A Formal Model of Text Summarization Based on Condensation Operators of a Terminological Logic

Ulrich Reimer
Swiss Life
Information Systems Research Group
CH–8022 Zurich, Switzerland
reimer@swisslife.ch

Udo Hahn
Freiburg University
Computational Linguistics Group (CLIF)
D–79085 Freiburg, Germany
hahn@coling.uni-freiburg.de

Abstract

We present an approach to text summarization that is entirely rooted in the formal description of a classification-based model of terminological knowledge representation and reasoning. Text summarization is considered an operator-based transformation process by which knowledge representation structures, as generated by the text understander, are mapped to conceptually condensed representation structures forming a text summary at the representation level. The framework we propose offers a variety of subtle parameters on which scalable text summarization can be based.

1 Introduction

From its very beginning, the development of text understanding systems has been intimately tied to the field of knowledge representation and reasoning methods (Schank & Abelson 77). This close relationship was justified by the observation that any adequate form of text understanding not only requires grammatical knowledge about the particular language, but also, among others, has to incorporate knowledge about the domain the text deals with. Thus, the inferencing capabilities of knowledge representation languages were considered crucial for any adequate design of text understanding systems.

Out of this tradition a series of knowledge-based text summarization systems evolved, the methodology of which was almost exclusively based on the Schankian-type of *Conceptual Dependency (CD)* representations (e.g., (Cullingford 78, Lehnert 81, DeJong 82, Dyer 83, Tait 85, Alterman 86)). CD representations, however, are formally underspecified representation devices lacking any serious formal foundation. According to this, the summarization operations these first-generation systems provide use only informal heuristics to determine the salient topics from the text representation structures for the purpose of summarization. A second generation of summarization systems then adapted a more mature knowledge representation approach, one based on the evolving methodological framework of hybrid, classification-based knowledge representation languages (cf. (Woods & Schmolze 92) for a survey). Among these systems count SUSY (Fum et al. 85), SCISOR (Rau 87), and TOPIC (Reimer & Hahn 88), but even in these frameworks no attempt was made to properly integrate the text summarization process into the formal reasoning mechanisms of the underlying knowledge representation language.

This is where our interest comes in. We propose here a model of text summarization that is entirely embedded in the framework of a classification-based model of terminological reasoning. Text summarization is considered a formally guided transformation process on knowledge representation structures, the so-called text knowledge base, as derived by a natural language text parser. The transformations involved inherit the formal rigor of the underlying knowledge representation model, as corresponding summarization operators build on that model. Thus, our work describes a methodologically coherent, representation-theory-based approach to text summarization that has been lacking in the literature so far (for a survey cf. (Hutchins 87)). Aside from these purely representational considerations, the terminological reasoning framework for the summarization model we propose offers a variety of subtle parameters on which scalable summarization processes can be based. This contrasts, in particular, with those approaches to text summarization which almost entirely rely upon built-in features of frame and script-based representations and, consequently, provide rather simple reduction heuristics in order to produce text summaries (e g , (DeJong 82, Young & Hayes 85)) The formal model we present has been tested in TOPIC (Reimer & Hahn 88), a text summarization system which has been applied to expository texts in the domain of computer equipment as well as to various kinds of texts dealing with legal issues (company regulations, advisory texts, etc)

This paper is organized as follows In Section 2 we lay down a description of the syntax and semantics of the terminological logic which serves as the formal backbone for the specification of condensation operators on (text) knowledge bases From this formal description we then turn to the formal model of text summarization in Section 3

2 The Terminological Knowledge Representation Model

In the following, we describe a subset of a terminological logic (for an introduction to its underlying basic notational conventions, cf (Woods & Schmolze 92)) Section 2 1 considers the terminological component, while Section 2 2 deals with appropriate extensions for representing text-specific knowledge

2.1 The Basic Terminological Component

We distinguish two kinds of relations, namely properties and conceptual relationships A *property* denotes a relation between individuals and string or integer values A *conceptual relationship* denotes a relation between two individuals The concept description language provides constructs to formulate necessary (and possibly sufficient) conditions on the properties and conceptual relationships every element of a concept class is required to have The syntax of this language is given in Fig 1

$$\begin{aligned}
\langle \textit{terminology}\rangle &= \langle \textit{conc-intro}\rangle^{*}\\
\langle \textit{conc-intro}\rangle &= \langle \textit{conc-name}\rangle \leq \langle \textit{c-expr}\rangle\\
\langle \textit{c-expr}\rangle &= (\textbf{and}\ \langle \textit{c-expr}\rangle^{+}) \mid \langle \textit{conc-name}\rangle \mid\\
&\ (\textbf{all-p}\ \langle \textit{prop-name}\rangle\ \langle \textit{prop-range}\rangle) \mid\\
&\ (\textbf{all-r}\ \langle \textit{rel-name}\rangle\ \langle \textit{conc-name}\rangle^{+}) \mid\\
&\ (\textbf{exist-v}\ \langle \textit{prop-name}\rangle\ \langle \textit{value}\rangle) \mid\\
&\ (\textbf{exist-c}\ \langle \textit{rel-name}\rangle\ \langle \textit{conc-name}\rangle)\\
\langle \textit{prop-range}\rangle &= \langle \textit{int-range}\rangle \mid \langle \textit{string-range}\rangle\\
\langle \textit{conc-name}\rangle &= \langle \textit{identifier}\rangle
\end{aligned}$$

Figure 1 Syntax of a Terminological Logic

Every constructor in Fig 1 can be used to define a concept class (cf Fig 5) The **all-p** constructor introduces the class of individuals all of which have a certain property (whose value can vary from individual to individual) For example, (**all-p** *price* [$200,$5000]) denotes the class of individuals that have a property called 'price' with a value ranging between $200 and $5000 An individual can only have one value for each of its properties (cf Fig 2) The **all-r** constructor introduces a class of individuals that all participate in a certain kind of relationship to individuals from one of the concept classes given in the constructor For example, (**all-r** *equipped-with OperatingSystem ApplicationSoftware*) denotes the class of individuals that are in a relationship called 'equipped-with' only to individuals of the class 'OperatingSystem' or the class 'ApplicationSoftware' The distinction between the constructs **all-p** and **all-r** is uncommon in the domain of terminological logics (Woods & Schmolze 92), because primitive types like string and integer are usually considered to be concept classes as well As we will see in Section 3, the terminological reasoning underlying the text condensation process exploits this distinction between properties and relationships

The **exist-v** constructor introduces the class of individuals that all have a certain property value For example, (**exist-v** *weight* 6 *5lbs*) denotes the class of individuals that have a property called 'weight' with the value '6 5lbs ' The **exist-c** constructor defines the class of individuals that have a conceptual relationship to at least one individual of a specific concept class For example, (**exist-c** *has-part Cpu*) denotes the class of individuals that are in a relationship called 'has-part' to at least one individual of the class 'Cpu' With the **and** constructor several class descriptions can be combined into one (cf Fig 5) The model-theoretic semantics of the terminological language we use is depicted in Fig 2

2.2 Representing Text Knowledge

TOPIC's text parser heavily relies on terminological knowledge about the domain the texts deal with (Hahn 89) In the course of text analysis, the parser extends this domain knowledge incrementally by new concept definitions In order to distinguish between prior domain knowledge and newly acquired text knowledge we extend our basic terminological language with the constructs specified in Fig 3 The operator $\leq_T$ indicates a primitive concept originating from the text analysis Only a limited number of constructs can be used for such a concept definition – they correspond to the kinds of knowledge the parser can extract from a text (see Fig 5)

- A new concept can only be acquired when the text makes a reference to a superordinate concept already known in the domain knowledge Thus, the concept expression on the right-hand side of the $\leq_T$ construct must comprise a reference to a superordinate concept, as expressed

$$\begin{array}{rcl}
\varepsilon[c] \subseteq \varepsilon[cexpr] & , & \text{if} \quad c \leq cexpr \\
\varepsilon[\textbf{all-p}\ prop\ r_1 \quad r_n)] & = & \{x \in D \mid \|\{y \in D \mid \langle x, y\rangle \in \varepsilon[prop]\}\| = 1 \wedge \\
 & & \quad \forall y \quad ((\langle x, y\rangle \in \varepsilon[prop] \Rightarrow y \in (\varepsilon[r_1] \cup \quad \cup \varepsilon[r_n]))\} \\
\varepsilon[\textbf{all-r}\ rel\ c_1 \quad c_n)] & = & \{x \in D \mid \exists y \quad \langle x, y\rangle \in \varepsilon[rel] \wedge \\
 & & \quad \forall y \quad ((\langle x, y\rangle \in \varepsilon[rel] \Rightarrow y \in (\varepsilon[c_1] \cup \quad \cup \varepsilon[c_n]))\} \\
\varepsilon[(\textbf{exist-v}\ prop\ v)] & = & \{x \in D \mid \langle x, v\rangle \in \varepsilon[prop]\} \\
\varepsilon[(\textbf{exist-c}\ rel\ c)] & = & \{x \in D \mid \exists y \in \varepsilon[c] \quad \langle x, y\rangle \in \varepsilon[rel]\}
\end{array}$$

Figure 2 Model-Theoretic Semantics of the Constructs from Figure 1

$$\begin{array}{rcl}
\langle tconc\text{-}intro\rangle & = & \langle conc\text{-}name\rangle \leq_T (\textbf{and}\ \langle conc\text{-}name\rangle\ \langle tc\text{-}expr\rangle^+) \\
\langle tc\text{-}expr\rangle & = & (\textbf{exist-v}\ \langle prop\text{-}name\rangle\ \langle value\rangle\ \langle flag\rangle) \mid \\
 & & (\textbf{exist-c}\ \langle rel\text{-}name\rangle\ \langle conc\text{-}name\rangle\ \langle flag\rangle) \mid \\
 & & (\textbf{ccount}\ \langle aweight\rangle) \mid \\
 & & (\textbf{pcount}\ \langle prop\text{-}name\rangle\ \langle aweight\rangle) \mid \\
 & & (\textbf{rcount}\ \langle rel\text{-}name\rangle\ \langle conc\text{-}name\rangle\ \langle aweight\rangle)
\end{array}$$

Figure 3 Additional Terminological Constructs for Representing Text Knowledge

by the syntax

- Properties of a new concept can be learned (**exist-v** construct)
- Relationships to other concepts can be learned (**exist-c** construct) in case the relationship range is already defined by a corresponding **all-r** construct

The text-knowledge-specific versions of the **exist-v** and **exist-c** constructs have an additional argument which serves as a flag that is set whenever one of these constructs is added to a concept description (i e , when the associated property or relationship has been learned) The text condensation component of TOPIC makes use of this flag in order to determine those facts which have been learned since a certain reference point (where all flags were set to 0)

Besides acquiring new domain knowledge from a text, the parser performs book-keeping activities in order to record how often a concept, a property of a concept, or a relationship to another concept is explicitly or implicitly mentioned in the text For this purpose, we provide the constructs **ccount**, **pcount**, and **rcount** for concept descriptions These constructs belong to the text knowledge and can be applied to concept descriptions derived from the text as well as to concepts of the domain knowledge The **ccount** (**pcount**) construct indicates how often (a property of) a concept has been mentioned, whereas (**rcount** *rel conc aweight*) indicates how often the relationship *rel* to a concept *conc* has been referred to We call the numbers introduced by the count operators *activation weights* An (**rcount** *rel conc aweight*) construct can only occur as part of a text concept description when it also contains a construct (**all-r** *rel* c_1 c_n) where *conc* is subsumed by one of the c_is If this is not the case, **rcount** refers to a concept being related via a relationship *rel* which is not in the range of this relationship – thus, the **rcount** statement would make no sense Since none of the count constructs (and the flags) make an assertion about the meaning of the concepts involved, they have no influence on the concepts' extension (cf Fig 4) Fig 5 illustrates the application of multiple knowledge base operations resulting in the text knowledge representation for the newly learned concept 'Notebooster' as a specialization of 'Notebook'

3 Text Knowledge Condensation

The text condensation process examines the text knowledge base generated by the parser to determine certain distributions of activation weights, patterns of property and relationship assignments to concept descriptions, and particular connectivity patterns of active concepts in the concept hierarchy These constitute the basis for the construction of thematic descriptions as the result of text condensation Only the most significant concepts, relationships and properties (hereafter called *salient*) are considered as part of a topic description (cf Section 3 1) Thus, text condensation (or, equally, text summarization) can be considered an *abstraction process on (text) knowledge bases*

A *topic description* is a combination of salient concepts, relationships and properties of a formal text unit The computation of these concepts is started only in certain well-defined intervals In the sublanguage domain of expository texts, at least, topic

$$
\begin{aligned}
\varepsilon[c] \subseteq \varepsilon[cexpr] \quad &, \quad \text{if} \quad c \leq_T cexpr \\
\varepsilon[(\textbf{ccount}\ i)] &= D \\
\varepsilon[(\textbf{pcount}\ prop\ i)] &= D \\
\varepsilon[(\textbf{rcount}\ rel\ c\ i)] &= D \\
\varepsilon[(\textbf{exist-v}\ prop\ v\ f)] &= \varepsilon[(\textbf{exist-v}\ prop\ v)] \\
\varepsilon[(\textbf{exist-c}\ rel\ c\ f)] &= \varepsilon[(\textbf{exist-c}\ rel\ c)]
\end{aligned}
$$

Figure 4 Model-Theoretic Semantics of the Constructs from Figure 3

Domain Knowledge (Definition of a Concept Class)

```
      Notebook ≤ (and (all-r manufactured-by Manufacturer)
                      (exist-c has-part Cpu) (exist-c has-part RAM1)
                      (exist-c has-part HardDisk1)
                      (all-p weight [1lb ,15lbs ]) (all-p price [$200, $5000])
                      (all-r equipped-with OperatingSystem ApplicationSoftware)
                      (exist-c equipped-with MS-DOS))
          RAM1 ≤ (and (all-p size [1MB, 64MB])   )
     HardDisk1 ≤ (and (all-p size [100MB, 1GB])   )
```

Text Knowledge

```
   Notebooster ≤T (and Notebook (ccount 12)
                       (exist-c manufactured-by LeadingEdgeTech 1)
                       (rcount manufactured-by LeadingEdgeTech 1)
                       (exist-c has-part 486SL 1) (rcount has-part 486SL 3)
                       (exist-c has-part RAM1-1 1) (rcount has-part RAM1-1 2)
                       (rcount equipped-with MS-DOS 2)
                       (exist-v weight 6 5lbs  1) (pcount weight 1))
        RAM1-1 ≤T (and RAM1 (ccount 1)
                       (exist-v size 8MB 1) (pcount size 1))
```

Figure 5 Knowledge Representation Structures Resulting from Text Parsing

shifts occur predominantly at paragraph boundaries Therefore, text condensation is started at the end of every paragraph so that thematic overlaps as well as topic breaks between adjacent paragraphs can be detected and the extension of a topic be exactly delimited The condensation process yields a *set of topic descriptions*, each one characterizing one or more adjacent paragraphs of the text (cf Section 3 2) Finally, the entire collection of topic descriptions of a single text can be generalized in terms of a hierarchical *text graph* (cf Section3 3), the representation form of a text summary

3.1 Condensation Operators

We apply several operators to text knowledge bases to determine which concepts, properties, and relationships play a dominant role in the corresponding texts and thus should become part of their topic description All of these operators are grounded in the semantics of the underlying terminological logic Some of the operators make additional use of cut-off values which are heuristically motivated and have been evaluated empirically

Salient Concepts:

There are several criteria to determine salient concepts The most simple, less "knowledgeable" criterion considers all those concepts salient whose activation weight exceeds the average activation weight of all active concepts [1] A second criterion renders a concept salient, if the total sum of references made to properties of it and to relationships to other concepts is greater than it is, on the average, the case for all other active concepts (SC1) exploits the structure of the aggregation hierarchy and evaluates it by the associated activation weights (for the definitions of sets and functions we use below, cf Table 1)

(SC1) c is a salient concept iff

$$\sum_{rp_i \in R \cup P} rpcount(c, rp_i) > \frac{\sum_{c_i \in AC} \sum_{rp_j \in R \cup P} rpcount(c_i, rp_j)}{\|AC\|}$$

While (SC1) checks the total number of references made to *any* property or relationship, (SC2) is concerned with the number of *different* properties and relationships mentioned

[1] Throughout the paper, we call a concept c an active one, if $ccount(c) > 0$ (cf Table 1)

$$ccount(c) = n \Leftrightarrow c \leq (\text{and} \quad (\text{ccount } n) \quad) \text{ or } c \leq_T (\text{and} \quad (\text{ccount } n) \quad)$$

$$rpcount(c,rp) = \begin{cases} \sum_{c' \in C} rcount(c,rp,c'), & \text{if } rp \in R \\ pcount(c,rp), & \text{if } rp \in P \end{cases}$$

$$rcount(c,rel,c') = \begin{cases} n, & \text{if } c \leq (\text{and} \quad (\text{rcount } rel\ c'\ n) \quad) \\ n, & \text{if } c \leq_T (\text{and} \quad (\text{rcount } rel\ c'\ n) \quad) \\ 0, & \text{else} \end{cases}$$

$$pcount(c,prop) = \begin{cases} n, & \text{if } c \leq (\text{and} \quad (\text{pcount } prop\ n) \quad) \\ n, & \text{if } c \leq_T (\text{and} \quad (\text{pcount } prop\ n) \quad) \\ 0, & \text{else} \end{cases}$$

$$rpactive(c,rp) = \begin{cases} 1, & \text{if } rpcount(c,rp) > 0 \\ 0, & \text{else} \end{cases}$$

$$existcount(c,rp) = \begin{cases} \sum_{c' \in C} existc(c,rp,c'), & \text{if } rp \in R \\ \sum_{v \in V} existv(c,rp,v), & \text{if } rp \in P \end{cases}$$

$$existc(c,rel,c') = \begin{cases} 1, & \text{if } c \leq_T (\text{and} \quad (\text{exist-c } rel\ c'\ f) \quad) \wedge f \neq 0 \\ 0, & \text{else} \end{cases}$$

$$existv(c,prop,v) = \begin{cases} 1, & \text{if } c \leq_T (\text{and} \quad (\text{exist-v } prop\ v\ f) \quad) \wedge f \neq 0 \\ 0, & \text{else} \end{cases}$$

$$is\text{-}a(c_1,c_2) \Leftrightarrow c_1 \leq c_2 \vee c_1 \leq_T c_2 \vee c_1 \leq (\text{and} \quad c_2 \quad) \vee c_1 \leq_T (\text{and} \quad c_2 \quad)$$

$C = \{c \mid c \leq cexpr$ or $c \leq_T cexpr$ is part of the knowledge base$\}$

$AC = \{c \mid c \in C \wedge ccount(c) > 0\}$

$V =$ the set of all property values occurring in the knowledge base

$P =$ the set of all properties occurring in the knowledge base

$R =$ the set of all relationships occurring in the knowledge base

Table 1 Auxiliary Set and Function Definitions for Salience Computation

(SC2) c is a salient concept iff

$$\sum_{rp_i \in R \cup P} rpactive(c, rp_i) > \frac{\sum_{c_i \in AC} \sum_{rp_j \in R \cup P} rpactive(c_i, rp_j)}{\|AC\|}$$

The following two criteria exploit the inherent specialization structure of concept hierarchies (cf also (Lin 95) for a similar perspective on using semantic generalization relations for the computation of *concept* salience) They thus resemble criteria as used for the definition of macro rules to achieve summaries of texts (Correira 80, Dijk 80, Fum et al 85) These criteria also incorporate some notion of graph connectivity that has previously been considered by (Lehnert 81) for text summarization purposes (SC3) determines an active concept c as being salient *iff* a significant amount of subordinates of c are active, too (SC4) is similar but it marks all non-active (!) concepts as being salient which are related to a significant number of active subordinates Thus, concepts can be included in the topic description which have never been mentioned explicitly in a text (SC4) only yields the most specific concepts, i e , it excludes concepts for which the main criterion is fulfilled, but which are superordinate to another concept that also fulfills the criterion Lastly, (SC4) has a more stringent cut-off criterion This is necessary because it makes non-active concepts salient, accordingly, one has to be careful not to include irrelevant concepts Therefore, (SC4) requires a quarter of all subordinates (at least 3) to be active, while (SC3) has a relative cut-off value which gives lower percentages for greater numbers of subordinates (the cut-off values have been determined empirically)

(SC3) c is a salient concept iff

$$ccount(c) > 0 \ \wedge \ \|\{c' \mid is\text{-}a(c',c)\} \cap AC\| \geq \frac{\|\{c' \mid is\text{-}a(c',c)\}\|}{\|\{c' \mid is\text{-}a(c',c)\} \cap AC\|}$$

(SC4) c is a salient concept iff

$$\|\{c' \mid is\text{-}a(c',c)\} \cap AC\| \geq 3 \text{ and}$$

$$ccount(c) = 0 \wedge c \in cand \wedge \neg\exists c' \in cand \quad is\text{-}a(c',c)$$

where

$$cand = \{c \mid \|\{c' \mid is\text{-}a(c',c)\} \cap AC\| \geq 0\ 25 \ \ \|\{c' \mid is\text{-}a(c',c)\}\|\ \}$$

Salient Relationships and Salient Properties: Just as certain concepts may have been dealt with more extensively in a text than other ones, single features of a concept definition may have been more focused on than other features of the same concept The following criterion renders a relationship (or property) rp salient if the number of concepts (or property values) to which c has been related via rp is greater than it is, on the average, the case for relationships (or properties) in c Note that c must be a concept learned during text parsing, as learning new features is only possible for such concepts (SR1) is evaluated for salient concepts only because we are not interested in salient features of concepts being irrelevant for a topic description

(SR1) A relationship or property rp of a salient concept c is considered salient in the context of c iff $\sum_{rp_i \in R \cup P} rpactive(c, rp_i) \geq 3$ and it holds that

$$existcount(c, rp) > \frac{\sum_{rp_j \in R \cup P} existcount(c, rp_j)}{\sum_{rp_j \in R \cup P} rpactive(c, rp_j)}$$

Related Salient Concepts:
A concept c' is considered a *related salient concept* for the salient concept c if there is a relationship rel from c to c' where the sum of the activation weights of all relationships of type rel from c to c' or to subordinates of c' is greater than the average activation weight of all active relationships for c If c' is determined as a related salient concept for c, then the associated relationship rel becomes a salient relationship of c This criterion combines knowledge about conceptual aggregation and concept hierarchies with a numerical weights

(SRC1) A relationship rel between a salient concept c and some concept c' is considered salient and c' is considered a related salient concept iff $\sum_{rel_i \in R} rpactive(c, rel_i) \geq 3$ and the following holds

$$\sum_{\{c_i \mid c_i = c' \vee is\text{-}a(c_i, c')\}} rcount(c, rel, c_i) > \frac{\sum_{rel_j \in R} rpcount(c, rel_j)}{\sum_{rel_j \in R} rpactive(c, rel_j)}$$

In the following, (c) denotes a salient concept c, $(c \quad r)$ a salient relationship r of concept c, and $(c \quad r \quad c')$ denotes a related salient concept c' for concept c with respect to the relationship r

3.2 Paragraph-Level Topic Descriptions

The condensation operators just introduced are applied at the end of every paragraph to the text knowledge base which results from parsing that paragraph They yield a set of salient concepts, relationships, properties, and related salient concepts In the next step, these raw data are combined to form a compound topic description for that paragraph The combination is performed according to the following rules

- A salient concept (c) which is already covered by a salient relationship or property $(c \quad rp)$ or a related salient concept $(c \quad r \quad c')$ is removed
- A salient relationship $(c \quad r)$ already covered by a related salient concept $(c \quad r \quad c')$ is removed

After having determined the topic description td of the previous paragraph a check is made whether this paragraph deals with the same topic as the immediately preceding paragraph(s), or *vice versa* If this is the case, the topic description td of the current paragraph is added to the topic description of the preceding paragraph(s), otherwise a new current topic description is created and set to td Formally (cf also Table 2)

> Let td be the topic description of the last paragraph and td_i be the topic description of one or more paragraphs immediately preceding td, then
>
> td_i is set to $td_i \cup td$ if $td_i \cup td = td_i \vee td_i \cup td = td$
> otherwise td_i is not modified and td_{i+1} is set to td

For example, the following two topic descriptions of adjacent paragraphs would be combined into one {(Notebooster has-part 486SL), (Notepad)}, {(Notebooster has-part)}

Analyzing a text this way yields a set of consecutive topic descriptions $td_1, \quad , td_n$, each one characterizing the topic of one or more adjacent paragraphs To every topic description td_i we associate the corresponding text passage and the facts acquired from it We call the resulting compound structure, in which different media combine, a *(hyper)text constituent*

3.3 The Text Graph

From the topic description contained in a text constituent, more generic constituents can be derived in terms of a hierarchy of topic descriptions, forming a *text graph* The construction of a text graph proceeds from the examination of every pair of basic topic descriptions and takes their conceptual commonalities to generate more generic thematic characterizations Exhaustively applying this procedure (also taking the newly generated topic abstractions

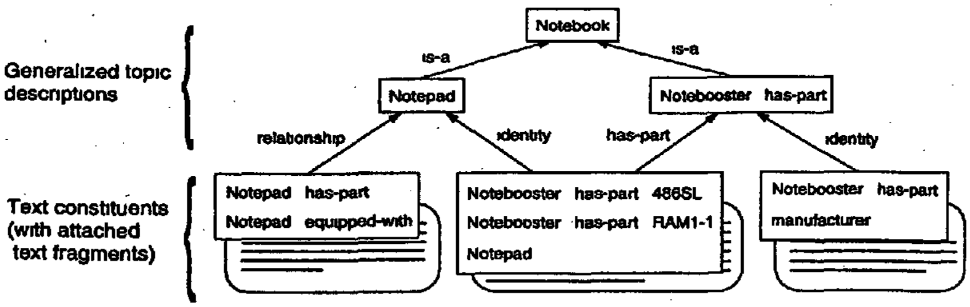

Figure 6 An Illustrative Fragment of a Text Graph (redundant Is-A relations are omitted)

$$
\begin{aligned}
td \cup \{(c)\} &= \begin{cases} td \text{ , if } \exists r \ \ (c \ \ r) \in td \\ \qquad\qquad \vee \exists r, c' \ \ (c \ \ r \ \ c') \in td \\ td \cup \{(c)\} \text{ , else} \end{cases} \\
td \cup \{(c \ \ r)\} &= \begin{cases} td \text{ , if } \exists c' \ \ (c \ \ r \ \ c') \in td \\ td \cup \{(c \ \ r)\} \setminus \{(c)\} \text{ , else} \end{cases} \\
td \cup \{(c \ \ r \ \ c')\} &= td \cup \{(c \ \ r \ \ c')\} \setminus \{(c), (c \ \ r)\} \\
td \cup td' &= \bigcup_{e \in td \cup td'} \{e\}
\end{aligned}
$$

Table 2 The Operator ∪ for Combining Topic Descriptions (\ stands for the set complement operator)

into consideration) results in a text graph as a hierarchy of topic descriptions 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 Their hierarchical organization yields different levels of granularity of text summarization (see Fig 6) It is exactly this emergent generalization property of the text graph that we consider the source of our scalability arguments Very brief summaries, only intended to capture the main topics of the text, can be generated from the upper level of the text graph Continuously deepening the traversal level of the text graph provides access to more and more specific information Our procedure thus combines the potential for supplying summaries on the indicative as well as informative level of text knowledge abstraction (cf (Borko & Bernier 75) for the distinction between indicative and informative abstracting)

4 Related Work

The task domain of text summarization is characterized by a *"clash of civilizations"* From the point of view of natural language understanding proper (Schank & Abelson 77, Dyer 83) it is considered a heavily knowledge-based task requiring a substantial knowledge background In the field of information retrieval, however, the corresponding task of automatic abstracting, has been considered from its very beginning (Luhn 58), a problem that can be dealt with by surface-level pattern matching techniques and statistical methods originally developed for lexical selection tasks such as automatic indexing or classification (Salton et al 94) This approach has recently been given a lot of attention again, mainly due to the renaissance of statistical methodology in the field of parsing and tagging (Kupiec 95) Given a statistical approach, however, automatic abstracting boils down to a sentence extraction problem, *viz* determining the most salient sentences based on surface-level lexical or positional indicators

We adhere to the knowledge-based paradigm of abstracting and propose to fully integrate text knowledge abstraction in a terminological reasoning model In such an approach, text understanding and summarization are considered within a formally homogeneous framework Moreover, and most important, this model allows for a *staged* provision of information in summaries based on conceptual criteria (as illustrated by the discussion of text graphs) Such a functionality is unlikely to be achieved by surface-oriented approaches due to their inherent limitations to provide cohesive summaries from large sets of extracted sentences (Paice 90)

5 Conclusions

We have introduced an approach to text summarization which is solidly rooted in the formal semantics of the underlying terminological representation system In this approach, text summarization is an operator-based transformation process on knowledge representation structures that have been derived by the text understanding system Currently, the summarization process considers only activity and connectivity patterns in the text knowledge base In the future, we plan to augment these criteria and to exploit text coherence patterns for summarization (cf (Hahn 90) and related proposals by (Alterman 86)) The implementation of the summarization system and its associated text understander have proved functional with expository texts in the domain of information technology as well as with texts from the legal and business domains

References

Alterman, R [1986] Summarization in the small In N E Sharkey (Ed), *Advances in Cognitive Science 1* (pp 72-93) Chichester Ellis Horwood

Borko, H , Bernier, C L [1975] *Abstracting Concepts and Methods* New York etc Academic Press

Correira, A [1980] Computing story trees *American Journal of Computational Linguistics, 6* (3-4), 135-149

Cullingford, R E [1978] *Script Application Computer Understanding of Newspaper Stories* New Haven, CT Department of Computer Science, Yale University (Research Rep 116)

DeJong, G [1982] An overview of the FRUMP system In W Lehnert & M H Ringle (Eds), *Strategies for Natural Language Processing* (pp 149-176) Hillsdale, NJ L Erlbaum

Dijk, T A van [1980] *Macrostructures an Interdisciplinary Study of Global Structures in Discourse, Interaction and Cognition* Hillsdale, NJ L Erlbaum

Dyer, M G [1983] *In-Depth Understanding a Computer Model of Integrated Processing for Narrative Comprehension* Cambridge, MA MIT Press

Fum, D , Guida, G , Tasso, C [1985] Evaluating importance a step towards text summarization *IJCAI'85 Proc of the 9th International Joint Conf on Artificial Intelligence* (Vol 2, pp 840-844) Los Angeles, Cal , 18-23 August 1985 Los Altos, CA W Kaufmann

Hahn, U [1989] Making understanders out of parsers semantically driven parsing as a key concept for realistic text understanding applications *International Journal of Intelligent Systems, 4* (3), 345-393

Hahn, U [1990] Topic parsing accounting for text macro structures in full-text analysis *Information Processing & Management, 26* (1), 135-170

Hutchins, J W [1987] Summarization some problems and methods *Informatics 9 Proc by the Aslib Co-ordinate Indexing Group Meaning the Frontier of Informatics* (pp 151-173) Cambridge, U K , 26-27 March 1987 London Aslib

Kupiec, J , Pedersen, J , Chen, F [1995] A trainable document summarizer In *SIGIR '95 Proc of the 18th Annual International ACM SIGIR Conf on Research and Development in Information Retrieval* (pp 68-73) Seattle, Wash , USA, July 9-13, 1995

Lehnert, W [1981] Plot units and narrative summarization *Cognitive Science, 5*, 293-331

Lin, C -Y [1995] Knowledge-based automatic topic identification *Proc of the 33rd Annual Meeting of the Association for Computational Linguistics* (pp 308-310) Cambridge, Mass , USA, 26-30 June 1995

Luhn, H P [1958] The automatic creation of literature abstracts *IBM Journal of Research and Development, 2* (2), 159-165

Paice, C D [1990] Constructing literature abstracts by computer techniques and prospects *Information Processing & Management, 26* (1), 171-186

Rau, L F [1987] Knowledge organization and access in a conceptual information system *Information Processing & Management, 23* (4), 269-283

Reimer, U , Hahn, U [1988] Text condensation as knowledge base abstraction *Proc of the 4th Conf on Artificial Intelligence Applications [CAIA]* (pp 338-344) San Diego, Cal , March 14-18, 1988

Salton, G , Allan, J , Buckley, C , Singhal, A [1994] Automatic analysis, theme generation, and summarization of machine-readable texts *Science, 264* (3, June), 1421-1426

Schank, R C , Abelson, R P [1977] *Scripts, Plans, Goals and Understanding an Inquiry into Human Knowledge Structures* Hillsdale, NJ L Erlbaum

Tait, J I [1985] Generating summaries using a script-based language analyser In L Steels & J A Campbell (Eds), *Progress in Artificial Intelligence* (pp 312-318) Chichester Ellis Horwood

Woods, W A , Schmolze, J G [1992] The KL-ONE family *Computers and Mathematics with Applications, 23* (2-5), 133-177

Young, S R , Hayes, P J [1985] Automatic classification and summarization of banking telexes *Proc of the 2nd Conf on Artificial Intelligence Applications [CAIA]* (pp 402-408) Miami Beach, FL, December 11-13, 1985