ABSTRACT

Document summarization techniques can be profitably used for automatic production and delivery of multimedia information. In this paper we describe a system for summarizing HTML documents (retrieved from the Internet) using several heuristic optimization criteria. An overview of the system and some preliminary results are described.

1. INTRODUCTION

Web summarization – i.e. the process of automatic extraction of relevant information given a list of topics from different web sites – is of great utility for a variety of applications and in particular for automatic indexing and categorization in order to facilitate the production and accessibility of new multimedia contents.

To give an example, let us consider a news reporter that needs to have some info delivered to his PDA or cell-phone for writing an article or a news and does not want to waste time analyzing a great amount of information sources: a summarization system could helpfully produce a short summary of the gathered retrieved information, and in addition, such a summary could be easily managed and accessed using light mobile devices.

Summarization is a well known research field in the artificial intelligence community. Most of the proposed summarization techniques are query independent and follow one of the following two approaches: they simply extract relevant parts of the text, considering the document as a sequence of unstructured set of text blocks, or they employ Natural Language Processing techniques. The former approach ignores the structural information of documents, while the latter is more computationally expensive for large data-sets and sensitive to the different writing styles.

Anyway, in the last few years, several attempts have been done for improving the summarization tasks.

Varadarajan and Hristidis [1] propose a method to add structure, in form of a graph, to the text documents in order to allow effective query specific summarization. That is, they view a document as a set of interconnected text fragments (passages) and focus on keyword queries since keyword search is the most popular information discovery method on documents, because of its power and ease of use. Their technique has the following key steps: first, at the preprocessing stage, they add structure to every document, which can then be viewed as a labeled, weighted graph, called document graph. Then, at query time, given a set of keywords, they perform keyword proximity search on the document graphs to discover how the keywords are associated in the document graphs. For each document its summary is the minimum spanning tree on the corresponding document graph that contains all the keywords (or equivalent based on a thesaurus).

In [2], such a technique is used to implement a system that given a keyword query, dynamically generates new pages, called composed pages, which contain all query keywords. The composed pages are generated by extracting and stitching together relevant pieces from hyper-linked Web pages, and retaining links to the original Web pages. To rank the composed pages, the authors consider both the hyper link structure of the original pages, as well as the associations between the keywords within each page.

In [3], the authors proposed the system SWEeT (Summarizer of WEB Topics). In SWEeT, the user query is given to a search engine that answers with a set of relevant documents. Their contents, together with some additional information, e.g. date of publication, are extracted and passed on to the summarizer. The first task for the summarizer is to extract the most important sentences from the set of documents. The approach follows what has been called a term-based strategy: find the most important information in the document(s) by identifying its main terms, and then extract from the document(s) the most important information (i.e., sentences) about these terms. Moreover, to reduce the dimensionality of the term space, they use the latent semantic analysis, which can cluster similar terms and sentences into topics on the basis of their use in context.

Eventually, Fayzullin et al. [4] proposed some heuristic methodologies for extracting summaries in annotated video databases, based on the idea of producing high quality summaries w.r.t. three different criterion, i.e. Continuity, Priority...
and Repetition (CPR). In short, a summary must include high priority frames and must be continuous and non-repetitive.

Our paper uses the main ideas proposed in the previous papers in order to design an efficient and simple summarization system particularly suited for web information sources. In particular, differently from the other papers, we design a multi-sources summarization algorithm that particularly fit the internet search engines requirements.

2. TOPIC DETECTION AND SUMMARIZATION

In our vision, a web summarization system is very similar to a classic search engine: a user, by means of several key words, activates a Search Engine Module that stores in a Web Pages Repository all web pages satisfying searching criteria. Successively, the Information Extraction Module processes the stored web pages and extracts, for each web page, the most relevant information w.r.t. the given input keywords. More in details, in a first step useful text of web pages is extracted by parsing HTML code and, in a second stage, appropriate NLP algorithms determine, for each sentence in the text, relevant topics; these ones are definitively stored in a Sentence Repository together with original sentences.

In our model, a topic detection algorithm determines a set of topics, i.e. labels, and an associated confidence value, representing the relative importance of the label w.r.t. the other ones in the text (namely $A$). In other words, a topic detection process is a function $\phi : A \rightarrow \{\gamma_i, \epsilon_i\}$, being the couple $(\gamma_i, \epsilon_i)$ a generic label with the related confidence.

More in details, the $\phi$ function is accomplished as follows: we first extract “names” from the text (names of people or organizations, geographical locations and so on), using the Named Entity Recognition (NER) module of Stanford NLP libraries; we substitute them with the related “entity” (e.g. the sentence “Bob works in BMW” is transformed into “Person works in Organization”). Then, we perform tokenization, stemming and part of speech tagging. To the aim of disambiguating word senses and obtaining particular labels describing the events presented in the annotations, we apply a Word Sense Disambiguation (WSD) algorithm to nouns. In particular we exploit the algorithm described in [5], that disambiguates the words applying “minimum common hyperonym” technique to the parsed and stemmed initial sentence.

In the successive step, a suitable Topic Extraction algorithm determines the set of labels that are more significant to the description of the content of the text [6]. In particular, for each label extracted by the WSD module, a confidence value is computed considering both the semantic similarity (by exploiting the same algorithm used for WSD) of the labels to the other ones, and the related frequency in the text. Finally, the Topic Extractor selects the top-K labels by using a a confidence threshold $\tau$ determined in an experimental way.

Then, a Summary Management Module has the task of generating the best summary related to the search keywords. To these purposes an optimization algorithm is used (as will be successively described) to create a summary that takes into account some goodness criteria such as the priority respect to the user needs, the legibility or continuity and the non-repetition of sentences. The obtained summaries are finally stored into a Summaries Repository and delivered to the users in different formats and ways by the Delivery Module.

2.1. The summarization algorithm

Let us first give some preliminary definitions that will explain what we mean with “summary”.

Definition 1 (Summarizable Sentence)

A Summarizable Sentence $(\sigma)$ is a triple:

$$\sigma = \langle s, \Lambda_k, WS \rangle$$

$s$ being the original sentence, $\Lambda_k = \lambda_1, ..., \lambda_k$ being the set of more significant labels associated to $s$ and $WS$ being an identifier of the web page source from which $s$ comes from.

For example, $\Lambda_k$ could be obtained using the previously cited topic extraction algorithms.

Definition 2 (Semantic Correlation)

Let $\sigma_i$ and $\sigma_j$ be two summarizable sentences. The Semantic correlation between the two sentences is defined as:

$$S(\sigma_i, \sigma_j) = \frac{1}{k_i \cdot k_j} \sum_{h=1}^{k_i} \sum_{z=1}^{k_j} e^{-w_1 \cdot (\lambda^i_h, \lambda^j_z)} \left(1 - e^{-w_2 \cdot d(\lambda^i_h, \lambda^j_z)}\right)$$

$\lambda^i_h$ and $\lambda^j_z$ being the values of $h-th$ and $z-th$ labels of $\sigma_i$ and $\sigma_j$ sentences respectively, $l(\lambda^i_h, \lambda^j_z)$ being the path length between $\lambda^i_h$ and $\lambda^j_z$ and $d(\lambda^i_h, \lambda^j_z)$ being the depth in the WordNet hierarchy of the subsumer $\lambda^i_h$ and $\lambda^j_z$; $w_1$ and $w_2$ being the parameters scaling the contribution of shortest path length and depth, respectively.

The previous definition derives from the empirical consideration that the more two sentences, whose related concepts derive from a common root concept, are similar the more they have close positions, considering a hierarchical concepts network or a vocabulary (in this case WordNet); the similarity decreases exponentially as suggested in several psycho-perceptual studies. The previous definitions can be easily extended to the case of the comparison between a summarizable sentence and a set of query keywords (they can be considered as labels of a query sentence).

Definition 3 (n-length Summary)

A n-length summary $\Sigma_n$ is a sequence of summarizable sentences:

$$\Sigma_n = \{\sigma_1, ..., \sigma_n\}$$

$n$ being the length of the summary.
Now, we are in the position of introducing the concepts of \textit{Importance}, \textit{Consistence} and \textit{Recurrence} that are at the base of our summarization algorithm.

\textbf{Definition 4 (Importance)}

An Importance ($I$) is a function that associates an importance score $\tau_I \in [0, 1]$ to a given summary $\Sigma$, with respect to a set of keywords $\Omega = \omega_1, \ldots, \omega_m$:

\begin{equation}
I : \Sigma, \Omega \rightarrow \tau_I
\end{equation}

where the importance score is calculated by means of the following formula:

\begin{equation}
\tau_I = \frac{\sum_{h=1}^{n}(w_k \cdot S(\sigma_h, \Omega) + w_s \cdot l(\sigma_h))}{n}
\end{equation}

being $\sigma_h$ the $h$-th sentence of the summary $\Sigma$ (and $n$ its related length), $l(s)$ a function that associates to each sentence a degree of relevance on the base of its original location in the web page text and $w_k$ and $w_s$ two coefficients such that $w_k + w_s = 1$.

\textbf{Definition 5 (Consistence)}

A Consistence ($C$) is a function that associates a consistence score $\tau_C \in [0, 1]$ to a given summary $\Sigma$:

\begin{equation}
C : \Sigma \rightarrow \tau_C
\end{equation}

being the consistence score calculated by means of the following formula:

\begin{equation}
\tau_C = \frac{\sum_{h=1}^{n}(\kappa_h \cdot \sigma_h \cdot \sigma_h)}{n}
\end{equation}

being $n$ the length of summary $\Sigma$, $N_S$ the number of different web pages, $n_{h_S}$ the total number of summary sentences coming form the $h - th$ web page, $n_{h_S}$ the number of consecutive sentences coming from the $h - th$ web page and $\kappa_h$ is a particular weight that the take into account the reliability or credibility of the $h - th$ web page source.

\textbf{Definition 6 (Recurrence)}

A Recurrence ($R$) is a function that associates a recurrence score $\tau_R \in [0, 1]$ to a given summary $\Sigma$:

\begin{equation}
R : \Sigma \rightarrow \tau_R
\end{equation}

where the recurrence score is calculated by means of the following formula:

\begin{equation}
\tau_R = \frac{\sum_{h \neq z} S(\sigma_h, \sigma_z)}{n}
\end{equation}

being $\sigma_h$ and $\sigma_z$ two distinct sentences of the summary $\Sigma$ and $n$ its related length.

In other terms, the \textit{Importance} criterion evaluates how the summary takes into account the user keywords; \textit{Consistence} is a criterion that gives more importance to those blocks of consecutive sentences coming from the same sources; \textit{Recurrence} criterion, eventually, measures how semantically similar are the concepts presented in the sentences.

\textbf{Definition 7 (Optimal Summary Valuation)}

Suppose that $\mathcal{S}$ is a set of $n$-length summaries $\Sigma_n$ and $\alpha$, $\beta$ and $\gamma \geq 0$ are integers. A summary valuation is a function $\text{eval} : \mathcal{S} \rightarrow \mathbb{R}$, of the form

\begin{equation}
\text{eval}(\mathcal{S}) = \alpha \cdot \tau_I + \beta \tau_C + \gamma(1 - \tau_R)
\end{equation}

In a previous paper of one of the authors [4], it has been demonstrated that the Optimal Summary valuation is a NP-hard problem. In the following we will give a sub-optimal heuristic algorithm.

Our algorithm is based on genetic programming; it starts working on an initial solution that takes into account only the \textit{Importance} criterion, then a \textit{mutation} operator is applied.

\begin{algorithm}[H]
\caption{Importance, Consistency and Recurrence - ICR Algorithm}
\begin{algorithmic}[1]
\Require $\Sigma$, an initial set of summarizable sentences of length $n$
\Require $k$, the length of desired summary
\Ensure $\Sigma^*$, the optimal summary of length $k$
\State \textbf{Compute the importance score} $\tau_I$ for $\Sigma$
\State \textbf{Select} $k$ sentences in $\Sigma$ that maximize $\tau_I$ and build $\Sigma^*$
\State $\text{OF}_{\max} = \text{OF}(\Sigma^*)$
\While {not stop-condition}
\For {each $s : s \in \Sigma \land s \notin \Sigma$}
\State $\Sigma \leftarrow \text{Mutation}(\Sigma, s)$
\EndFor
\For {each $\Sigma_h \in \Theta$}
\State $\text{OF}_h = \text{OF}(\Sigma_h)$
\EndFor
\State Select $\Sigma_h$ with maximum $\text{OF}_h$ value
\If {$\text{OF}_h(\Sigma_h) > \text{OF}_{\max}$}
\State $\Sigma_{\max} = \text{OF}(\Sigma_h)$
\State $\Sigma \leftarrow \Sigma_h$
\Else
\State Update stop-condition
\EndIf
\EndWhile
\State \Return $\Sigma$
\end{algorithmic}
\end{algorithm}

In particular, the \textit{mutation} operator generates, starting from a given summary $\Sigma$ of length $k$ and a sentence $s$, a set of $k$ summaries obtained by substituting each sentence of $\Sigma$ with the input sentence $s$. The mutated summary with the maximum value of objective function is accepted as new summary if it improves the current value of the objective function. In each current step of the algorithm the implementation of the mutation operator avoids to consider the summaries that have been already examined in past iterations.
The stop condition takes into account both the number of not-improving iterations and the maximum number of allowed iterations. It can be easily demonstrated that the complexity of the proposed algorithm is \(O(k \times (n - k))\).

3. RESULTS AND CONCLUSIONS

The goal of our experiments is to evaluate the goodness of automatically generated summaries with respect to a human ground truth.

To this aim, we asked a group of 15 people\(^1\) to generate, for four distinct queries (“The Hobbit movie by Gugilemo del Toro”, “The last album of Beyonce’ Knowles”, “The D-Day”, “Napoleon and his defeat in Waterloo”), 3 different summaries, each one respectively containing 5, 10 and 15 sentences, using the information coming from 2 general digital encyclopedia (wikipedia, encarta), 2 domain digital information sources (imdb and lastfm) for the first two topics and 4 news web-sites.

After this preliminary step and starting from the 180 obtained summaries, we have built, for each one of the discussed topics, 4 “optimal” summaries (composed by 5, 10 and 15 sentences), one for each query, by considering those sentences that have been more frequently chosen by humans. Then, such optimal summaries have been compared with those generated by the system in terms of the classical measure of precision.

The precision is defined as \(P = \frac{|A \cap H|}{|A|}\), \(P\) and \(H\) being the set of sentences generated by the system and human optimal summary, respectively. In table 1, we report the results that we obtained by varying the weights of importance, consistence and non-recurrence criteria. In particular, we have used three different kinds of weights configuration for managing the summary goodness criteria: high importance \((\alpha = 0.55, \beta = 0.25, \gamma = 0.20)\), high consistence \((\alpha = 0.3, \beta = 0.5, \gamma = 0.2)\) and high non-recurrence \((\alpha = 0.35, \beta = 0.20, \gamma = 0.45)\). In addition, the table reports computation times for producing the summaries w.r.t the summary lengths.

These configurations have been selected in the tuning step of the system: experimentally, these configurations obtain the better performances in term of Precision.

Note that we have the better results when the semantic expressed by a query is more detailed. In these cases the system seems to work better when the length of the summary increases. However, in all the cases the criterion that returns the best results is the importance.

Further works will be devoted to improve our work into main directions: i) extend the methodology to multimedia data; ii) design more detailed experiments and iii) compare the results with similar summarization systems.

---

\(^{1}\)The people involved in the experiments were mainly students from the University of Naples related to the database and programming courses.

<table>
<thead>
<tr>
<th>Query</th>
<th>(P) (summary length)</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Hobbit</td>
<td>5  10  15</td>
</tr>
<tr>
<td>high importance</td>
<td>0.6 0.6 0.87</td>
</tr>
<tr>
<td>high non-recurrence</td>
<td>0.4 0.6 0.87</td>
</tr>
<tr>
<td>high consistence</td>
<td>0.4 0.6 0.87</td>
</tr>
<tr>
<td>times (sec.)</td>
<td>0.25 5 7.5</td>
</tr>
<tr>
<td>Beyonce'</td>
<td>5  10  15</td>
</tr>
<tr>
<td>high importance</td>
<td>0.2 0.4 0.8</td>
</tr>
<tr>
<td>high non-recurrence</td>
<td>0.4 0.53 0.6</td>
</tr>
<tr>
<td>high consistence</td>
<td>0.2 0.4 0.8</td>
</tr>
<tr>
<td>times (sec.)</td>
<td>0.4 8 12</td>
</tr>
<tr>
<td>D-Day</td>
<td>5  10  15</td>
</tr>
<tr>
<td>high importance</td>
<td>0.2 0.3 0.6</td>
</tr>
<tr>
<td>high non-recurrence</td>
<td>0.4 0.4 0.5</td>
</tr>
<tr>
<td>high consistence</td>
<td>0.2 0.4 0.6</td>
</tr>
<tr>
<td>times (sec.)</td>
<td>0.4 8 12</td>
</tr>
<tr>
<td>Napoleon</td>
<td>5  10  15</td>
</tr>
<tr>
<td>high importance</td>
<td>0.53 0.6 0.7</td>
</tr>
<tr>
<td>high non-recurrence</td>
<td>0.4 0.53 0.6</td>
</tr>
<tr>
<td>high consistence</td>
<td>0.53 0.53 0.6</td>
</tr>
<tr>
<td>times (sec.)</td>
<td>0.5 10 15</td>
</tr>
</tbody>
</table>

Table 1. Precision and Time Results for summaries related to four topics: The Hobbit, Beyonce’, D-Day and Napoleon.

4. REFERENCES


