Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 38 (2014) 52 – 59

Procedia Computer Science

10th Italian Research Conference on Digital Libraries, IRCDL 2014

Ranking Sentences for Keyphrase Extraction: A Relational Data Mining Approach

Michelangelo Ceci[a,*], Corrado Loglisci[a], Lucrezia Macchia[a]

[a]*Dipartimento di Informatica, Università degli Studi di Bari "Aldo Moro", Bari, Italy*

Abstract

Document summarization involves reducing a text document into a short set of phrases or sentences that convey the main meaning of the text. In digital libraries, summaries can be used as concise descriptions which the user can read for a rapid comprehension of the retrieved documents. Most of the existing approaches rely on the classification algorithms which tend to generate "crisp" summaries, where the phrases are considered equally relevant and no information on their degree of importance or factor of significance is provided. Motivated by this, we present a probabilistic relational data mining method to model preference relations on sentences of document images. Preference relations are then used to rank the sentences which will form the final summary. We empirically evaluate the method on real document images.

Keywords: Document summarization; Ranking; Relational data mining

1. Introduction

The growing amount of documents available in digital libraries makes difficult and arduous obtaining the desired information, and therefore demands for the development of technologies to effectively support the user in a rapid comprehension once interesting documents have been retrieved. Numerous studies have been carried out in Natural Language Processing and, in particular, in the subfield of Automatic Text Summarization in order to generate a summarizing text which conveys the most salient and important information of the original document(s)[1]. A *summary* can be defined as a text that is produced from one or more texts, that contains a significant portion of the information in the original text(s), and that is no longer than half of the original text(s). Summaries can be categorized in *extracts*, when they are created by selecting the keyphrases of the original text, and *abstracts*, when they are created by inferring the meaning of the source document or by re-generating the content of it[2]. The techniques oriented to the generation of abstracts require linguistic knowledge and sophisticated resources, and perform a deep analysis of the textual content by taking into account typical language constructs, such as discourse structure. The techniques based on the extracts rather perform a shallow analysis of the text and do not require linguistic knowledge. Although the extract-based techniques can produce summaries with evident problems of interpretation and cohesion among the selected portions

[*] Corresponding author. Tel.: +39-0805442285; fax: +39-0805442285.
E-mail address: michelangelo.ceci@uniba.it (Michelangelo Ceci).

doi:10.1016/j.procs.2014.10.011

of text, they have been proven to yield summaries whose informative level is satisfactory. This is particularly true when the extract is used as a component of another system and is not directly used by humans.

A strategy widely investigated for extractive approaches, is that of selecting the more salient sentences through Machine Learning or Data Mining algorithms which aim at either recognizing and then classifying the sentences to be included in the summary or ranking the source sentences and then selecting those with highest rank[3]. Typically, sentences are described in terms of lexical and structural features (e.g., keywords frequency, title keywords, sentence location, indicator phrases, etc.[4]) and represented as vectors of quantitative and categorical measures of those features (attribute-value representation). However, in some practical applications, it is also possible to exploit additional information conveyed by the structure of the original document. For example, in the case of document images obtained by scanning paper documents, sentences can be related to layout components or paragraphs. Another example is that of semistructured documents such as XML/HTML documents where sentences can be related to sections. In such situations, the classical attribute-value representation (based on the single-table assumption[5]), according to which sentences would be represented in a single table of a relational database (each row represents a sentence and columns correspond to properties of the sentence) appears to be too restrictive for at least three reasons. First, sentences cannot be realistically considered independent observations, because their arrangement is mutually constrained. Second, relationships among sentences in the same paragraph cannot be properly represented by a fixed number of attributes in a table. Third, the representation of properties of objects related to sentences (such as layout components or sections) would lead to redundancy problems that cause changes in the underlying probability distribution of examples. Since the single-table assumption limits the representation of relationships between examples, it also prevents the discovery of this kind of patterns, which can be very useful in the context of document summarization.

In this paper we propose an extractive approach aiming at learning to rank the source sentences extracted from document images. The proposed approach overcomes limitations posed by the single-table assumption by resorting to the relational data mining setting[5] according to which data are represented in several tables of a relational database possibly related according to foreign key constraints. This allows us to distinguish between the *reference objects* of analysis (sentences) and other *task-relevant spatial objects* (e.g. layout components), and to represent their interactions. This also allows us to represent different entities in different ways: sentences can be represented exploiting lexical and structural properties while layout components, for example, can be represented according to geometrical properties.

2. Background and Related Works

Background of this work is the Document image analysis system WISDOM++ [1] that enables the transformation of document images into XML format by means of several complex steps[6]. Initial processing steps include binarization, skew detection, noise filtering, and segmentation. The document image is then decomposed into several constituent items which represent coherent components of the documents (e.g., text lines or halftone images), without any knowledge of the specific format. This layout analysis step precedes the interpretation or understanding of document images, whose aim is that of recognizing "logic components", that is, logically relevant layout components (e.g., title and section title of a scientific paper)[7] as well as extracting abstract relationships between layout components (e.g., reading order)[8]. By moving towards a higher level of abstraction, it is also possible to identify "semantic components" (e.g., motivations and experiments of a scientific paper) composed by several logic components (possibly belonging to different document pages) by exploiting their OCRed textual content. In this paper we add to WISDOM++ the keyphrase extraction step that is based on sentence ranking.

Sentence ranking is an approach investigated mostly with extract-based techniques which implement supervised Data Mining algorithms. This assumes the availability of a set of textual documents where ranking of the sentences of each document is given. Data Mining algorithms are then used to learn a ranking model to be applied on new documents. As in our case, in the literature, learning to rank from previously ranked sentences has been also interpreted as the problem of learning a preference function.

[1] http:/www.di.uniba.it/%7Emalerba/wisdom++/

Several supervised learning techniques for sentence ranking have already been reported in the literature. Xie et al.[9] propose a evolutionary algorithm to generate ranking functions. After having preprocessed the training documents into a set of sentence feature vector, the algorithm first produces different populations of ranking functions, then, with each of them, generates an extract which is evaluated w.r.t. objective summary through the cosine measure. The best function is then used to produce the next population of functions and so on until the stop criterion is satisfied: the final ranking function is used to rank new sentences. Svore et al.[10] resort to a neural network pair to generate a ranking composed of three sentences. More precisely, given a training set consisting of the sentences (whose features include also properties derived by third-party sources) and a set of three human-generated sentences, ranking function is learned on pairs of sentences ranked w.r.t. to the similarity with three human-generated sentences. The three highest ranked sentences (namely those more similar to the human extract) are then selected for the summary. Although all cited approaches allow us to obtain summaries with high informative level, as stated before, they suffer from limitation imposed by single-table assumption that, among others, does not guarantee an adequate representation of additional information conveyed by the possible structure of the original document.

3. Sentence Ranking

The problem of sentence ranking is solved by learning a preference model that leads to identify preference relations to be applied to new documents. In the proposed approach, the model is probabilistic and exploits relational patterns extracted from training data.

3.1. Mining Preference Relations

The problem of mining preference relations between sentences can be formalized as follows. *Given*:

- a database schema S with h relational tables $S = \{T_1, T_2, \ldots, T_h\}$
- two sets PK and FK of primary and foreign key constrains on tables in S
- a target relation $T \in S$ representing sentences that play the role of reference objects
- a precedence relation $PT \in S$ with two attributes. Each tuple in this table represents an ordered pair of reference objects where the first reference object precedes the second one.

Find: A probability estimation $P(a \prec b|a, b)$ for any couple of sentences a and b belonging to a new document represented according to the schema $S - PT$. Objects in $S - \{T, PT\}$ play the role of task relevant objects, while the precedence relation implicitly defines a partial ordering between two sentences.

It is noteworthy that in our approach, differently from other approaches, it is also possible to avoid to consider some sentences belonging to parts of the document that are not considered *relevant* for the task at hand (e.g. sentences in tables or sentences in references of a scientific paper). This means that the preference relation in training data does not necessarily express total ordering of sentences in training documents.

By applying the Bayes theorem, $P(a \prec b|a, b)$ can be computed as:

$$P(a \prec b|a, b) = P(a \prec b)P(a, b|a \prec b)/P(a, b) \tag{1}$$

where:

- $P(a \prec b)$ in (1) denotes the prior probability that a sentence precedes another. This probability might be different from 0.5, since, as stated before, training reference objects might not be totally ordered.
- $P(a, b) = P(a \prec b)P(a, b|a \prec b) + P(b \prec a)P(a, b|b \prec a)$

In order to simplify the estimation of the likelihood $P(a, b|a \prec b)$, conditional independence is assumed (*naïve Bayes assumption*), according to which $P(a, b|a \prec b)$ can be factorized as follows:

$$P(a, b|a \prec b) = P(a_1, \ldots, a_m, b_1, \ldots, b_m|a \prec b) = P(a_1|a \prec b) \times \ldots \times P(a_m|a \prec b) \times P(b_1|a \prec b) \times \ldots \times P(b_m|a \prec b)$$

where $a_1, \ldots, a_m$ represent the set of attribute values of a and $b_1, \ldots, b_m$ represent the set of attribute values of b. However, the formulation reported above for naïve Bayesian classifiers is clearly limited to attribute-value representations. In the case of relational data, some extensions are necessary. The basic idea is that of using a set of relational patterns $\mathfrak{R}(a, b)$ to describe the considered objects, and then to define a suitable decomposition of the likelihood *à la* naive Bayesian classifier to simplify the probability estimation problem.

Before describing how the likelihood is computed, it is necessary to provide a formal definition of relational pattern, which is a conjunction of unary and binary predicates[2] of two different types:

Definition 1 (Property predicate)**.** *A predicate $p/2$ is a property predicate associated to a table $T_i \in S - PT$ if the first argument of p represents the primary key of T_i and the second argument represents another attribute in T_i which is neither the primary key of T_i nor a foreign key.*

Definition 2 (Structural predicate)**.** *A predicate $p/2$ is a structural predicate associated to a table $T_i \in S - PT$ if a foreign key in $S - PT$ exists that connects T_i to a table $T_j \in S$. The first argument of p represents the primary key of T_j and the second argument represents the primary key of T_i.*

Definition 3 (Relational Pattern)**.** *A Relational Pattern is in the form:*

$$\langle S \rangle \{, \langle attr(A) \rangle\}_{0..1} \{, \langle attr(B) \rangle\}_{0..1} \quad \{, \langle rel(C_k, C_j) \rangle \{, \langle attr(C_j, v) \rangle\}_{0..*}\}_{0..*}$$

where $attr/1$ represents the predicate associated to the target relation T (the argument is the primary key of T), $rel/2$ represents a generic structural predicate, $attr/2$ represents a generic property predicate and S is in the form of preference(A,B). *A pattern P in this form is a relational pattern if the property of linkedness*[11] *is satisfied (e.g. each variable C_k or C_j should be linked to the variables A or B by means of structural predicates).*

The likelihood is then computed as follows: $P(a, b | a \prec b) = P(\bigwedge_{R_k \in \mathfrak{R}(a,b)} R_k | a \prec b)$ where $\mathfrak{R}(a, b)$ is the set of relational patterns that cover the tuple $(a, b) \in PT$. The coverage of (a, b) by a relational pattern $R_k \in \mathfrak{R}(a, b)$ demands for matching all variables in R_k against some tuples in the set of relations $S - PT$ according to foreign key constraints. The set $\mathfrak{R}(a, b)$ is a subset of the set $\mathfrak{R}$ of all possible relational patterns ($\mathfrak{R}(a, b) \subseteq \mathfrak{R}$) whose construction is explained in section 3.2.

The application of the classical naïve Bayes independence assumption to all literals in $\bigwedge_{R_k \in \mathfrak{R}(a,b)} R_k$ is not correct, since it may lead to underestimate the probabilities for the case of precedence relations for which several patterns are represented (see[12]). In fact, when working with redundant literals in F', $P(a, b | a \prec b)$ will approach zero. We solve this problem by exploiting the notion of factorization[13] that allows us to remove redundant literals. For this reason, we impose $P(a, b | a \prec b) = P(F | a \prec b)$ for any minimal factor F of F' and we compute this probability using the naïve Bayesian assumption on literals in F.

3.2. Patterns construction

The relational pattern discovery is performed by exploring level-by-level the lattice of relational patterns ordered according to a generality relation ($\geqslant$) between patterns. Formally, given two patterns $P1$ and $P2$, $P1 \geqslant P2$ denotes that $P1$ ($P2$) is more general (specific) than $P2$ ($P1$). Hence, the search proceeds from the most general pattern and iteratively alternates the candidate generation and candidate evaluation phases (levelwise). In[14], the authors propose an enhanced version of the level-wise method[15] to discover patterns from data in multiple tables of a relational database. Candidate patterns are searched in the space of linked relational patterns, which is structured according to the θ-subsumption generality order[16].

This makes possible to perform a levelwise exploration of the lattice of relational patterns ordered by θ-subsumption. In particular, patterns are discovered by generating the pattern space one level at a time starting from the most general pattern (the pattern that contains only the $preference/2$ predicate) and then by applying a breadth-first evaluation in the lattice of relational patterns ordered according to $\geqslant_\theta$.

Indeed, we are not interested in all possible patterns, but only in those satisfying the following property:

[2] Henceforth, "/n" indicates the predicate arity. Unary (binary) predicates are indicated as /1 (/2).

Algorithm 1 ranking identification algorithm

```
findranking ($G = \langle V, E \rangle$): Ranking L
L← $\emptyset$;
while ($\#L <> \#V$) do
    $L.add\left(\underset{b_i \in V/L}{\arg\max} \; SUMPREF_G(b_i)\right)$;
end while
```

$$(supp_{a<b}(P) > minSup \vee supp_{b<a}(P) > minSup) \wedge (GR_{a<b}(P) > minGR \vee GR_{b<a}(P) > minGR)$$

where: $minSup \in [0, 1)$ and $minGR \in [1, +\infty)$ are user defined thresholds; $supp_{a<b}(P)$ represents the support of the pattern P with respect to a preference relation; $GR_{a<b}(P)$ represents the growth rate computed as $supp_{a<b}(P)/supp_{b<a}(P)$.

This restriction of the search space permits us to apply different pruning criterion. The monotonicity property of the generality order $\geqslant_\theta$ with respect to the support value (i.e., a superset of an infrequent pattern cannot be frequent)[17] can be exploited to avoid generation of infrequent relational patterns. The monotonicity property does not hold for the growth rate: a refinement of a pattern whose growth rate is lower than the threshold *minGR* may or may not be a pattern with growth rate lower than *minGR*. However, the growth rate can be used for pruning as well. In particular, it is possible to stop the search when it is not possible to increase the growth rate with additional refinements[18]. Finally, as stopping criterion, the number of levels in the lattice to be explored can be limited by the user-defined parameter $MAX_L \geq 1$ which limits the maximum number of predicates in a candidate emerging pattern.

3.3. Ranking Reconstruction

In this step, the goal is to build a ranking of sentences (reference objects). Formally, *Given:* A database with schema $S - PT$ (the same schema used for training), *Find:* A total ordering of sentences in the target table T belonging to *relevant* semantic components.

The algorithm follows the proposal reported in[19] and we aim at iteratively evaluating the most promising sentence to be appended to the resulting ranking. Let:

- $G = \langle V, E \rangle$ be a *labeled* directed graph where $V = \{b \in T\}$ and $E = \{(a, b, w_{a,b}) \in V^2 \times [0, 1] | w_{a,b} = P(a \prec b | a, b)\}$ is the set of weighted edges where weights are the probabilities $P(a \prec b | a, b)$ computed according to (1),
- $SUMPREF_G : V \rightarrow [0, \#V]$ be a preference function defined as: $SUMPREF_G(a) = \sum_{b \in V, b \neq a} w_{a,b}$,

Algorithm 5 fully specifies the method for the ranking identification. The rationale is that at each step, a sentence is added to the final ranking. Such a sentence is that for which $SUMPREF_G(\_)$ is the highest. Higher values of $SUMPREF_G(\_)$ are given to sentences which have a high sum of probabilities to precede other sentences. Once the ranking of the sentences has been identified, the best *m* sentences are used to define the summary.

4. Data extraction and representation

Reference objects correspond to descriptions of sentences extracted from document images. The representation of the sentences is obtained by means of a phase of natural language processing which extracts sentence features. Extraction includes tokenization, sentence splitting, part-of-speech (POS) tagging, stop-word removing and stemming.

The execution in sequence of these techniques allows us to represent sentences in terms of the features:

- ADJECTIVE_POS_FREQUENCY, VERBALFORM_POS_FREQUENCY, NOUN_POS_FREQUENCY express the normalized frequency of the words of some POS categories included in the analyzed sentence w.r.t. the total set of words in the same sentence.
- TF_IDF_WORD1, ..., TF_IDF_WORDn denote the presence in the sentence of the *n* words having highest $tf - idf$ values over the training corpus.

- POSITION_INSIDE_DOCUMENT, POSITION_INSIDE_SECTION represent the normalized position of the sentence in the document and in the semantic component, respectively.

In addition, we also consider the presence of indicator phrases, typically used in discourse analysis, that give important information about the structure of the discourse[4]. Indicator phrases are expressed as a set of CUE_WORDs.

We use semantic components (SEMANTIC_COMPONENTS), logical components (BLOCKS) and the preference table (PREFERENCE). Logical components are described according to features that can be classified as: *Locational x_pos_centre*/*y_pos_centre*: position of the centroid of the logical component w.r.t. the x / y axis; *Geometrical height*/*width*: the height/width in pixels of a logical component; *Logical*: "logical label" associated to a logical component; *Content type type_of*: content type of a logical component (Possible values are: {image, text, horizontal line, vertical line, graphic, mixed}).

5. Experiments

We explored the applicability the proposed method to the domain of document image understanding in order to generate summaries from key-phrases found in the semantically relevant layout components.

Two datasets of document images were considered. The first dataset (denoted as **TPAMI**) is a set of twenty-three scientific papers published as either regular or short in the IEEE Transactions on Pattern Analysis and Machine Intelligence in the January and February issues of 1996 in the multi-column document. We processed 210 document images in all, an average number of 9,13 images per document. The second dataset (denoted as **ICML**) is a set of thirty scientific papers published as either regular or short in the International Conference on Machine Learning of 2009 in the multi-column format. We processed 240 document images in all, an average number of 8 images per document. Papers are processed in order to segment them, perform layout analysis, identify logic type of logical components and identify semantic components. Admissible semantic components are *abstract, method, motivation, experiment result, rejected* among them, in this work, relevant semantic components considered for summarization are *method* and *motivation*. We consider only *method* and *motivation* because these components typically report the main contribution of the paper. Figure 1 reports an illustration. Three sentences (denoted as a, b, d) are selected from the section of *Introduction*, which is recognized as the component *motivation*. While one sentence (denoted as c) is selected from the section of *Bayesian Networks and Probabilistic Inference* (recognized as the component *method*). The summary includes these four sentences ranked as a, b, c, d.

The preference relation is constructed by ranking the sentences contained in relevant semantic components on the basis of the abstract. In particular, the ranking used for training exploits the cosine similarity between the sentences in the *abstract* of the document wa_j and sentences in *method* and *motivation* semantic components w_k:

$$sim(wa_j, w_k) = \left(\sum_{i=1...n} wa_{j,i} \cdot w_{k,i}\right) / \left(\sqrt{\sum_{i=1...n} (wa_{j,i})^2} \cdot \sqrt{\sum_{i=1...n} (w_{k,i})^2}\right) \tag{2}$$

where each sentence (wa_j or w_k) is represented in form of a $tf - idf$ vector of n elements. The score used for ranking (and, then defining the training preference relation) is:

$$score(w_k) = max_j \ sim(wa_j, w_k)$$

We evaluated the results with two evaluation measures. The first measure is *ROUGE-N* and it is determined with the created summary and reference summary (abstract). It is implemented as

$$ROUGE - N = \frac{\sum_{n-gram \in Summary} Count_{match}(n - gram)}{\sum_{n-gram \in Abstract} Count(n - gram)} \tag{3}$$

where $Count_{match(n-gram)}$ is the maximum number of n-grams that co-occur in the $Summary$ and $Abstract$, $Count_{n-gram}$ is the count of the n-grams in the $Abstract$. We used *ROUGE-N* as *ROUGE-1*, namely, the 1-grams were considered. The second measure is the cosine similarity (as defined above) but it is determined with the created summary and reference summary (abstract).

Evaluation was performed by means of a six fold cross validation for **TPAMI** dataset and five fold cross validation for **ICML** dataset with the following setup: $minSup = 0.05$, $minGR = 1.1$, $MAX_L = 3$, $n = 10$ and $m = 10$.

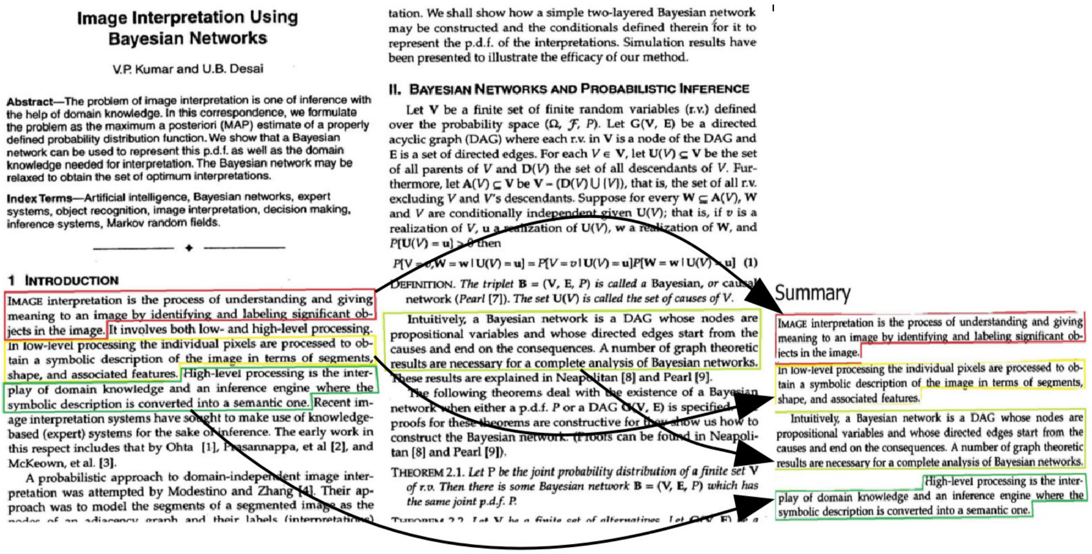

Image Interpretation Using Bayesian Networks

V.P. Kumar and U.B. Desai

Abstract—The problem of image interpretation is one of inference with the help of domain knowledge. In this correspondence, we formulate the problem as the maximum a posteriori (MAP) estimate of a properly defined probability distribution function. We show that a Bayesian network can be used to represent this p.d.f. as well as the domain knowledge needed for interpretation. The Bayesian network may be relaxed to obtain the set of optimum interpretations.

Index Terms—Artificial intelligence, Bayesian networks, expert systems, object recognition, image interpretation, decision making, inference systems, Markov random fields.

1 INTRODUCTION

IMAGE interpretation is the process of understanding and giving meaning to an image by identifying and labeling significant objects in the image. It involves both low- and high-level processing. In low-level processing the individual pixels are processed to obtain a symbolic description of the image in terms of segments, shape, and associated features. High-level processing is the interplay of domain knowledge and an inference engine where the symbolic description is converted into a semantic one. Recent image interpretation systems have sought to make use of knowledge-based (expert) systems for the sake of inference. The early work in this respect includes that by Ohta [1], Prasannappa, et al [2], and McKeown, et al. [3].

A probabilistic approach to domain-independent image interpretation was attempted by Modestino and Zhang [4]. Their approach was to model the segments of a segmented image as the nodes of an adjacency graph and their labels (interpretations)

tation. We shall show how a simple two-layered Bayesian network may be constructed and the conditionals defined therein for it to represent the p.d.f. of the interpretations. Simulation results have been presented to illustrate the efficacy of our method.

II. BAYESIAN NETWORKS AND PROBABILISTIC INFERENCE

Let **V** be a finite set of finite random variables (r.v.) defined over the probability space $(\Omega, \mathcal{F}, P)$. Let $\mathbf{G}(\mathbf{V}, \mathbf{E})$ be a directed acyclic graph (DAG) where each r.v. in **V** is a node of the DAG and **E** is a set of directed edges. For each $V \in \mathbf{V}$, let $\mathbf{U}(V) \subseteq \mathbf{V}$ be the set of all parents of V and $\mathbf{D}(V)$ the set of all descendants of V. Furthermore, let $\mathbf{A}(V) \subseteq \mathbf{V}$ be $\mathbf{V} - (\mathbf{D}(V) \cup \{V\})$, that is, the set of all r.v. excluding V and V's descendants. Suppose for every $\mathbf{W} \subseteq \mathbf{A}(V)$, **W** and V are conditionally independent given $\mathbf{U}(V)$; that is, if v is a realization of V, **u** a realization of $\mathbf{U}(V)$, **w** a realization of **W**, and $P[\mathbf{U}(V) = \mathbf{u}] > 0$ then

$$P[V = v, \mathbf{W} = \mathbf{w} \mid \mathbf{U}(V) = \mathbf{u}] = P[V = v \mid \mathbf{U}(V) = \mathbf{u}]P[\mathbf{W} = \mathbf{w} \mid \mathbf{U}(V) = \mathbf{u}] \quad (1)$$

DEFINITION. *The triplet* $\mathbf{B} = (\mathbf{V}, \mathbf{E}, P)$ *is called a* Bayesian, *or* causal network (*Pearl* [7]). *The set* $\mathbf{U}(V)$ *is called the set of causes of* V.

Intuitively, a Bayesian network is a DAG whose nodes are propositional variables and whose directed edges start from the causes and end on the consequences. A number of graph theoretic results are necessary for a complete analysis of Bayesian networks. These results are explained in Neapolitan [8] and Pearl [9].

The following theorems deal with the existence of a Bayesian network when either a p.d.f. P or a DAG $\mathbf{G}(\mathbf{V}, \mathbf{E})$ is specified. The proofs for these theorems are constructive for they show us how to construct the Bayesian network. (Proofs can be found in Neapolitan [8] and Pearl [9]).

THEOREM 2.1. *Let* P *be the joint probability distribution of a finite set* **V** *of r.v. Then there is some Bayesian network* $\mathbf{B} = (\mathbf{V}, \mathbf{E}, P)$ *which has the same joint p.d.f.* P.

THEOREM 2.2. *Let* **V** *be a finite set of alternatives. Let* $\mathbf{G}(\mathbf{V}, \mathbf{E})$ *be a*

Summary

IMAGE interpretation is the process of understanding and giving meaning to an image by identifying and labeling significant objects in the image.

In low-level processing the individual pixels are processed to obtain a symbolic description of the image in terms of segments, shape, and associated features.

Intuitively, a Bayesian network is a DAG whose nodes are propositional variables and whose directed edges start from the causes and end on the consequences. A number of graph theoretic results are necessary for a complete analysis of Bayesian networks.

High-level processing is the interplay of domain knowledge and an inference engine where the symbolic description is converted into a semantic one.

Fig. 1. A concrete illustration the document image of a scientific paper (left) and the corresponding ranked key-phrases (right)

Dataset	WISDOM++	GREEDYEXP	GREEDYUNIFORM	SVD	FURTHEST
TPAMI	82.19	79.52	79.63	80.61	82.26
ICML	61.25	59.39	60.54	53.53	54.68

Table 1. Cosine similarity computed between the abstracts and the generated summaries.

Dataset	WISDOM++	GREEDYEXP	GREEDYUNIFORM	SVD	FURTHEST
TPAMI	0.96	0.98	0.98	0.90	0.61
ICML	0.82	0.87	0.87	0.80	0.44

Table 2. Rouge-1 computed between the abstracts and the generated summaries.

Comparisons was performed between the WISDOM++ and several unsupervised techniques[20] with respect to the two evaluation measures.

The results in Table 1 show better performances of our solution, while the results in Table 2 reveal a behaviour comparable with the best competitors. It is worth of noting the difference between the two measures. When using the cosine similarity, we evaluate frequency-based quantities of the words representative of the documents. So, an high value of the similarity indicates that the summary contains sentences where representative words occur. Differently, Rouge-1, accounts the presence of the same 1-ngrams which could be even few representative for the content of the document. This denotes the capacity of the proposed solutions to produce summary with phrases which are really salient for the meaning of the document.

6. Conclusions

In this paper we propose an extractive approach aiming at learning to rank the source sentences extracted from document images. The proposed approach resorts to the relational data mining setting in order to adequately exploit lexical properties of the text as well as structural, logical and semantic properties conveyed by the nature of the original documents. The method is based on a probabilistic learner that makes use of discovered relational patterns. Experiments on real-world datasets and comparisons with other techniques prove the effectiveness of the proposed approach. For future work, we plan two research directions. The first one is to apply the proposed method on a large corpus (not necessarily scientific documents). In the second one, we intend to consider the opportunity of automatically defining the optimal number of sentences to be included in the summary. To this purpose, automatic threshold determination algorithms can be used.

Acknowledgements

This work fulfils the research objectives of the PON020056-33489339 project ”PUGLIA@SERVICE - Internet-based Service Engineering enabling Smart Territory structural development” and the PON 020056-33470993 project ”VINCENTE - A Virtual collective INtelligenCe ENvironment to develop sustainable Technology Entrepreneurship ecosystems” both funded by the Italian Ministry of University and Research.

References

1. Jones, S., Paynter, G.W.. Automatic extraction of document keyphrases for use in digital libraries: Evaluation and applications. *JASIST* 2002;**53**(8):653–677.
2. Jackson, P.. The oxford handbook of computational linguistics edited by ruslan mitkov. *Computational Linguistics* 2004;**30**(1):103–106.
3. Spärck Jones, K.. Automatic summarising: The state of the art. *Inf Process Manage* 2007;**43**(6):1449–1481. doi:http://dx.doi.org/10.1016/j.ipm.2007.03.009.
4. Paice, C.D.. Constructing literature abstracts by computer: Techniques and prospects. *Inf Process Manage* 1990;**26**(1):171–186.
5. Džeroski, S., Lavrač, N.. *Relational Data Mining*. Springer-Verlag; 2001.
6. Ceci, M., Loglisci, C., Ferilli, S., Malerba, D.. Project d.a.m.a.: Document acquisition, management and archiving. In: Agosti, M., Esposito, F., Meghini, C., Orio, N., editors. *IRCDL*; vol. 249 of *Communications in Computer and Information Science*. Springer. ISBN 978-3-642-27301-8; 2011, p. 115–118.
7. Ceci, M., Berardi, M., Malerba, D.. Relational data mining and ILP for document image understanding. *Applied Artificial Intelligence* 2007;**21**(4&5):317–342.
8. Ceci, M., Berardi, M., Porcelli, G., Malerba, D.. A data mining approach to reading order detection. In: *ICDAR*. IEEE Computer Society. ISBN 978-0-7695-2822-9; 2007, p. 924–928.
9. Zhuli, X., Li, X., Barbara, D.E., Peter, N., Weimin, X., Thomas, T.. Using gene expression programming to construct sentence ranking functions for text summarization. In: *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics; 2004, p. 1381. doi:http://dx.doi.org/10.3115/1220355.1220557.
10. Svore, K., Vanderwende, L., Burges, C.. Enhancing single-document summarization by combining ranknet and third-party sources. In: *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Association for Computational Linguistics; 2007, .
11. Helft, N.. *Progress in Machine Learning*; chap. Inductive generalization: a logical framework. Sigma Press; 1987, p. 149–157.
12. Ceci, M., Appice, A., Malerba, D.. Emerging pattern based classification in relational data mining. In: Bhowmick, S.S., Küng, J., Wagner, R., editors. *DEXA*; vol. 5181 of *Lecture Notes in Computer Science*. Springer. ISBN 978-3-540-85653-5; 2008, p. 283–296.
13. Robinson, J.A.. A machine oriented logic based on the resolution principle. *Journal of the ACM* 1965;**12**:23–41.
14. Ceci, M., Appice, A., Malerba, D.. Discovering emerging patterns in spatial databases: A multi-relational approach. In: Kok, J.N., Koronacki, J., de Mántaras, R.L., Matwin, S., Mladenic, D., Skowron, A., editors. *PKDD*; vol. 4702 of *Lecture Notes in Computer Science*. Springer. ISBN 978-3-540-74975-2; 2007, p. 390–397.
15. Mannila, H., Toivonen, H.. Levelwise search and borders of theories in knowledge discovery. *Data Mining and Knowledge Discovery* 1997; **1**(3):241–258.
16. Plotkin, G.D.. A note on inductive generalization. *Machine Intelligence* 1970;**5**:153–163.
17. Agrawal, R., Imielinski, T., Swami, A.N.. Mining association rules between sets of items in large databases. In: Buneman, P., Jajodia, S., editors. *International Conference on Management of Data*. 1993, p. 207–216.
18. Appice, A., Ceci, M., Malgieri, C., Malerba, D.. Discovering relational emerging patterns. In: Basili, R., Pazienza, M., editors. *AI*IA 2007: Artificial Intelligence and Human-Oriented Computing*; vol. 4733 of *LNAI*. GERMANY: Springer. ISBN 978-3-540-74781-9; 2007, p. 206–217.
19. Kamishima, T., Akaho, S.. Learning from order examples. In: *ICDM*. IEEE Computer Society. ISBN 0-7695-1754-4; 2002, p. 645–648.
20. Liu, K., Terzi, E., Grandison, T.. Manyaspects: a system for highlighting diverse concepts in documents. *PVLDB* 2008;**1**(2):1444–1447.