot of enthusiasm with regard to the availability, ease of access, and dissemination of information contained in large electronic text files [17]. Unlike past generations of text-based information systems dealing exclusively with document *surrogates* such as abstracts, title headings, or keywords, they allow the immediate manipulation of source texts, i.e. *full*-texts such as letters, memos, minutes, magazine articles. The presumed potential inherent to these full-text databases has already been recognized and visionary (text) knowledge workbenches have been described that anticipate rather elaborated devices for in-depth document analysis (e.g., full-blown electronic encyclopedias [33] in terms of encyclopedic expert systems [21], or sophisticated question-answering facilities on top of inferential text knowledge bases [28]). However, the major thrust of development is still oriented towards appropriate *hardware support* (e.g., specialized text retrieval machines for full-text searching [15, 19], or appropriate mass storage media [30]), techniques for the *formal administration* of full-text databases (e.g., non-standard extensions of data-base systems and query languages [2, 1] or access methods [22, 7] to cover specific properties inherent to unformatted, textual data), and the provision of *user interfaces* which ease the interaction with full-text files and supply versatile tools for the manipulation of documents (e.g., linking document fragments, document versions, or critical annotations to documents in Hypertext environments [3]).

Methods for treating the *contents* of documents have been of much less concern, although the information retrieval problem -- identifying relevant information from large sets of document/data items -- is at least as crucial for electronic (storage) media as for standard print media. Three main approaches to the automatic content analysis of large expository full-texts (as opposed to document surrogates) can be distinguished: Simple *statistical* models involve frequency counts of document terms for indexing, document clustering based on term association factors for classification, and probabilistic relevance measures for retrieval [4]. This provides keyword-level analyses of texts on which reference retrieval mechanisms for documents are based. It is clearly inadequate, given the rich information potential inherent to full-text files. *Linguistic* approaches tend to favor structural aspects of text analysis which have proved useful in the acquisition of facts from linguistically constrained texts in limited domains [24] and the design of analogous retrieval mechanisms [10], but face serious problems when texts with complex semantic phenomena (ambiguities, paraphrase and inference relations, etc.) have to be processed. While the focus of *knowledge-based* systems, in particular, has been on these semantic issues of text understanding they usually concentrate on *methodological* problems, e.g., specialized parsing and learning strategies, memory structures (for an overview, cf. [6, 18]), but often neglect equally sophisticated *functional* features in an integrated system architecture (some notable exceptation is made in CyFr [26]).

Starting from a knowledge-based approach to full-text analysis, too, our efforts nevertheless have tried to avoid these shortcomings. The design of our system was strongly influenced by *functional* considerations, namely to supply various coherent levels of information related to the original text, each corresponding to different information requirements: *reference* to a subset of relevant documents, access to specific *facts*, display of *significant passages*, or provision of *synoptic overviews* of the source text on various levels of specificity such that these retrieval options should complement each other.

In the exposition to follow, corresponding methodological devices of the text condensation system TOPIC [13, 14] are described. Its *text parsing* component incorporates a lexically distributed text grammar [11] in the format of word experts [29] and a description of domain-specific background knowledge in terms of a frame representation model [23]. Text parsing is realized as a process that transforms the initial frame knowledge base into a knowledge base that augments the initial one by the specific knowledge encoded in the text under analysis (Sec.2). Starting from these text knowledge structures, the process of *text condensation* transforms them into a more abstract (condensed) representation which comprises only the most significant thematic aspects of a text (Sec.3). The resulting representation structures describe the topics a text deals with. Iterated condensation steps additionally provide more generic topic descriptions, thus leading to a unified conceptual hierarchy of thematic descriptions in terms of a text graph.

Now, what is this all good for? A text information system which consists of a collection of full-texts together with a description of their contents by associated text graphs supports different types of target information structures: besides *summarizing* texts at varying levels of thematic abstraction, access to significant *passages* of the source text as well as access to the *factual knowledge* acquired from a text during its analysis is supported by the text graph structure. Due to the hierarchic nature of a text graph, a text appears to be divided into a more or less fine-grained description of different topics. The choice of an appropriate entry point to a text graph varies dynamically according to the level of generality underlying a given query. In this way, linear document structure as enforced by conventional print media is overcome by offering a dynamic assembly of relevant informational units -- each on the appropriate level of explicitness -- according to the current user needs. Obviously, the text representation provided by a text graph is a step towards the *automatic* creation of *hypertexts* [3] as used in an intelligent information retrieval environment. In fact, the availability of full-texts in electronic files turns out to be a technical prerequisite of more flexible text and knowledge retrieval options. Their sophistication clearly out-performs those applicable to printed text, and thus constitutes progress towards text-based information management through the application of artificial intelligence techniques.

In this paper, we concentrate on the methodological issues underlying the generation of text graphs in TOPIC[1], i.e. the transformation of the knowledge base that results from text parsing to a representation format which provides a description of the various topics a text deals with at variable levels of detail. The following section gives an outline of the principles underlying the text parsing component. This is necessary in order to indicate the effects various text phenomena have on the structure of the knowledge base. Subsequently, the process of text condensation is described as the transformation of knowledge structures of the text parse into more abstract condensate structures (Sec.3).

2. Text Parsing

Building up on principles of integrated parsing [27] and lexical distribution of linguistic knowledge [29] a distributed conceptual parser has been devised in the TOPIC system. It considers the interaction of (traditionally speaking) nominals in text, i.e. nouns, adjectives, and their combination in noun phrases including text-specific phenomena, like nominal anaphora and ellipsis. Since, in particular, it does not account for the role of verbs, the parser's understanding capabilities are restricted to the recognition of semantic and thematic relationships in texts on a terminological (taxonomic) level of knowledge representation. However, the limitations inherent to this partial text parsing approach have been purposefully balanced with the requirements to provide abstracts on an indicative level of text condensation, i.e. characterizing the *aboutness* of a text [16]. This is achieved, basically, by relating each occurrence of a token in the text that matches a corresponding concept in the frame knowledge base to a set of primitive knowledge base operations (incrementation of activation weights, assignment of properties, etc.). The selection of an appropriate sequence of operations is determined by a system of word experts that investigate the particular functional role this token plays with respect to its textual environment (simple cases of lexical cohesion, anaphora, ellipsis, co-ordination, etc.). Parsing a text thus amounts to augmenting a previously given knowledge base that contains the domain-specific knowledge by text-specific information.

Applying a distributed conceptual parser to full-text material, the following cases are most likely to occur (concepts of the knowledge base are in bold face, the parser's current position is indicated by underlining):

a) Immediate Reference to a Domain Concept

Text *The **Zenon-X** has turned out to be a great success.*

The current text token refers to a frame in the knowledge base without any conceptual relation to adjacent (in particular: preceding) text tokens. This effects the incrementation of that frame's activation weight -- an appropriate counter is indicated by the Σ sign attached to each frame (in bold letters), slot (enclosed by angular brackets), and slot filler (enclosed in curly brackets) -- unless case d) applies:

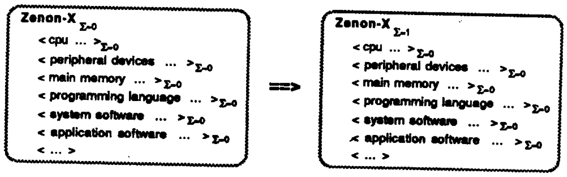

Zenon-X $_{\Sigma=0}$
< cpu ... >$_{\Sigma=0}$
< peripheral devices ... >$_{\Sigma=0}$
< main memory ... >$_{\Sigma=0}$
< programming language ... >$_{\Sigma=0}$
< system software ... >$_{\Sigma=0}$
< application software ... >$_{\Sigma=0}$
< ... >
==>
Zenon-X $_{\Sigma=1}$
< cpu ... >$_{\Sigma=0}$
< peripheral devices ... >$_{\Sigma=0}$
< main memory ... >$_{\Sigma=0}$
< programming language ... >$_{\Sigma=0}$
< system software ... >$_{\Sigma=0}$
< application software ... >$_{\Sigma=0}$
< ... >

Fig_1a State Transition in the Knowledge Base Corresponding to the Effects of Immediate Reference to a Dc:main Concept

b) Immediate Reference to a Property Class of a Domain Concept

Text *The **Zenon-X** has turned out to be a great success. Although being equipped with a rather slow **cpu** its major virtues ...*

The current text token refers to a slot of a frame in the knowledge base which has already been introduced in the text previously (*elliptical lexical cohesion*). This causes the activation weights of the slot and its associated frame to be incremented:

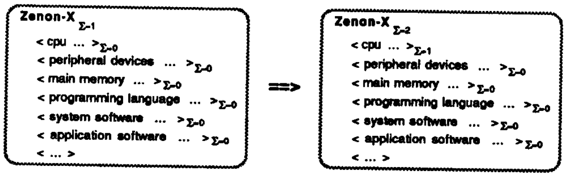

Zenon-X $_{\Sigma=1}$
< cpu ... >$_{\Sigma=0}$
< peripheral devices ... >$_{\Sigma=0}$
< main memory ... >$_{\Sigma=0}$
< programming language ... >$_{\Sigma=0}$
< system software ... >$_{\Sigma=0}$
< application software ... >$_{\Sigma=0}$
< ... >
==>
Zenon-X $_{\Sigma=2}$
< cpu ... >$_{\Sigma=1}$
< peripheral devices ... >$_{\Sigma=0}$
< main memory ... >$_{\Sigma=0}$
< programming language ... >$_{\Sigma=0}$
< system software ... >$_{\Sigma=0}$
< application software ... >$_{\Sigma=0}$
< ... >

Fig_1b State Transition in the Knowledge Base Corresponding to the Effects of Immediate Reference to a Property Class of a Domain Concept

c) Immediate Reference to Permitted/Actual Properties of a Domain Concept

Text *The **Zenon-X** runs **Turbo-PROLOG** under **Xenix**.*

The current text token references an expression which (according to some integrity constraint) is a permitted or actual slot filler of a frame that already has been introduced in the text previously (*simple lexical cohesion*). The expression may either denote another frame in the knowledge base (as is the case with ***Xenix*** in our example) or a so-called terminal slot filler which may characterize numerical values of units (e.g. ***1 MB*** of ***main memory***), adjectives (e.g., the ***European market***), etc. With respect to slot filling two cases then have to be distinguished: if not present as an actual slot filler, the current expression is assigned as slot filler to the appropriate slot of that frame (assignment of a *permitted* property), *and* the activation weights of the slot filler, slot and its associated frame get incremented simultaneously (see Fig_1c below); if, on the other hand, the expression is already present as an *actual* property (slot filler), only the incrementation of activation weights is carried out:

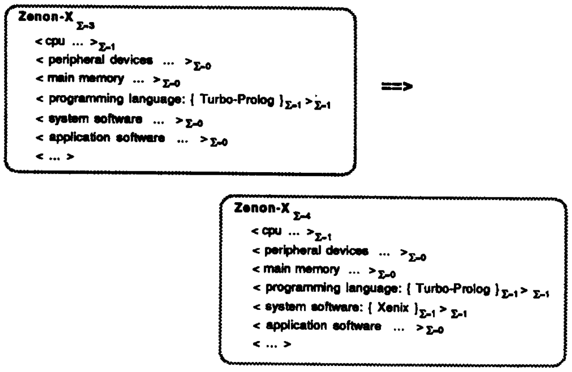

Zenon-X $_{\Sigma-3}$
< cpu ... >$_{\Sigma-1}$
< peripheral devices ... >$_{\Sigma-0}$
< main memory ... >$_{\Sigma-0}$
< programming language: { Turbo-Prolog }$_{\Sigma-1}$ >$_{\Sigma-1}$
< system software ... >$_{\Sigma-0}$
< application software ... >$_{\Sigma-0}$
< ... >
==>
Zenon-X $_{\Sigma-4}$
< cpu ... >$_{\Sigma-1}$
< peripheral devices ... >$_{\Sigma-0}$
< main memory ... >$_{\Sigma-0}$
< programming language: { Turbo-Prolog }$_{\Sigma-1}$ >$_{\Sigma-1}$
< system software: { Xenix }$_{\Sigma-1}$ >$_{\Sigma-1}$
< application software ... >$_{\Sigma-0}$
< ... >

Fig_1c State Transition in the Knowledge Base Corresponding to the Effects of Immediate Reference to Permitted Properties of a Domain Concept

As can already be seen from the examples, the parser easily adapts to a particular kind of text coherence pattern, namely the continuous thematic progression based on lexical cohesion properties of the concepts involved ([11] for more details). However, the close correspondence which, so far, has been assumed to hold between the occurrence of a text token and the proper designation of a corresponding concept in the world knowledge base, is quite misleading with respect to anaphoric regularities holding on the *text* level of linguistic analysis. Consider the case of

d) Mediated Reference to a Domain Concept

Text *Actually, the* ***Zenon-X*** *has turned out to be a great success on the* ***U.S.*** *market. The* ***machine's*** *distribution in* ***Europe****, however, is currently hampered by ...*

The current text token literally references a frame in the knowledge base, though actually it refers to a subordinate of this frame which has already been introduced in the text previously (*nominal anaphora*). This well-known text phenomenon is tackled by pretending that the more specific frame has occurred. This means incrementing the activation weight of the more specific frame (in our example **Zenon-X** instead of **machine** the latter being a superordinate of the former) in order to preserve correct focus indications (analogous steps have to be considered for mediated references to property classes or permitted/actual properties of a domain concept):

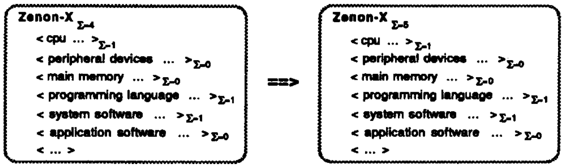

Zenon-X $_{\Sigma-4}$
< cpu ... >$_{\Sigma-1}$
< peripheral devices ... >$_{\Sigma-0}$
< main memory ... >$_{\Sigma-0}$
< programming language ... >$_{\Sigma-1}$
< system software ... >$_{\Sigma-1}$
< application software ... >$_{\Sigma-0}$
< ... >
==>
Zenon-X $_{\Sigma-5}$
< cpu ... >$_{\Sigma-1}$
< peripheral devices ... >$_{\Sigma-0}$
< main memory ... >$_{\Sigma-0}$
< programming language ... >$_{\Sigma-1}$
< system software ... >$_{\Sigma-1}$
< application software ... >$_{\Sigma-0}$
< ... >

Fig_1d State Transition in the Knowledge Base Corresponding to the Effects of Mediated Reference to a Domain Concept

We skip here more complicated details of linguistic aspects of text analysis relating, for instance, to local interruptions (e.g., through quantification or negation of domain concepts) or redirections of cohesion development (e.g., as indicated by various types of conjunctions and other cue words for topical shifts), although appropriate provisions have been made in TOPIC to account for them. Instead, we emphasize that for the purpose of text condensation *incrementation of activation weights* and *property assignment* (slot filling) have to be considered as the basic operations issued from the text parser to the frame knowledge base to create an adequate representation of the text's contents. The knowledge structures built up by the text parser this way form a text knowledge base and are taken as the starting point of the text condensation process to be described in the remainder of this paper.

3. Text Condensation

The activation weights and slot filling patterns of single frames as well as particular connectivity patterns holding within a group of frames in the text knowledge base provide the basis for the construction of a thematic description by the text condensation process. Only the most significant concepts are included, while the knowledge structures that are irrelevant with respect to an overall thematic characterization have to be filtered out. Thus, text condensation can be conceived as a kind of *abstraction* process on (text) knowledge bases.

Parsing the complete text, i.e. continuously elaborating the representation structures of the frame knowledge base through the basic operations mentioned above, and then starting the condensation process only at the very end of that text would result in a serious loss of organizational structure a text implicitly exhibits through the way its topics are elaborated. At the other extreme, determining condensation patterns at every instance of a concept occurring in the text, would cause an unnecessary computation overhead, since major topical movements are unlikely to be tied to a single domain-specific concept. Instead, we have decided on a quite pragmatic approach to initialize the condensation procedures, trading off between computation efforts and the proper recognition of textual macro structures. It is based on the observation that in the sublanguage domain we are currently working in, topic movements occur predominantly at paragraph boundaries. Therefore, text condensation procedures are started at the end of each paragraph so that thematic overlaps as well as topic shifts and breaks between adjacent paragraphs can be identified, and the extension of a topic be exactly delimited.

The abstraction mechanisms inherent to the text condensation process produce aggregated text knowledge structures, so-called *topic descriptions*, which represent the topics of the paragraph under analysis (Sec.3.1). Combining the topic descriptions of the whole text yields a coherent representation of its topical structure in terms of a *text graph* (Sec.3.2).

3.1 Generating Topic Descriptions

A thematic characterization of a paragraph is generated from the activation weights as well as the slot filling and connectivity patterns in the text knowledge base. In a first condensation step those concepts are determined which are significantly salient, while in a second one these salient concepts are re-combined to form the topic description of a thematically coherent part of the text.

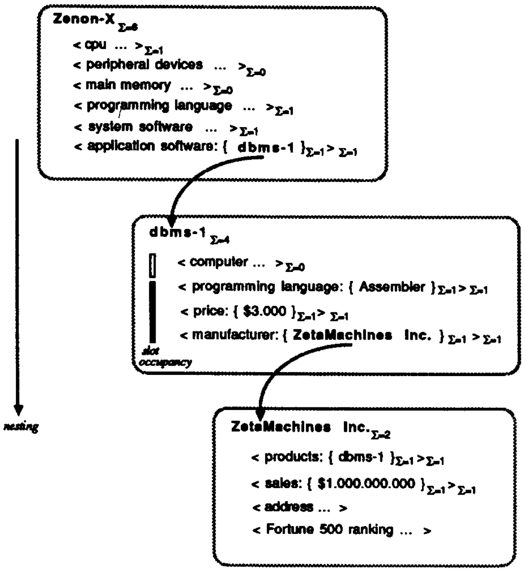
The Determination of Dominant Concepts. A major criterion for determining the salience of a concept is the frequency of its explicit *and* implicit mention in the text as recognized by the text parsing process. Since the resulting activation weights in the knowledge base are independent from linguistic surface phenomena (cf. the case of nominal anaphora in Sec.2), and since they also reflect the influence of a variety of conceptual relations (e.g., related to lexical cohesion regularities, ibid.), the **activation weights** attached to the knowledge structures can directly be used as relevance indicators for associated concepts[2]. Those of the active concepts that are salient enough to take part in a topic description will be called *dominant concepts*.

We distinguish between dominant slot fillers, dominant slots, dominant frames, and dominant clusters of frames. In order to preserve the correct interpretation of their contextual meaning a dominant slot is represented as a linear graph whose nodes are labelled by the corresponding frame and slot name and whose vertex is labelled by an appropriate type symbol (s for slot), i.e. *frame--s--slot*; accordingly, dominant slot fillers are characterized by *frame--s--slot--sf--filler* where sf stands for slot filler.

First, we give a fairly specific measure to identify ***dominant slot fillers*** by comparing the activation weight of a particular slot filler with the average activation weight over all active slot fillers in the text knowledge base. Let *Active_Fillers* denote the set of all slot fillers in a text knowledge base whose activation weight is greater than zero, and *sf_weight(f,s,sf)* characterize the activation weight of slot filler *sf* in slot *s* of frame *f*. If for some frame, slot, and slot filler we have

$$\sum_{i} \sum_{j} \sum_{k} \text{sf\_weight}(\text{frame}_i, \text{slot}_j, \text{filler}_k) \,/\, |\text{Active\_Fillers}| \leq \text{sf\_weight}(\text{frame}, \text{slot}, \text{filler})$$

then the graph *frame--s--slot--sf--filler* constitutes a part of the current topic description. Similarly, ***dominant slots*** and ***dominant frames*** are determined by examining if their activation weight exceeds the average weight of slots and frames, resp. (for more technical details, cf. [14]). In general, dominance of concepts is here related to the **activation level** of the whole knowledge base.



Zenon-X Σ=6
< cpu ... >Σ=1
< peripheral devices ... >Σ=0
< main memory ... >Σ=0
< programming language ... >Σ=1
< system software ... >Σ=1
< application software: { dbms-1 }Σ=1 > Σ=1
dbms-1 Σ=4
< computer ... >Σ=0
< programming language: { Assembler }Σ=1 > Σ=1
< price: { $3.000 }Σ=1 > Σ=1
< manufacturer: { ZetaMachines Inc. }Σ=1 > Σ=1
slot occupancy
nesting
ZetaMachines Inc. Σ=2
< products: { dbms-1 }Σ=1 >Σ=1
< sales: { $1.000.000.000 }Σ=1 >Σ=1
< address ... >
< Fortune 500 ranking ... >

Fig_2 Illustration of Significant Degrees of Slot Occupancy and Depth of Nesting of Slot Fillers

Another group of dominance measures considers the impact of certain **slot filling patterns** on the identification of salient concepts. As a consequence, the following criterion relies more closely upon structural properties of frame representation languages than the one just considered, since it relates to the elaboration of a slot filler in terms of **slot occupancy** and the **depth of nesting** of slot fillers. For example, consider in Fig_2 *dbms-1* as a slot filler of the *application software* slot of the *Zenon-X* frame. The *programming language*, *price*, and *manufacturer* slots of the *dbms-1* frame all are active and have been filled appropriately. Nesting is further enforced, since special treatment has been given to the *manufacturer* slot of *dbms-1* as its slot filler *ZetaMachines Inc.* itself has received further slot fillers (concerning *products*, *sales*, etc.). Accordingly, a slot is taken as a ***dominant slot*** if a frame is assigned to it as a slot filler such that the majority of its slots have been filled, too (significant degree of slot occupancy), or if a slot filler exists which is further elaborated in more detail (significant degree of nesting of slot fillers) -- an analogous criterion can be established for that slot filler making it a ***dominant slot filler***. In either of these cases the graph *frame--s--slot[--sf--filler]* (in Fig_2: *Zenon-X--s--application software[--sf--dbms-1]*) constitutes a part of the current topic description. Note that it seems unreasonable in this particular case to *over*-generalize, and maintain that the corresponding frame (*Zenon-X* in the example below) is dominant, too (although sometimes use is made of these implicit dominance relations).

Besides weighting and slot filling criteria on which dominance computation is based, we investigated the role of **connectivity patterns** based on generalization hierarchies of frames. A significant number of active frames which have a common superordinate frame may constitute a *dominant cluster of frames*. The cluster is represented by the common superordinate which will be called a *cluster frame* (when identified it forms a defective linear graph composed only of a single node which is labelled by the corresponding frame name). Since the cluster frame does not need to be active and even need not be mentioned explicitly in the text, this type of dominance measure introduces a further independence of the condensation process from the terminology used in a text (besides phenomena already covered by the text parser).

The following variant, introduced to illustrate the determination of ***cluster frames***, has to take the following trade-off into account: on the one hand the common superordinate should be as specific as possible (raising recall), while on the other hand a more generic superordinate stands for a larger cluster and thus more frequently turns out to be a relevant (though less specific) topic. The solution we have devised is outlined in the following algorithm (Fig_3). It captures the heuristics to start from the most general concepts contained in the knowledge base (so-called 'top frames') and descend the generalization hierarchy downwards as long as no *significant* loss of active concepts occurs . Significant loss in the upper level of the generalization hierarchy is defined relative to an empirically justified cut-off value for active concepts (lines 3 and 7) while on the lower levels (line 11) no further loss of active concepts is permitted:

```
[01]  for all frames without a superordinate (top frames) do
[02]   begin
[03]     if number of active subordinates of the current top frame
               > cut-off value then
[04]     begin
[05]       for all immediate subordinates of the current top frame do
[06]        begin
[07]          if number of active subordinates of the current subordinate
                    > cut-off value then
[08]          begin
[09]             repeat
[10]                determine that immediate subordinate of the current subordinate
                    whose number of active subordinates is maximal and
                    call it SUB$_{max}$;
[11]                if the number of active subordinates of the current subordinate
                    is equal to the number of active subordinates of SUB$_{max}$ then
[12]                   let SUB$_{max}$ denote the current subordinate;
[13]             until further descent causes a loss of active subordinates;
[14]             the current subordinate is cluster frame;
[15]          end;
[16]        end;
[17]       if no cluster frame is found then
[18]          the current top frame is cluster frame;
[19]     end;
[20]  end.
```

Fig_3 Algorithm for Knowledge-Based Clustering

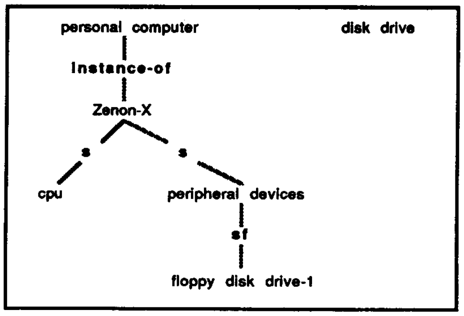
The cluster algorithm above gives a flavor of the power of *knowledge-based clustering*. Unlike standard clustering procedures which are solely based on the exploitation of plain co-occurrence characteristics of terms (cf., e.g., [25]), this algorithm takes two additional kinds of information into account which improve the significance of its results and its efficiency drastically. First, it relies on the existence of *semantically* distinguished types of relational links which makes possible to suppress irrelevant relationships between concepts (in the example above only concept specialization is considered). Additionally, sophisticated use of *heuristics* which prune the search space eliminates the problem of combinatorial explosion which would arise if an unrestricted search in the text knowledge base is conducted, trying to find all maximal subsets of active frames which have a common subordinate frame by successively checking all possible subsets (i.e. considering all elements of the corresponding power set!). We advocate this approach as an alternative to the one suggested by Taylor [31] and Lehnert [20] in that it is not based on *general* graph characteristics for computing clusters in some arbitrary net representation, but heavily exploits the *semantics* of the various representation constructs [23] used to represent the result of a text analysis.

Relating Dominant Concepts to Yield Topic Descriptions. The dominance measures determine a collection of formally unrelated dominant concepts as linear graphs. In the next step of the condensation process these separate graphs have to be combined to form a compound *topic description*. The combination is performed by simply overlapping identical nodes of the same type -- frame (dominant frames and cluster frames are treated as one type), slot, or slot filler -- occurring in different dominance descriptions and adding specialization links between frames where possible (see the example given in Fig_4 below). By virtue of the parsing process, which provides the input to the condensation procedure, the resulting *semantic net* can be interpreted in the following way. A simple occurrence of a frame which remains unconnected to some slot link (as with *disk drive* in Fig_4) means that the associated text passage deals with that concept in a fairly general way. Otherwise, the more elaborated the structure of a topic description, the greater the thematic specificity of the related concepts (cf., e.g., *Zenon-X*--s--*peripheral devices*--sf--*floppy disk drive-1* in Fig_4). Finally, if a slot (filler) name equals the name of a frame in the knowledge base and that frame does not occur as either a dominant frame or a cluster frame the corresponding concept is only relevant with respect to the frame where it is associated to as a slot (filler). So it stands for a rather specialized thematic aspect, but is of only minor importance to the overall topical structure of the text, thus making it a primary candidate for elimination by later abstraction procedures.

Example. The following dominance descriptions are given:

dominant entries:	*Zenon-X*--s--*peripheral devices*--sf--*floppy disk drive-1*
dominant slot:	*Zenon-X*--s--*peripheral devices* *Zenon-X*--s--*cpu*
dominant (cluster) frames:	*Zenon-X* *personal computer* *disk drive*

Their combination yields the following topic description to which a corresponding verbalization[3] is attached. As can be seen from the example below, descending the graph of a topic description leads to more specific thematic aspects:



"The text passage is about personal computers. The Zenon-X is discussed in more detail with respect to its cpu and its peripheral devices. Besides disk drives in general, the floppy disk drive which is available for the Zenon-X is focused on."

Fig_4 An Illustration of a Topic Description and Its Corresponding Verbalization

Topic descriptions are computed for every paragraph. If the descriptions of two adjacent paragraphs match they are merged to form a compound thematic unit. Determining if two topic descriptions match is straightforward: discarding slot fillers from their semantic net representations (thus generalizing them) two topic descriptions match if these reduced semantic nets are identical or if one of them is a subgraph of the other.

Example. The following two topic descriptions match because taking away the slot fillers makes the second one a subgraph of the first graph (the matching area, i.e. **Zenon-X**--**s**--**peripheral devices**, is indicated by shadow fonts):

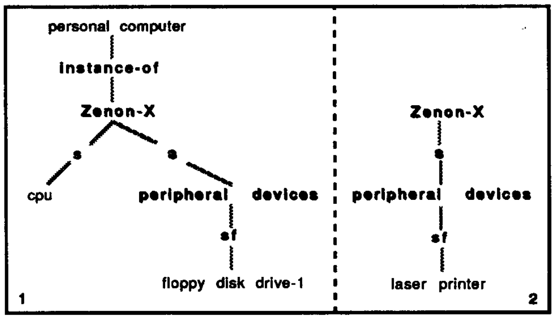


Fig_5 Matching of Topic Descriptions

The thematic descriptions of adjacent text passages are merged as long as the current topic description matches that of the preceding paragraph(s). If they do not match, the preceding passage(s) are delimited as a separate, thematically coherent text constituent, and a new constituent will be set up at the current paragraph. In technical terms, a *text constituent* is consists of:

- the topic description which provides an *abstracted* view on the contents of the (sequence of) text passage(s) making up that particular text constituent (essential for *summaries*)
- a link to the associated (sequence of) text passage(s) in the *full-text* data base (essential for *passage retrieval*)
- the text knowledge base which characterizes the *original* result of text parsing (essential for accessing *facts* acquired through text analysis)

3.2 The Text Graph

The topic descriptions whose generation has been discussed in the preceding section already exhibit a hierarchical structure. The idea underlying text graph construction now focuses on the fact that from a topic description *as a whole* more generic descriptions can be derived. Thus, the construction of a text graph proceeds from the examination of every pair of topic descriptions and takes their commonalities as a more generic thematic characterization. Applying this procedure as many times as possible (taking also the newly generated topic abstractions into account) a hierarchy of topic descriptions evolves which is represented in terms of a *text graph* (see Fig_6). The most specific descriptions (they correspond to the text constituents) form the leaf nodes of the text graph. The generalized topic descriptions constitute its non-leaf nodes. Given such a text graph a text appears to be divided into a variable number of thematically coherent units. The degree of specificity or generality of their thematic description can dynamically be chosen by the user depending on her/his current information needs. The root nodes offer the most abstract characterization of the associated text as a whole (since a text graph is poly-hierarchic it may have more than one root). By descending a text graph, more and more specific topic descriptions are encountered until the most specific ones, finally, allow access to the facts acquired from the text, or, alternatively, to original text passages of the source text.

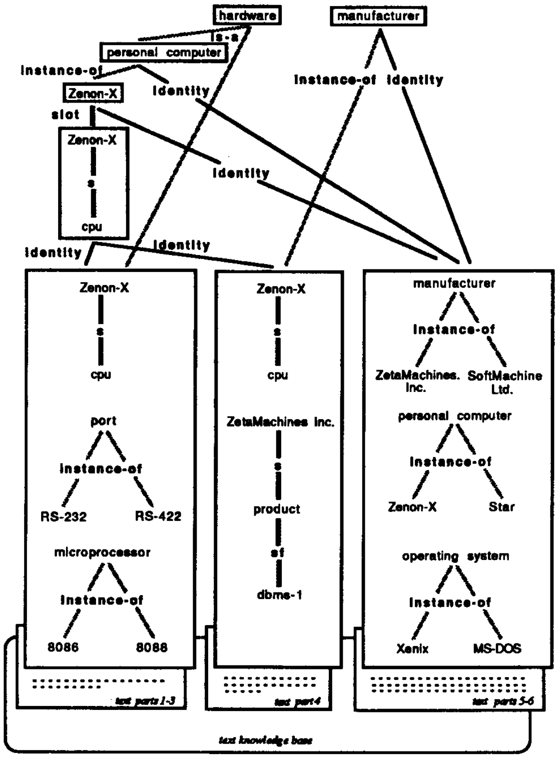


Fig_6 A Text Graph Fragment (derivable links have been omitted)

The following kinds of *abstraction relations* between text graph nodes can occur (see Fig_6):

- an **identity** link goes from a node n to a superordinated node n′ if node n′ consists of a graph that occurs in node n, or if node n′ consists of a single node which occurs in node n as part of a concept specialization graph (using *is-a* or *instance-of* links) - and only if one of these cases holds;
- an **is-a (instance-of)** link goes from a node n to a superordinated node n′ iff n′ consists of a graph whose only element is an *is-a* (*instance-of*) superordinate of the top-most element of a graph that occurs in node n;
- a **slot** link goes from a node n to a superordinated node n′ iff node n′ consists of a graph whose only element occurs top-most in a graph inside node n where a slot element is subordinated to it.

Unlike competing knowledge-based approaches to text summarization, our method allows *dynamic* construction of text summaries, instead of constraining it by *static*, built-in representation facilities to an entirely fixed level of detail (as with sketchy scipts in the FRUMP system [5]). In addition, it neither depends on interest profiles [8] nor on importance rating rules [9] given *prior* to text analysis. Due to the rich internal structure of the topic descriptions being generated, the selection of interesting topics can be postponed until actually *querying* the text knowledge base. This, obviously, offers much more *flexibility* than biasing the text analysis process by a priori filters which are likely to change from user to user.

4. Conclusion

We have outlined a model of knowledge-based text condensation which builds upon the knowledge structures generated by a semantic text parser. The result of text condensation is a text graph which allows flexible, content-oriented access to full-text information. This way, on-line texts can be dealt with as hypertexts offering a dynamic, query-time assembly of informational units just as considered relevant by the user. (S)he may choose among access to relevant *facts*, the provision of *topical descriptions* on different abstraction levels, and finally, the retrieval of *significant passages* of the original text. The latter yields authentic information unavailable in the text knowledge base due to the limited understanding capacity of the text parser. The variety of target information structures in the TOPIC system shows some similarity in performance with prototype electronic encyclopedia systems [33] or inferential text knowledge bases [28]. The major difference, however, (using terminology from Weyer & Borning) lies in the *automatic* generation of semantically rich world models and the corresponding supply of *conceptual* navigation criteria and knowledge-based information filters.

Finally, one should stress the methodological generality of *text graphs* as a powerful index structure for KBMS, not only limited to the text and knowledge retrieval options outlined so far. We are currently planning to use this device as methodological backbone for an *office document management system*, thus adding intelligent mail and message functions to it. Further extensions are on the way to relate text graphs to special requirements arising in the area of *text-based knowledge acquisition* in expert system environments [12], e.g., by adding KBMS facilities to handle conflicting and time-dependent knowledge as well as stringent features for version control. The most serious limitation deserving further investigation concerns conceptual relations holding between the nodes of text graphs of *different* texts and the provision of uniform access mechanisms to these text graphs. Extending the notion of a text graph as described in Sec.3.2, we intend to construct a corresponding *hyper textgraph* facility in order to supply a uniform conceptual view on the knowledge structures of full-text databases.

Footnotes

[*] The work reported in this paper was supported by *Bundesministerium für Forschung und Technologie (BMFT) / Gesellschaft für Information und Dokumentation (GID)* under contract 1020016 0.

[1] TOPIC which currently runs on various UNIX™ machines is written in C. Our experimental experience is based on the analysis of about 20 full-texts (each composed of approx. 2000-3000 text tokens). The distributed conceptual parser contains approx. 60 functionally decomposed grammar modules, while the size of the initial frame knowledge base varies between 150-200 frames. It incrementally gets augmented through new concepts learned as a by-product of text parsing. On the average, each of the 20 text knowledge bases generated so far consists of 1/4 of the items constituting the initial domain knowledge base which underlies the text analysis program (plus extensions coming from each particular text through concept learning).

[2] In order to rule out interferences between two subsequent paragraphs all activation weights in the text knowledge base related to the analysis of the previous paragraph are zeroed before continuing the analysis of the next paragraph. However, to avoid losing conceptual reference points related to foregoing text an appropriate focus indicator is provided in the parsing memory.

[3] TOPIC does not have natural language generation facilities. Instead, an experimental *graphical* interface [THIEL/HAMMWÖHNER 1986] is supplied to (hopefully) gain more flexibility (through browsing, zooming, and further direct navigation operators) than might be available from *natural language* summaries alone.

REFERENCES

[1] F. BANCILHON and P. RICHARD, "Managing Texts and Facts in a Mixed Data Base Environment," G. Gardarin & E. Gelenbe (eds), *New Applications of Data Bases*, London, Academic P., pp.87-107, 1984.

[2] H. BILLER, "On the Architecture of a System Integrating Data Base Management and Information Retrieval," G. Salton and H.-J. Schneider (eds), *Research and Development in Information Retrieval*, Berlin, Springer, pp.80- 97, 1983.

[3] J. CONKLIN, "Hypertext: An Introduction and Survey," *Computer*, vol. 20., no.9, pp.17-41, 1987.

[4] W.B. CROFT, "*TESS: An Effective Text Storage and Search System.*," Comp. and Inf. Sci. Dept., Univ. of Massachusetts at Amherst, COINS Tech Report 83-06, 1983.

[5] G. DeJONG, "*Skimming Stories in Real Time: An Experiment in Integrated Understanding*," Yale Univ., Ph.D.Diss., 1979.

[6] G. DeJONG, "Artificial Intelligence Implications for Information Retrieval," *Proc. 6th Ann. Int. ACM SIGIR Conf on Res. and Devel. in Information Retrieval*. Bethesda, Maryland, June 6-8, 1983, pp.10-17.

[7] C. FALOUTSOS and S. CHRISTODOULAKIS, "Access Methods for Documents," D. Tsichritzis (ed) *Office Automation*,. Berlin, Springer, pp.317-338, 1985.

[8] D. FUM *et al.*, "Forward and Backward Reasoning in Automatic Abstracting," *COLING 82: Proc. 9th Int. Conf. on Computational Linguistics*, Prague, July 5-10, 1982, Prague, Academia, pp.83-88.

[9] D. FUM *et al.*, "Evaluating Importance: A Step towards Text Summarization," *IJCAI 85: Proc. 9th Int. Joint Conf. on Artificial Intelligence*, 18-23 Aug. 1985, Los Angeles, Ca., Los Altos/CA, Morgan Kaufmann, pp.840-844, 1985.

[10] R. GRISHMAN and L. HIRSCHMAN, "Question Answering from Natural Language Medical Data Bases," *Artificial Intelligence* vol.11, pp.25-43, 1978.

[11] U. HAHN, "A Generalized Word Expert Model of Lexically Distributed Text Parsing," B. du Boulay *et al.* (eds) *Advances in Artificial Intelligence - II. 7th Europ. Conf. on Artificial Intelligence: ECAI-86*. Brighton, U.K., July 20-25, 1986, Amsterdam, North-Holland, pp. 417-425, 1987.

[12] U. HAHN, "Modeling Text Understanding: The Methodological Aspects of Automatic Acquisition of Knowledge through Text Analysis," *Proc. 1st Int. Symp. on Artificial Intelligence and Expert Systems. Part A: Theoretical Foundations and Research Projects in Artificial Intelligence*. Berlin, May 18-20, 1987, AMK Berlin, pp.167- 219.

[13] U. HAHN and U. REIMER, "TOPIC Essentials," *COLING 86: Proc. 11th Int. Conf. on Computational Linguistics*. Bonn, Aug. 25-29, 1986, pp.497-503.

[14] U. HAHN and U. REIMER, "Semantic Parsing and Summarizing of Technical Texts in the TOPIC System," R. Kuhlen *Informationslinguistik*, Tübingen, Niemeyer, pp. 153-193, 1986.

[15] L. A. HOLLAAR *et al.*, "Architecture and Operation of a Large, Full-Text Information-Retrieval System," D.K. Hsiao (ed) *Advanced Database Machine Architecture*, Englewood Cliffs/NJ, Prentice-Hall, pp. 256-299, 1983.

[16] W.J. HUTCHINS, "On the Problem of ´Aboutness´ in Document Analysis," *J. of Informatics* vol.1, no.1, pp.17-35, 1977.

[17] R. KAUNITZ, "Knowledge-Based Publishing: The Future is Now," *Proc. 5th National Online Meeting*. New York, April 10-12, 1984, Medford/NJ, Learned Information, pp. 135-140, 1984.

[18] M. LEBOWITZ, "Intelligent Information Systems," *Proc. 6th Annual Int. ACM SIGIR Conf. on Res. and Devel. in Information Retrieval*. Bethesda, Maryland, June 6-8, 1983, pp.25-30.

[19] D.L. LEE and F.H. LOCHOVSKY, "Text Retrieval Machines," D. Tsichritzis (ed) *Office Automation*, Berlin, Springer, p.339-375, 1985.

[20] W.G. LEHNERT, "Plot Units: A Narrative Summarization Strategy," W.G. Lehnert and M.H. Ringle (eds) *Strategies for Natural Language Processing*, Hillsdale/NJ, Erlbaum, pp.375-412, 1982.

[21] D.B. LENAT *et al.*, "Knoesphere: Building Expert Systems with Encyclopedic Knowledge," *IJCAI-83: Proc. 8th Int. Joint Conf. on Artificial Intelligence*. 8-12 August 1983, Karlsruhe, W. Germany, Los Altos/CA, W. Kaufmann, pp.167-169, 1983.

[22] F. RABITTI and J. ZIZKA, "Evaluation of Access Methods to Text Documents in Office Systems," C.J. van Rijsbergen (ed) *Research and Development in Information Retrieval*, Cambridge, Cambridge U.P., pp.21-40, 1984.

[23] U. REIMER and U. HAHN, "On Formal Semantic Properties of a Frame Data Model," *Computers and Artificial Intelligence* vol.4, no. 4, pp.335-351, 1985.

[24] N. SAGER, "Natural Language Information Formatting: The Automatic Conversion of Texts to a Structured Data Base," M. Yovits (ed): *Advances in Computers*, vol. 17, New York, Academic P., pp.89-162, 1978.

[25] G. SALTON, "*Dynamic Information and Library Processing*," Englewood Cliffs/NJ, Prentice-Hall, Ch.8: Automatic Document and Query Classification, 1975.

[26] R.C. SCHANK *et al.*, "Conceptual Information Retrieval," R.N. Oddy *et al.* (eds) *Information Retrieval Research*, London, Butterworths, pp. 94-116, 1981.

[27] R.C. SCHANK *et al.*, "An Integrated Understander," *Amer. J. of Computational Linguistics* vol.6, no.1, pp.13-30, 1980.

[28] R.F. SIMMONS, "A Text Knowledge Base from the AI Handbook," *Information Proc. & Management* vol.23, no.4, pp.321-339, 1987.

[29] S. SMALL and C. RIEGER, "Parsing and Comprehending with Word Experts (A Theory and its Realization)," W.G. Lehnert and M.H. Ringle (eds) *Strategies for Natural Language Processing*, Hillsdale/NJ, Erlbaum, pp. 89-147, 1982.

[30] J.W.T. SMITH, "*Full Text Information Systems: A Review and Assessment of the Technology*," Loughborough Univ. of Technology, Dept. of Lib. and Inf. Sci., Loughborough, Ms. of Sci. Diss., 1979.

[31] S.L. TAYLOR, "*Automatic Abstracting by Applying Graphical Techniques to Semantic Networks*," Northwestern Univ., Comp. Sci., Evanston/Ill., Ph.D.Thesis, 1974.

[32] U. THIEL and R. HAMMWÖHNER, "Graphical Interaction with a Full-Text Oriented Information System: The Retrieval Component of the End User Interface TOPOGRAPHIC," *Proc. 2nd Int. Conf. on the Application of Micro-Computers in Information, Documentation and Libraries*, Amsterdam, North-Holland, 1986.

[33] S.A. WEYER and A.H. BORNING, "A Prototype Electronic Encyclopedia," *ACM Transactions on Office Information Systems* vol., no.1, pp.63-88, 1985.