Automatic Text Summarization with Genetic Algorithm-Based Attribute Selection

Carlos N. Silla Jr.[1], Gisele L.Pappa[2], Alex A. Freitas[2], Celso A.A. Kaestner[1]

[1] Pontifícia Universidade Católica do Paraná (PUCPR)
Av. Imaculada Conceição 1155, 80215-901. Curitiba, PR, Brazil
{silla; kaestner}@ppgia.pucpr.br
[2] Computing Laboratory, University of Kent
Canterbury, CT2 7NF, UK
{glp6; A.A.Freitas}@kent.ac.uk

Abstract. The task of automatic text summarization consists of generating a summary of the original text that allows the user to obtain the main pieces of information available in that text, but with a much shorter reading time. This is an increasingly important task in the current era of information overload, given the huge amount of text available in documents. In this paper the automatic text summarization is cast as a classification (supervised learning) problem, so that machine learning-oriented classification methods are used to produce summaries for documents based on a set of attributes describing those documents. The goal of the paper is to investigate the effectiveness of Genetic Algorithm (GA)-based attribute selection in improving the performance of classification algorithms solving the automatic text summarization task. Computational results are reported for experiments with a document base formed by news extracted from *The Wall Street Journal* of the TIPSTER collection –a collection that is often used as a benchmark in the text summarization literature.

1 Introduction

We are surely living in an era of information overload. Recent studies published by the University of Berkeley [8] indicate that in 2002 about 5 million *terabytes* of information were produced (in films, printed media or magnetic/optic storage media). This number is equivalent to twice as much the corresponding number for 1999, which indicates a growth rate of about 30% per annum. The *Web* alone contains about 170 *terabytes*, which is roughly 17 times the size of the printed material in the USA's Congress Library.

On the other hand, it is very difficult to use the available information. Many problems – such as the search for information sources, the retrieval/extraction of information and the automatic summarization of texts – became important research topics in Computer Science. The use of automatic tools for the treatment of information became essential to the user, because without those tools it is virtually impossible to exploit all the relevant information available in the *Web* [22].

In this scenario, the task of automatic text summarization is very important. The goal of an automatic text summarization system is to generate a summary of the original text that allows the user to obtain the main pieces of information available in that text, but with a much shorter reading time [12]. The summaries are produced based on attributes (or features) that are usually derived empirically, by using statistical and/or computational linguistics methods. The values of these attributes are derived from the original text, and the summaries typically have 10%-30% of the size of the original text [11].

One of the approaches that has been recently used to perform automatic text summarization is the use of Machine Learning methods [13]. In this context automatic text summarization is cast as a classification (supervised learning) task [5], [6], as will be discussed in Section 3. Other approaches for text summarization (which do not involve machine learning) are described in [11], [12].

In addition, an important data preprocessing task for effective classification is the attribute selection task, which consists of selecting the most relevant attributes for classification purposes [7]. This task is important because many original attributes can be irrelevant for classification, in which case their removal tends to improve the performance of the classification algorithm. Furthermore, attribute selection reduces the processing time taken by the classification algorithm, and it can also lead to the discovery of smaller, simpler classification models (e.g. smaller decision trees, as observed in this paper).

The goal of the paper is to investigate the effectiveness of Genetic Algorithm (GA)-based attribute selection in improving the performance of classification algorithms solving the automatic text summarization task. GAs have been chosen as the attribute selection methods because they have been very successful in this data preprocessing task [3], [2]. This is mainly due to their ability to cope well with attribute interaction (which is the crucial problem in attribute selection) [2]. More precisely, this paper investigates the effectiveness of two GAs for attribute selection in improving the performance of two different kinds of classification algorithms – viz. a decision tree-induction algorithm and the Naïve Bayes classifier.

The remainder of this paper is organized as follows. Section 2 discusses GA-based attribute selection. Section 3 describes the *ClassSumm* system for summarization cast as a classification problem. Section 4 reports computational results. Finally, Section 5 presents the conclusions and discusses future work.

2 Attribute Selection with a Multi-Objective Genetic Algorithm

Attribute selection is one of the most important tasks that precedes the application of data mining algorithms to real world databases [7]. It consists of selecting a subset of attributes relevant to the target data mining task, out of all original attributes. In this paper, the target task is classification, and one attribute is considered relevant if it is useful for discriminating examples belonging to different classes.

Attribute selection algorithms they differ from each other in two main components: the kind of search method they use to generate candidate attribute subsets and the way they evaluate the quality of a candidate attribute subset. The search methods can be

classified in three main classes: exponential (e.g exhaustive search), randomised (e.g. genetic algorithms) and sequential (e. g. forward and backward sequential selection [7]) methods. In this paper we are interested in genetic algorithms (GA), since they are a robust search method, capable of effectively exploring large search spaces - which is usually the case in attribute selection. They also have the advantage of performing a global search – unlike many greedy, local search algorithms. In the context of data mining, this global search means that GAs tend to cope better with attribute interaction than greedy search methods [2].

The evaluation of the quality of each candidate solution can be based on two approaches: the filter and the wrapper approach. In essence, in the wrapper approach the attribute selection method uses the classification algorithm (as a black box) to evaluate the quality of a candidate attribute subset. In the filter approach the attribute selection method does not use the classification algorithm. We use the wrapper approach, because it tends to maximize predictive accuracy. (Note that this approach has the disadvantage of being significantly slower than the filter approach.)

The attribute selection task usually involves the optimisation of more than one objective, e.g. the predictive accuracy and the comprehensibility of the discovered knowledge. This is a challenging problem, because the objectives to be optimised can be conflicting with one another and they normally are non-commensurable – i.e., they measure different aspects of the target problem.

In the multi-objective optimisation framework [1], when many objectives are optimised there is no single best solution. Rather, there is a set of optimal solutions, each one involving a certain trade-off among the objectives. Multi-objective optimisation is based on Pareto dominance, that is: a solution S_1 dominates another solution S_2 iff S_1 is not worse than S_2 w.r.t. any objective and S_1 is strictly better than S_2 w.r.t. at least one objective. In multi-objective optimisation the system searches for non-dominated solutions.

MOGA is a Multi-Objective Genetic Algorithm designed to select attribute subsets for classification. It follows the basic ideas of GAs, i.e., it evolves a population of individuals, where each individual is a candidate solution to a given problem. In MOGA, each individual consists of M genes, where M is the number of original attributes in the data being mined. Each gene can assume values 0 or 1, indicating the absence or presence of the corresponding attribute in the selected subset of attributes.

Each individual is evaluated by a fitness function, which measures the quality of its attribute subset. At each generation (iteration) the fittest (the best) individuals of the current population survive and produce offspring resembling them, so that the population gradually contains fitter and fitter individuals – i.e., better and better candidate solutions to the underlying problem. The fitness function of MOGA is based on the wrapper approach, and involves the minimisation of both the classification error rate and the size of the decision tree built by J4.8[21]. MOGA searches for non-dominated solutions w.r.t. these two objectives. The version used in this paper returns, as the selected attribute subset, the non-dominated solution which dominates the largest number of solutions in the last generation. For more details about MOGA the reader is referred to [15],[16].

3 The ClassSumm System for Text Summarization

The *ClassSumm (Classification-based Summarization)* system, proposed by [5], [6], is a system for automatic text summarization based on the idea of casting that task as a classification task and then using corresponding Machine Learning methods.

The system consists of the following main steps:

(1) the system extracts the individual sentences of the original documents, using one the approches analysed in [18], in this work it was used the regular expression approach;

(2) each sentence is associated with a vector of predictor attributes (features), whose values are derived from the content of the sentence;

(3) each sentence is also associated with one of the following two classes: *Summary* (i.e., the sentence belongs to the summary) or *Not-Summary* (i.e., the sentence does not belong to the summary).

This procedure allows us to cast text summarization as a classification, supervised learning problem. As usual in the classification task, the goal of the classification algorithm is to discover, from the data, a relationship (say, an IF-THEN classification rule) that predicts the correct value of the class for each sentence based on the values of the predictor attribute for that sentence. More precisely, this casting leads to the following steps for solving a text summarization problem:

(1) The system constructs a training set where each example (record) corresponds to a sentence of the original documents, and each example is represented by a set of attribute values and a known class.

(2) A classification algorithm is trained to predict each sentence's class (*Summary* or *Not-Summary*) based on its attribute values.

(3) Given a new set of documents, the system produces a test set with predictor attributes in the same format as the training set. However, the values of the classes are unknown in the test set.

(4) Each sentence in the test set is classified, by the trained algorithm produced in step (2), in one of the two classes: *Summary* or *Not-Summary*.

Note that this procedure does not take into account the size of the summary to be generated. In practice the user often wants a summary of a specified size – in terms of percentage of the original document size. In order to take this into account, one uses a classification algorithm that, instead of directly predicting the class of each sentence, assigns to each sentence a measure of the relevance of that sentence for the summary. This produces a ranking of the sentences. Then the top *N* sentences in that ranking are assigned the class *Summary* and all the other sentences are assigned the class *Not-Summary*, where *N* is a user-specified parameter.

The classification algorithms used in the current version of ClassSumm are Naïve Bayes [13] and C4.5 [17]. In the former the relevance of a sentence for the summary is directly obtained from the conditional probability of the class *Summary* given the attribute values in the sentence. In the case of C4.5 the relevance of a sentence for the summary is obtained from the confidence factor associated with each leaf node of the induced tree.

The attributes used by ClassSumm can be categorized into two broad groups: shallow and deep attributes. Shallow attributes are based on heuristics and statistical

methods; whereas deep attributes are based on linguistic knowledge. Both kinds of attributes are used in this paper. This work focuses on English texts only.

As usual in text processing systems, a preliminary preprocessing phase is performed [19]. This phase consists of four steps:

(1) identifying the sentences of the document; (2) converting all characters to lower case (*case folding*); (3) removing very common words (*stop words*) which do not contribute to the meaning of the text – e.g., "the", "a", etc.; (4) removing suffixes (i.e., performing *stemming*), so that words such as "learned" and "learning" are converted to the standard form "learn". These preprocessing steps help to significantly reduce the number of words, which is very important to improve the cost-effectiveness of automatic text summarization.

After this preprocesing, each sentence of the document is represented by an attribute vector consisting of the following elements:

1. Position: indicates the position of the sentence in the text, in terms of percentile, as proposed by Nevill-Manning [14];
2. Size: indicates the number of terms (words) in the sentence;
3. Average-TF-ISF: the TF-ISF (term frequency – inverse sentence frequency) measure [4] is a variation of the TF-IDF measure [20] widely used in information retrieval. (The difference between the two measures is explained in detail in [4].) The value of TF-ISF for each term of a sentence is computed, and the value of the Average-TS-ISF attribute for that sentence is the average value over the TF-ISF values for all the terms in that sentence;
4. Similarity to Title: The computation of this measure is based on the vectorial representation of the document, where each sentence is represented by a vector formed by its terms [20]. Initially the title of the document is preprocessed, forming a vector of terms, and then the similarity between each sentence and the title of the document is calculated by the co-sine measure [20];
5. Similarity to Keywords: Analogously to the previous attribute, this attribute is computed by using the vectorial representation of the document and calculating the similarity between each sentence and the vector of keywords by using the co-sine measure. This assumes the document has a set of author-provided keywords, which is the case in this work;
6. Cohesion w.r.t. All Other Sentences: This attribute is computed by calculating the distance between a sentence and every other sentence in the document. The sum of all those distances is the value of this attribute for the sentence in question;
7. Cohesion w.r.t. the Centroid: First the system computes the centroid of the document, which is simply a vector consisting of the arithmetic means of all sentence vectors' elements. Then the value of this attribute for a given sentence is computed by calculating the similarity between the sentence and the centroid vector – again, using the co-sine measure.

The next two attributes use a kind of linguistic structure built as an approximation to the text's rhetorical tree. This structure is obtained by running a hierarchical clustering algorithm, which forms clusters of similar sentences based on the vectorial representation of the sentences. The output of the clustering algorithm is a clustering tree where the leaf nodes are sentences and internal nodes represent clusters that have

more and more sentences as the root of the tree is approached. The root of the tree represents a single cluster with all sentences in the document. The similarity measure used by the clustering algorithm is, again, the co-sine measure. Once a clustering tree has been produced by the hierarchical clustering algorithm, that tree is used to compute the following attributes for each sentence:

8. The depth of the sentence in the tree, i.e, the number of nodes that are ancestors of the leaf node representing that sentence.
9. The direction of the sentence in the tree, computed by following the path from the root towards the sentence up to depth four. At each depth level the direction can be *Left*, *Right* ou *None* (in case the current level is greater than the level of the sentence). This produces four attributes, each with one direction value. These attributes indicate the approximate position of the sentence in the rhetorical tree, incorporating linguistic knowledge into the set of predictor attributes.

The following attributes are obtained from the original text before the application of the preprocessing phase, and they also incorporate linguistic knowledge into the set of predictor attributes.

10. Indicators of Main Concepts: these indicators are computed by using a morphological *part-of-speech tagger* that identifies nouns in the document. The motivation for focusing on nouns is that they tend to be more meaningful (at least as individual words) than other part-of-speech classes. The 15 most frequent nouns in the document are selected to be the indicators of main concepts. For each sentence, the value of this attribute is true if the sentence contains at least one of those 15 indicators, and false otherwise.
11. Presence of Anaphors: From a linguistic point of view, the presence of anaphors in a sentence usually indicates that the information in the sentence is not essential, being used only to complement the information in a more relevant sentence. In ClassSumm the anaphors are identified by using a fixed list of words indicating anaphors. For each sentence, the value of this attribute is true if at least one of the first six words of the sentence is one of the words in the anaphor list, and false otherwise.
12. Presence of Proper Nouns: This attribute is computed directly from the output of a *part-of-speech tagger*. The value of the attribute is true if the sentence contains at least one proper noun, and false otherwise.
13. Presence of Discourse Markers: Some discourse markers, such as *because*, *furthermore*, also tend to indicate the presence of non-essential information. Discourse markers are identified by using a fixed list of words. The value of this attribute is true if the sentence contains at least one word in the list of discourse markers, and false otherwise.

Before the classification algorithm is applied to the training set, all the above non-binary attributes are normalized to the range [0..1] and then discretized. We adopt a simple “class-blind” discretization method, which consists of separating the original values into equal-width intervals; this procedure has produced good results in our previous experiments [5].

4 Computational Results

Previous work has reported results comparing ClassSumm with other Summarization methods [5], [6]. In those previous projects all original attributes were used. This paper focuses on a different issue. It investigates whether the performance of ClassSumm can be improved by using sophisticated attribute selection methods in a preprocessing step. An attribute selection method outputs only a subset of relevant attributes to be given to the classification algorithm, which hopefully will increase the predictive accuracy of the classification algorithm – which is also the accuracy of the decisions about which sentences should be included in the summary. The attribute selection methods used here are two kinds of Genetic Algorithms, namely a single-objective and a Multi-Objective Genetic Algorithm (MOGA) described in Section 2.

Experiments were carried out with a document base formed by news extracted from *The Wall Street Journal* of the TIPSTER collection [10]. This collection is often used as a benchmark in the text summarization literature.

For each document, a summary was produced using one of the following two approaches: (1) An automatically-generated summary, formed by the document's sentences that are most similar (according to the co-sine measure) to the summary provided by the author of the text, following the procedure proposed by Mani and Bloedorn [9]. This kind of summary is called an "ideal automatic summary". (2) A manually-generated summary, produced by an English teacher by selecting the most relevant sentences of the text. This is called an "ideal manual summary".

In all the experiments the training set consisted of 100 documents with their respective ideal automatic summaries. Experiments were carried out with two different kinds of test set. More precisely, in one experiment the test set consisted of 100 documents with their respective ideal automatic summaries, and in another experiment the test set consisted of 30 documents with their ideal manual summaries. In all experiments the training set and the test set were, of course, disjoint sets of documents, since the goal is to measure the predictive accuracy (generalisation ability) in the test set, containing only examples unseen during training.

In order to evaluate how effective Genetic Algorithm (GA)-based attribute selection is in improving the predictive accuracy of ClassSumm, two kinds of GAs for attribute selection have been used – both of them following the wrapper approach. The first one was the Multi-Objective GA (MOGA) discussed in Section 2. MOGA was used to select attributes for J4.8, a well-known decision-tree induction algorithm [21]. Recall that MOGA performs a multi-objective optimisation (in the Pareto sense) of both J4.8's error rate and the decision tree size. The results of training J4.8 with the attributes selected by the MOGA were compared with the results of training J4.8 with all original attributes, as a control experiment.

The second kind of GA used in the experiments was a simpler GA, called Single-Objetive GA (SOGA). It optimises only the error rate of a classification algorithm. SOGA was implemented directly from the MOGA implementation, by simply modifying MOGA's fitness function and selection method to optimise a single objective. Due to the focus on a single objective, the classifier used in this experiments was Naïve Bayes, whose measure of performance involves only error rate (no measure of size of the induced model). Again, the results of training Naïve

Bayes with the attributes selected by the SOGA were compared with the results of training Naïve Bayes with all original attributes, as a control experiment.

The results are reported in Tables 1 and 2, which refer to the results for the test sets containing ideal automatic summaries and ideal manual summaries (as explained earlier), respectively. Each of these tables reports results for two kinds of summary size (10% and 20% of the original document). Finally, for each kind of test set and each summary size, the results of four methods are compared – two methods using the classification algorithms (J4.8 and Naïve Bayes) with all attributes and two methods using those algorithms with GA-based attribute selection, as explained above. In the experiments with J4.8 the reported results include the accuracy in the test set (in the range [0..1]), the decision tree size (number of tree nodes), and the number of selected attributes (or "all" – i.e., 16 attributes – when no attribute selection was done). In the experiments with Naïve Bayes, of course only the accuracy and number of selected attributes are reported.

Table 1: Results for test set containing "ideal" automatic summaries

Summary Size = 10% of original document			
Method	**Accuracy**	**Tree size**	**# selected attrib.**
J4.8	0.18	42	All
MOGA–J4.8	0.33	7	4
Naïve Bayes	0.39	N/a	All
SOGA–Naïve Bayes	0.38	N/a	9
Summary Size = 20% of original document			
Method	**Accuracy**	**Tree Size**	**# selected attrib.**
J4.8	0.44	164	All
MOGA–J4.8	0.47	4	2
Naïve Bayes	0.51	N/a	All
SOGA–Naïve Bayes	0.52	N/a	11

Table 2: Results for test set containing "ideal" manual summaries

Summary Size = 10% of original document			
Method	**Accuracy**	**Tree Size**	**# selected attrib.**
J4.8	0.15	42	All
MOGA–J4.8	0.25	7	3
Naïve Bayes	0.23	N/a	All
SOGA–Naïve Bayes	0.22	N/a	12
Summary Size = 20% of original document			
Method	**Accuracy**	**Tree Size**	**# selected attrib.**
J4.8	0.33	164	All
MOGA–J4.8	0.35	4	1
Naïve Bayes	0.36	N/a	All
SOGA–Naïve Bayes	0.35	N/a	11

Several trends in the results can be observed in Tables 1 and 2. First, as expected, in both Table 1 and Table 2 the accuracy associated with the larger summaries (20%

of original document) is considerably larger than the accuracy associated with the smaller summaries (10% of the original document). This reflects the fact that, as the size of the summary increases, the classification problem becomes easier – e.g., the class distribution becomes less unbalanced (i.e, closer to a 50-50% class distribution).

Second, the GA-based attribute selection procedure had different effects on the performance of ClassSumm, depending on the kind of GA and classifier used in the experiments. The use of MOGA-based attribute selection led to an increase in J4.8's accuracy. This increase was very substantial in the smaller (10%) summaries, but relatively small in the larger (20%) summaries. In addition, MOGA-based attribute selection was very effective in selecting a very small number of attributes, which led to a very significant reduction in the size of the induced decision tree. This significantly improves the comprehensibility of discovered knowledge, an important goal in data mining [2], [21].

On the other hand, the effect of SOGA-based attribute selection in Naïve Bayes' accuracy was not so good. The effect was very small and, overall, even slightly negative.

These results for the two kinds of GA-based attribute selection are qualitatively similar in both Table 1 and Table 2, so that they are independent from whether the test set contains automatic summaries or manual summaries.

5 Conclusions and Future Work

As mentioned earlier, the goal of this paper was to investigate the effectiveness of Genetic Algorithm (GA)-based attribute selection in improving the performance of classification algorithms solving the automatic text summarization task. Overall, the two main conclusions of this investigation were as follows.

First, the Multi-Objective GA (MOGA) was quite effective. It led to an increase in the accuracy rate of the decision tree-induction algorithm used as a classifier, with a corresponding increase in the accuracy of the text summarization system. It also led to a very significant reduction in the size of the induced decision tree. Hence, the multi-objective component of the GA, which aims at optimising both accuracy and tree size, is working well.

Second, the Single-Objective GA (SOGA), which aimed at optimising classification accuracy only, was not effective. Surprisingly, there was no significant difference in the results of Naïve Bayes with all attributes and the results of Naïve Bayes with only the attributes selected by this GA. This indicates that all the original attributes seem more or less equally relevant for the Naïve Bayes classifier.

It should be noted that there is a lot of room for improvement in the results of the system, since the largest accuracy rate reported in Tables 1 and 2 was only 52%. Despite all the effort put into the design and computation of the 16 predictor attributes (which involve not only heuristic and statistical indicators, but also several relatively sophisticated linguistic concepts – e.g. a rhetorical tree), the current attribute set still seems to have a relatively limited predictive power. This suggests that future work could focus on designing an extended set of predictor attributes with more predictive power than the current one. Considering the difficult of doing this in a manual fashion, one interesting possibility is to use attribute *construction* methods to automatically create a better set of predictor attributes.

References

1. Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons (2001)
2. Freitas, A. A.: Data Mining and Knowledge Discovery with Evolutionary Algorithms. Springer (2002)
3. Kudo, M., Sklansky, J.: Comparison of algorithms that select features for pattern classifiers. Patten Recognition 33 (2000) 25–41
4. Larocca Neto, J., Santos, A.D., Kaestner, C.A.A., Freitas, A.A.: Document clustering and text summarization. In: Proc. 4th Int. Conf. Pratical Applications of Knowledge Discovery and Data Mining. (2000) 41–55
5. Larroca Neto, J., Freitas, A.A., Kaestner, C.A.A.: Automatic text summarization using a machine learning approach. In: XVI Brazilian Symposium on Artificial Intelligence. Number 2057 in Lecture Notes in Artificial Intelligence, Springer (2002) 205–215
6. Larroca Neto, J.: A Contribution to the Study of Automatic Text Summarization Techniques (in Portuguese). Master's thesis, Pontífica Universidade Católica do Paraná (PUC-PR), Graduate Program in Applied Computer Science. (2002)
7. Liu, H., Motoda, H.: Feature Selection for Knowledge Discovery and Data Mining. Kluwer Academic Publishers (1998)
8. Lyman, P., Varian, H.R.: How much information. (Retrieved from http://www.sims.berkeley.edu/how-much-info-2003 on [01/19/2004]
9. Mani, I., Bloedorn, E.: Machine learning of generic and user-focused summarization. In: Proc. of the 15th National Conf. on Artificial Intelligence (AAI 98). (1998) 821–826
10. Mani, I., House, D., Klein, G., Hirschman, L., Obrsl, L., Firmin, T., Chrzanowski, M., Sundeheim, B.: The tipster summac text summarization evaluation. MITRE Technical Report MTR 92W0000138, The MITRE Corporation (1998)
11. Mani, I., Maybury, M.T.: Advances in Automatic Text Summarization. MIT Press (1999)
12. Mani, I.: Automatic Summarization. John Benjamins Publishing Company (2001)
13. Mitchell, T.M.: Machine Learning. McGraw-Hill (1997)
14. Nevill-Manning, C.G., Witten, I.H., Paynter, G.W., Frank, E., Gutwin, C.: KEA: Pratical Automatic Keyphrase Extraction. ACM DL 1999 (1999) 245–255
15. Pappa, G.L., Freitas, A.A., Kaestner, C.A.A.: Attribute selection with a multiobjective genetic algorithm. In: XVI Brazilian Symposium on Artificial Intelligence. Number 2057 in Lecture Notes in Artificial Intelligence, Springer (2002) 280–290
16. Pappa, G.L., Freitas, A.A., Kaestner, C.A.A.: A multi-objective genetic algorithm for attribute selecion. In: Proc. 4th Int. Conf. on Recent Advances in Soft Computing (RASC-2002), University of Nottingham, UK (2002) 116–121
17. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann (1993)
18. Silla Jr., C. N., Kaestner, C.A.A.: An Analysis of Sentence Boundary Detection Systems for English and Portuguese Documents. In: 5th International Conf. on Intelligent Text Processing and Computational Linguistics. Number 2945 in Lecture Notes in Computer Science, Springer (2004) 135–141
19. Sparck-Jones, K.: Automatic summarizing: factors and directions. In Mani, I.; Maybury, M. Advances in Automatic Text Summarization. The MIT Press (1999) 1–12
20. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Information Processing and Management 24 (1988) 513–523
21. Witten, I.H., Frank, B.: Data Mining: Pratical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann, San Francisco (1999)
22. Zhong, N., Liu, J., Yao, Y.: In search of the wisdom web. IEEE Computer 35(1) (2002) 27–31