Multilingual Flexible and Robust Summarization

Walter Kasper and **Jörg Steffen***
DFKI GmbH, Stuhlsatzenhausweg 3, D-66123 Saarbrücken, Germany
{kasper|steffen}@dfki.de

Abstract

SumEx is a multilingual, flexible and robust system for automatic text summarization following a sentence extraction approach. The paper presents an overview of this system.

1 Introduction

The MEMPHIS project (http://www.ist-memphis.org) aims at developing a portal for cross-lingual premium content services, targeting mainly portable thin clients, like mobile phones, PDAs etc. The core of the system is a cross-lingual transformation layer doing cross-lingual *information extraction* and *summarization* of source documents, *translation* to the customers' target languages as well as *generation* of documents according to the restrictions and requirements of the various target devices for distribution. Here we will report on current work on the summarization component **SumEx** of that system.

Fundamental requirements for the summarizer component from the application include:

- *easily adaptable* to input documents of various formats, topics and domains. At present, HTML, Reuter's NewsML and plain text documents are supported. Application areas are book announcements on various topics as well as financial news.
- *multilingual*: at present, **SumEx** supports German, English, French, Italian and Spanish.
- *flexible* with respect to output length to support various distribution channels: **SumEx** allows to specify upper limits as string lengths as well as compression rates such as 10% of original text.
- *indicative abstract* quality.[1]
- *adaptive* to user provided queries
- *robust* in dealing with all kinds of texts and styles

SumEx satisfies these requirements by following a sentence extraction methodology in the tradition of (Edmundson, 1969). One of its distinguishing features is the combination of several heuristics that use term weight statistics and exploit the layout characteristics and structure of documents (cf. (Preissner, 2000) for motivation.).

2 Overview of the Summarizer System

In order to be independent of the various input document formats, the documents first are transformed into a neutral XML format, the *Memphis File Format MFF*. This MFF representation maintains meta information about the document, such as its language, source, topics etc. and represents the source text in a standardized way including its most important properties, such as headings, paragraph structure and some layout characteristics.

The MFF document is passed to the summarizer. In a preprocessing step a tokenizer and segmenter with language specific rules enrich the text with information about text units ("sentences") and tokens. As far as available, a morphological analysis of the tokens provides

* The work reported here is part of the *MEMPHIS* project funded by the European Commission under contract IST-2000-25045 in the *Information Society Technologies* (IST) program. A predecessor of **SumEx** is described in (Preissner, 2000).

[1] See (Mani, 2001) for an excellent overview of summarization types and methods.

stems as normalized token forms and their *part-of-speech*. Customizable filters then allow to remove some "noise", found often in documents derived from HTML pages with complicated navigation etc. structures, such as text units with less than five tokens or not ending with a proper punctuation mark.

The major *summarization analysis* applies a set of heuristics to the text units. The heuristics are based on *term weight statistics*, *positional* and *layout* information (see Sect. 3). Each heuristic modifies an initial weight assigned to each text unit yielding a final ranking.

An extensible set of *output managers* finally allows to present the summary in various formats. By default, a plain text summary is created by collecting the text units with the highest ranking until the desired summary length is reached. To improve coherence, the selected text units can be displayed in their original order. In addition, a list of most relevant keywords and phrases can be derived from the term weight statistics. The desired length of the summary can be specified either as the maximal string length or as compression rate, such as 10% of the original text size.

3 The Heuristics

In the following section we take a closer look at the three heuristics currently implemented for weighting the text units.

The final weight of a text unit $w(tu)$ is calculated from the weights $w_{h_i}(tu)$ given to it by each heuristic h_i. The range of weights is different for each heuristic. To combine the weights, a normalization must be applied. All weights given by a single heuristic are projected into an interval $[0, 1]$ by keeping track of the minimum and maximum weight given by that heuristic and then modifying the weight for a text unit in the following way:

$$norm\_w_{h_i}(tu) = (w_{h_i}(tu) - min_{h_i})/(max_{h_i} - min_{h_i}) \quad (1)$$

To allow for fine tuning and experimentation, the influence of each heuristic on the final text unit weight can be modified via a modifier mod_i. So the final weight of a text unit is:

$$w(tu) = \sum_{i=1}^{3} mod_i * norm\_w_{h_i}(tu) \quad (2)$$

The weight for a text unit is derived from weights given to its tokens. For length normalization the sum of the tokens weights is divided by their number. Non-word tokens and tokens contained in language specific stop word lists, so-called non-content words, are ignored when calculating the text unit weight.

3.1 The Term Weight Heuristic

Term weighting is based on a standard *tf.idf* approach (cf. (Salton and Yang, 1973)), more precisely, the *atc* variant (cf. (Paijmans, 1997)), treating the normalized tokens as terms. The term frequency $tf(t)$ is the number of occurrences of term t in the input document. The *atc* scheme doesn't use the *simple* but the *weighted* term frequency

$$wtf(t) = 0.5 + 0.5 * tf(t)/max_{\text{tf}} \quad (3)$$

where max_{tf} is the frequency of the term with the highest frequency in the document. The inverse document frequency $idf(t)$ reflects the distribution of term t over a document corpus and is defined as

$$idf(t) = log_2(N/n(t)) \quad (4)$$

where N is the number of documents in the corpus and $n(t)$ is the number of documents in which t occurs at least once. For new terms t unseen in the training corpus, $idf(t) = log_2(N)$ is used as default. The tf.idf weight of a term is then defined as

$$w_{\text{tf.idf}}(t) = wtf(t) * idf(t) \quad (5)$$

The idf of a term is retrieved from an idf database that must be built in advance based on a training corpus. The reliability of the term weight heuristic is improved if the training corpus is from the same domain as the documents to summarize.[2] Finally a cosine normalization is applied by

$$norm\_w_{\text{tf.idf}}(t) = w_{\text{tf.idf}}(t)/\sqrt{\sum_{k=1}^{T} w_{\text{tf.idf}}(t_k)^2} \quad (6)$$

where T is the number of different terms in the document.

[2] It's also possible to define a domain dependent stop word list via the idf.

3.2 The Layout Heuristic

The idea of this heuristic is to exploit that authors often change font properties to highlight or mark important phrases and text parts, and so should be relevant for the summary, too. On the other hand, some text might be marked as unimportant, e.g. by using a font size smaller than the default.

The summarizer exploits style, size and color properties of fonts which are mapped to a weight. Before the layout weight for a text unit can be calculated, the document is scanned for tokens with special font markup. These tokens are collected in a *layout token set* and assigned a *layout weight*. In the MFF format all text with a non-standard font is enclosed within an opening and a closing font tag. The document is scanned for these font tags and with each opening tag found, the mapped weight of the font is added to the *local layout weight*. When the corresponding closing font tag is found, the local layout weight is reduced again.

All tokens that follow the opening font tag have the local layout weight added to their layout weight. A token is also added to the layout token set if not included yet. So the layout weights are summed up for tokens with more than one font markup or if there are several occurrences of the same token with a font markup. The layout token set allows to propagate a term's layout weight to term occurrences without markup. This takes into account that important terms may be marked when they are introduced first, but later used without it.

With the layout token set complete, it is checked for each text unit if it contains a token that is also in the layout token set. If yes, the token is given the layout weight of the token in the layout token set. The layout weight of a text unit is the sum of the weights given to its tokens divided by their number.

3.3 The Positional Heuristic

The idea of this heuristic is that headings and the first text unit of a paragraph are more relevant for a summary than other text units as has been shown in many summarization studies. The positional heuristic exploits the different levels of headings and the paragraph structure of the document. In contrast to the other heuristics, it weights text units directly and not via their tokens.

In a first step headings are given a weight according to their level, e.g. *section, subsection* etc. In the second step the paragraph structure of the document is recursively traversed. For each paragraph p the first text unit tu occurring in it is searched for recursively. It is possible that p consists of subordinated paragraphs only, e.g. *list items* and *table rows* count as subordinated paragraphs here. When tu is found it gets the weight given to the lowest level headings minus the depth of the paragraph p in the paragraph structure. If a text unit was already given a positional weight it cannot be given such a weight a second time, e.g. if it's also the first text unit in a sub-paragraph.

3.4 Query Adaptation

SumEx allows the user to provide a set of terms he is specially interested in. The system uses these *query terms* to honor sentences containing these terms specially and so adapts the summaries to these user queries. Technically, in the current system the query terms are treated in analogy to the layout heuristics by treating these terms as if they would appear in the document with a bold font style. This does not enforce that only sentences containing query terms are selected. For certain purposes such as use in a search engine it might be desirable to change this behavior.

4 Evaluation and Outlook

(Preissner, 2000) made an evaluation of the strategies and heuristics used in **SumEx**, giving an acceptability rate of 86% for the summaries. An evaluation for the new domains and applications in MEMPHIS — at present, book and media announcements and financial news — is ongoing. First results show that the layout heuristics does not play a major role as layout markup rarely is significant in the documents MEMPHIS deals with, especially in the financial news. In other cases, the layout supports the positional heuristics rather than leading to significant different results.

Significant further improvement we expect from exploiting and extending the mechanisms for query-adaptive summarization to tune the extracts to focus on the topics the MEMPHIS customer is interested in and has registered for.

This will mean to integrate topic/domain ontologies and classification into the summarizer process as additional knowledge sources.

Other future development will focus on

- improved tools for domain adaptation
- integration with information extraction
- strategies for shallow coherence smoothing for the extracts
- exploration of new heuristics using cue phrases to weight text units

References

H.P. Edmundson. 1969. New methods in automatic extracting. *Journal of the ACM*, 16:264–28. Reprinted in (Maybury and Mani, 1999, 23-42).

Inderjeet Mani. 2001. *Automatic Summarization*. Amsterdam/Philadelphia: John Benjamins.

Mark Maybury and Inderjeet Mani, editors. 1999. *Advances in Automatic Text Summarization*. Cambridge/Mass: MIT Press.

H. Paijmans. 1997. Gravity wells of meaning: Detecting information-rich passages in scientific texts. *Journal of Documentation*, 53:520–536.

Annette Preissner. 2000. Flexible hybrid summarization of multilingual markup documents. Diplomarbeit, Universität des Saarlandes, Lehrstuhl für Computerlinguistik, Saarbrücken.

G. Salton and C.S. Yang. 1973. On the specification of term values in automatic indexing. *Journal of Documentation*, 29:351–372.