A Genetic Fuzzy Automatic Text Summarizer

Daniel Leite, Lucia H. M. Rino

Departamento de Computação – Universidade Federal de São Carlos (UFSCar)
CP 676, 13565-905 São Carlos, SP, Brazil
Núcleo Interinstitucional de Lingüística Computacional - http://www.nilc.icmc.usp.br

{daniel_leite,lucia}@dc.ufscar.br

Abstract. *In this paper we report on a fuzzy-based ranking system for selecting sentences for extractive summarization. The fuzzy knowledge base was automatically generated through a genetic algorithm. A corpus of newswire texts and their corresponding manual summaries was used to generate fuzzy classification rules. ROUGE informativeness measure was adopted as the fitness function of such an algorithm.*

1. Introduction

Automatic summarization involves producing a condensed version of a source text through selecting or generalizing its relevant content. As a result, either an extract or an abstract is produced. In this work only extracts are focused upon. These are built by copying and pasting relevant text segments from a source text into the final text, provided that the original order is preserved.

We describe SuPor-2 Fuzzy, an automatic extractive text summarizer that employs a fuzzy ranking model for selecting the relevant sentences. It is built on a previous version – SuPor-2 (Leite and Rino, 2006), which proposes a supervised summarizer for Brazilian Portuguese (BP). SuPor-2 is based on Kupiec et al. (1995) machine learning approach and employs 11 features combined through a Naïve-Bayes classifier. SuPor-2 Fuzzy aimed at overcoming SuPor-2 performance, as reported in (Leite and Rino, 2006), which presents a comparison involving other seven summarizers for BP.

The following modifications of SuPor-2 were carried out: we exchanged the SuPor-2 classifier for a fuzzy rule-based classifier, adapted some of its features and exchanged others for fuzzy sets. Building the rules for the fuzzy knowledge base was entirely automatic. For that, we used both a genetic algorithm and training data. Whenever possible, linguistic resources employed in SuPor-2 were kept intact in SuPor-2 Fuzzy, in order to preserve the main linguistic features for summarization.

The motivation of our work was to verify if a fuzzy classification model would fit well for Automatic Summarization (AS), a field where it is seldom used. There is a trend in most recent works (see, e.g., Witten and Frank, 2005) to employ Naïve-Bayes, Logistic Regression, Support Vector Machines, or other linear ranking methods for the task. This has been even evidenced at DUC[1] 2007 (e.g., Pingali et al, 2007; Li et al., 2007). However, there have been some criticisms towards the assumptions that underlie those methods. For instance, Naïve-Bayes assumes that features are statistically independent (Mani, 2001), which is barely the case

[1] Document Understanding Conferences. Publications can be found at http://duc.nist.gov/ [March/2009]

when text processing takes place. Moreover, we claim that probabilistic classifiers are more suitable for the task because an adequate sentence ranking is desired, instead of just predicting a class, as we did before (Leite and Rino, 2006). Considering such scenario, we aimed at investigating how well a fuzzy model could overcome similar problems and deal with the inherent uncertainty involved in pinpointing how likely a sentence should be to compose an extract. Our aim is, thus, to define an adequate fuzzy ranking model for extractive summarization, as suggested in related areas (see, e.g., Kang et al., 2006).

To the best of our knowledge there are few researches that address fuzzy modeling for AS. Kiani-B and Akbarzadeh-T (2006), for example, propose a similar approach to ours. However, ours differs from that because we employ (a) more features and (b) a specific evaluation tool for summarization – ROUGE – as the fitness function for the genetic algorithm, as we shall describe later; (c) we used labeled data for generating rules instead of only unlabeled data. For assessing our proposal, SuPor-2 Fuzzy performance was just compared with its crisp version (i.e., SuPor-2) using the same training and test data and the same features.

The rest of this paper is organized as follows: in Section 2 we describe SuPor-2 feature set; Section 3 describes the adjustments made on SuPor-2 feature set for fuzzy modeling; Section 4 describes the fuzzy knowledge base, along with the genetic algorithm used in its construction. In Section 5 the ranking model based on fuzzy classification is presented. Section 6 embeds SuPor-2 Fuzzy assessment, followed by final remarks in Section 7.

2. SuPor-2 Features

SuPor-2 takes after Kupiec et al.'s (1995) approach on using a Naïve-Bayes probabilistic classifier to determine how likely a sentence is to be relevant for an extract. It embeds eleven features (F1-F11 below) that address either surface, or linguistic factors that interact with each other in order to pinpoint relevance. Surface features are understood here, as those shallow constructions that underlie, or are present in, the text. Usually, they may be withdrawn by skimming through the text (e.g., word positioning, cue words or phrases, title and subtitles cues) or from its varied representations (e.g., a vector resulting from stemming and filtering stop words). They may then be used to compute sentence relevance through statistical models or other calculations that do not depend directly on processing linguistic knowledge. Oppositely, linguistic features are those that relate to linguistic clues underlying the text, may those refer to intra- or inter-sentential relations between words, or even to the document structure. Hence, differently from the former case, such features are computed through linguistic or discourse models defined beforehand. These, in turn, help producing underlying representations of the text, which are indeed the ones that are automatically handled to produce a relevance map of the sentences.

Some of the features depict full well-known AS methods and require language-dependent resources to preprocess the text (e.g., thesauri, taggers, stoplists, or lexicons). They are described as follows (pair-wised features differ only in the preprocessing method)[2]:

F1 and F2 - Lexical Chaining (Barzilay and Elhadad, 1999). Relevant sentences to include in an extract are pinpointed here through signaling the strength of lexical chains. Three sets of candidate sentences may be pinpointed using three different heuristics, namely: (H1) select every sentence *s* of the source text based on each member *m* of every strong lexical chain of the text. In this case, *s* is the sentence that contains the first occurrence of *m*; (H2) this is

[2] This description is given elsewhere (Leite and Rino, 2006).

similar to H1, but instead of considering all the members of a strong lexical chain, it uses only the representative ones. A representative member is one whose frequency is greater than the average frequency of all words in the chain; (H3) a sentence s is chosen by focusing only on representative lexical chains of every topic of the source text. Features F1 and F2 address H3 by identifying topics using either TextTiling (Hearst, 1993) or paragraphs as topic units, respectively.

F3 - Sentence Length (Kupiec et al., 1995). This just conveys the sum of words in a sentence.

F4 - Proper nouns (Kupiec et al., 1995). This conveys the sum of proper nouns in a sentence.

F5 - Sentence Location (Edmundson, 1969). This feature takes into account the position of both a sentence in the paragraph and a paragraph in the text.

F6 and F7 - Word Frequency (Luhn, 1958). These mirror the normalized sum of the words of a sentence. The words must first be preprocessed through stemming (F6) or generating 4-grams (F7).

F8 and F9 - Relationship Mapping (Salton et al., 1997). Using a graph as a representative structure of a text, this method aims at signaling the connectiveness between paragraphs and is similar to Lexical Chaining. Paragraphs are thus the vertices of the graph. Connectivity is expressed through a similarity measure: the higher it is the more connected paragraphs are. Three different ways of producing extracts are used to choose highly connected paragraphs in order to make an extract more cohesive. They refer to distinct ways of traversing the graph, as follows: (P1) a dense or bushy path treats paragraphs as totally independent from each other and focuses on top-ranked ones, i.e., the ones that are denser. Clearly, P1 does not tackle cohesion properly in considering paragraph independence. (P2) a deep path aims at overcoming that by choosing paragraphs that may be semantically inter-related. Only one topic, even an irrelevant one, may be conveyed by an extract. Thus, P2 may lack proper coverage of the source text. (P3) a segmented path proposes solving both former problems by addressing all the topics at once. F8 and F9 address stemming and 4-gramming of words, respectively, for pre-processing.

F10 and F11 - Importance of Topics (Larocca Neto et al., 2000). To identify relevant sentences, this method first identifies and classifies topics according to their importance in the text. Then it selects sentences for an extract that better correlate with most important topics. Thus, it signals how relevant a sentence is to a given topic The more important a topic is the more sentences will be considered. However, to determine the ones that will actually be included in an extract, their similarity to their respective topic centroids (Larocca Neto et al., 2000) is calculated. The words must first be preprocessed through stemming (F10) or generating 4-grams (F11). Values convey the harmonic mean between the sentence similarity to the centroid of the topic in which it appears and the importance of that topic.

All values of numeric features are normalized with respect to the text in the [0,1] interval, according to the corresponding metric. For example, F3 considers the number of words in the text to normalize sentence length. Table 1 summarizes the domain values of all the features in SuPor-2. Lexical chaining (F1 and F2) signals the heuristics that recommend a sentence. Sentence location (F5) uses pair-wised letters to signal respectively the positions of a sentence within a paragraph and of a paragraph in the text (Initial, Medium, or Final). Finally, Relationship Mapping (F8 and F9) addresses which paths are used to select a sentence.

Table 1. Domains of SuPor-2 Features

F#	Domains
F1	{'None', 'H1', 'H2', 'H3', 'H1H2', 'H1H3', 'H2H3', 'H1H2H3'}.
F2	
F3	[0, 1]
F4	[0,1]
F5	{'II','IM','IF','MI','MM','MF','FI','FM','FF'}
F6	[0, 1]
F7	
F8	{'None','P1','P2','P3', 'P1P2', 'P1P3', 'P2P3', 'P1P2P3'}
F9	
F10	[0, 1]
F11	

3. Adapting SuPor-2 Features for Fuzzy Modeling

Most systems that use fuzzy rules for reasoning deal only with numeric variables (see, for example, Eberhart and Shi, 2007), instead of using categorical values as SuPor-2 does, which are in essence crisp. The reason is that the latter may not be suitable to represent uncertainty. Aiming at investigating how SuPor-2 would react towards classifying features based on a fuzzy model, its feature set had to be reformulated. In doing so, not all the original features could be prompted. We also refined some of the remaining ones and included others that were considered to be relevant in the past (Leite et al., 2007). Table 2 presents all the features that were employed in SuPor-2 Fuzzy. All features are normalized in set [0,1] for non-biased measures. The ones in bold refer to novel ones; the ones in italic, to SuPor-2 features that have been unfolded into more features.

Table 2. SuPor-2 Fuzzy Features

F#	Feature
F1'	*Similarity of the sentence to its topic centroid (stemming preprocessing)*
F2'	*Similarity of the sentence to its topic centroid (4-grams preprocessing)*
F3	Sentence Length
F4	Presence of proper names in the sentence
F5'	**TextRank measure**
F6	Word Frequency (stemming)
F7	Word Frequency (4-grams)
F8'	*Position of the sentence in its paragraph*
F9'	*Position of the paragraph in the text*
F10'	*Importance of the topic (stemming preprocessing)*
F11'	*Importance of the topic (4-grams preprocessing)*

The only features that remained untouched were F3, F4, F6, and F7. The others were modified as follows: The categorical representation of the position of a sentence in the paragraph and the position of a paragraph in the text were replaced by two other numerical features (F8' and F9'). F8' signals the relative position of a sentence in its paragraph. F9' signals the relative position of the paragraph considering the whole text. Lexical Chaining was not employed in SuPor-2 Fuzzy because we find it difficult to map its heuristics onto numerical entities. This is due to using heuristics to select sentences, instead of ranking them. Features of the Importance of Topics method (F10 and F11) were unfolded in two distinct features, and the harmonic mean between the sentence similarity to the centroid of a topic and the importance of that topic was dropped of. Such modifications resulted in features F1’-F2’ and F10’-F11’, pair-wised sets signaling distinct preprocessing means.

Similarly to Lexical Chaining, Relationship Mapping features are in essence categorical (paths). They would be difficult to map onto numerical features if there were no possibility of representing them through graphs. These made it simple to configure F5’. Actually, we adopted TextRank (Mihalcea, 2005), which computes a numerical value that represents the importance of a sentence based upon sentence connectedness (sentences as nodes and similarity between sentences signaled by edges). First TextRank computes sentence similarity through content overlap, and then it uses a random walk model to build a path between nodes, aiming at grading the importance of a sentence in the graph.

4. Fuzzy Knowledge Base

Designing a system based on fuzzy rules depends upon two phases of utmost importance: (a) modeling fuzzy sets that represent relevant variables; (b) determining the fuzzy rules that will manage fuzzy inference. Both fuzzy sets and rules partially amount to the fuzzy knowledge base.

4.1. Modeling Fuzzy Sets

We adopted a standard modeling (e.g., Eberhart and Shi, 2007) regarding fuzzy sets. For each feature, 3 uniform triangles depict 3 fuzzy set functions to address a variable membership. Linguistic variables assume values in the set {low, medium, high}. Such values pinpoint the likelihood of an element (a sentence in our case) to be a member of a set. They are calculated through equations 1, 2 and 3 below.

$$low(x) = \begin{cases} 1 - \dfrac{x}{0.5}, x \leq 0.5 \\ 0, x > 0.5 \end{cases} \qquad [1]$$

$$medium(x) = 1 - \frac{|x - 0.5|}{0.5} \qquad [2]$$

$$high(x) = \begin{cases} \dfrac{x - 0.5}{0.5}, x \geq 0.5 \\ 0, otherwise \end{cases} \qquad [3]$$

4.2. Generating Fuzzy Rules

A common approach to design fuzzy rule systems is to use the expertise of humans along with a trial-and-error approach. This can be very subtle, especially when the number of features is large and the problem is complex. Genetic algorithms have been shown to be superior to traditional automatic approaches in finding nearly optimal solutions in this complex high dimensional search space (Eberhart and Shi, 2007). For this reason, we chose a genetic algorithm to determine fuzzy rules in SuPor-2 Fuzzy. It follows the Pittsburgh approach (e.g., Herrera et al., 1993; Cordon et al., 1996) in that it encodes a complete rule set in a chromosome. The Michigan approach, which is Pittsburgh counterpart — encodes just one rule per chromosome. So, Pittsburgh approach performs better when using slow fitness functions for determining fuzzy rules.

Populating the knowledge base was initially carried out by both random rules and rules generated by the Wang and Mendel's method (1992). This employs labeled data. Proportions of such rules are respectively 40 and 60% in the knowledge base.

4.3. Chromosome Encoding

Each chromosome comprises 100 rules represented by integer numbers that signal both feature values and the aimed encoded class. SuPor-2 considers 11 features. Class instances are represented as a feature itself (actually, the 12th one in the corresponding vector). So, the chromosome vector comprises 1200 elements. Feature values are defined as follows:

For a rule antecedent: 0 in a position p_i, i=1,…,11, signals that feature *i* does not appear in the antecedent; 1-3 values represent low, medium and high degrees respectively. **For a rule consequent (Class):** 0 indicates the false class, i.e., the class of sentences that would not be present in the extract; 1 indicates the true class instead.

4.3.1. Illustrating Chromosome Encoding

The vector below shows a partial segment that encodes the rule

$$F1=high\,,\ F2=medium\,,\ F3=low \Rightarrow class = True.$$

3	2	1	0	0	0	0	0	0	0	0	**1**	...
p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	p_{11}	p_{12}	

4.4. The Fitness Function

We adopted ROUGE-2 measure (Lin, 2004) as the fitness function. ROUGE recall rate, in this case, is calculated on bigrams. It mirrors the informativeness of an extract by correlating it with a reference, human made, summary. To evaluate an individual, namely, a sentence that may be candidate to include in a SuPor-2 extract, the encoded fuzzy rules are used along the fuzzy classifier as described in Section 5. After evaluating all the sentences of a source text, its extract is produced. The fitness function of the individual is the ROUGE-2 measure of the produced extracts.

4.5. Crossing and Mutation

In the evolutionary process crossing and mutation operators are employed. Simple crossing (one point) operator was used (e.g., Herrera et al., 1993; Eberhart and Shi, 2007), with random crossing point choice. Non-uniform mutation was employed (e.g., Herreira et al., 1993; Eberthart and Shi, 2007). The number of exchanged elements (genes) is randomly chosen up to a maximum of 10%. Each vector element is updated using Equation 4.

$$c_k' = \begin{cases} c_k + \Delta(0, LS(c_k) - c_k), d = 1 \\ c_k + \Delta(0, c_k), d = 0 \end{cases} \qquad [4]$$

Variables in [4] are defined as follows: *d* is a random digit between 0 and 1; c_k is the current value of gene k; c_k' is the new value for gene k; LS(c_k) = 3 if c_k is a feature gene, and if c_k is a class, then $LS(c_k)$ =1); $\Delta(a,b)$ returns random digits between *a* and *b*.

4.6. The Genetic Algorithm

Figure 1 presents a high-level code of the genetic algorithm. *P(t)* indicates the current population at generation *t*.

```
Begin
t=0;
Initialize P(t) with random and Wang-Mendel rules;
Evaluate P(t);
while (t < MAX_GENERATIONS) do
 Begin
  t=t+1;
  Elitist selection of P(t) from P(t-1);
  Apply Crossing operator in P(t);
  Apply Mutation operator in P(t);
  Produce text extracts using P(t) rules
  Evaluate P(t) using the ROUGE measure of the produced extracts;
 End
End
```

Figure 1. Genetic Algorithm

5. Fuzzy Model for Ranking Sentences

As explained in the previous section, for ranking sentences according to their relevance the generated fuzzy rules are used. As a result, a true or false class is assigned to each sentence. The importance (I) of a sentence (s) is determined through Equation 5:

$$I(s) = \sum_{i \mid \exists R_i \to True} \omega(R_i, s) - \sum_{i \mid \exists R_i \to False} \omega(R_i, s) \qquad [5]$$

This formula considers the contribution of each rule (*R*) of the fuzzy rule set for the importance of sentence *s*. Rules indicating that *s* should not appear in the extract (R_i → *False*) are negatively biased. Function ω pinpoints the strength of the rule activation under sentence *s* constraints. In other words, it measures the compatibility degree between the rule antecedent and the whole vector of sentence features. This is accomplished by calculating the t-norm minimum for all membership degrees of the sentence features to the corresponding *low*, *medium* or *high* fuzzy sets (Klir and Yuan, 1995).

5.1. Illustrating the Fuzzy Ranking Model

Here we illustrate sentence ranking based on fuzzy rules considering the three fuzzy sets (triangles) modeled by formulas [1], [2] and [3].

(a) the fuzzy rule R, defined as *F1=medium, F2=medium, F3=medium, F4=medium, F5=high, F6=medium, F7=medium, F8=medium, F9=medium, F10=high, F11=high* ⇒ *class = True;*

(b) sentence *s* vector, given by *s = [F1=0.5, F2=0.5, F3=0.5, F4=0.5, F5=0.6, F6=0.5, F7=0.5, F8=0.5, F9=0.5, F10=1, F11=1]*

Through Equations 2 and 3 the membership degrees of *s* features yield the following values: *medium*(0.5) = 1.0, *high*(0.6) = 0.2, and *high*(1.0) = 1.0. The t-norm minimum (ω) below signals the strength of the rule activation for sentence *s*, hence its importance for summarization, given by *I(s)* above.

$$\omega = min\,(1, 1, 1, 1, 0.2, 1, 1, 1, 1, 1, 1) = 0.2$$

6. Assessing SuPor-2 Fuzzy

We could not compare SuPor-2 and SuPor-2 Fuzzy feature sets directly because they are substantially distinct (see Tables 1 and 2). So, a modified version of SuPor-2 was considered instead, which used the same features of SuPor-2 Fuzzy and the same Bayesian classifier as that of SuPor-2 (see Section 2 for details). We also considered GistSumm (Pardo et al., 2003) as a baseline, an unsupervised extractive summarizer driven by the gist of the source text.

TeMário corpus (Pardo and Rino, 2003) was used for training both, the modified SuPor-2 and SuPor-2 Fuzzy systems. It comprises 100 news texts in BP (c.a. 613 words) along with their manual, reference summaries. These account for more than 2900 labeled instances used as input data for the Wang and Mendel (1992) (see Section 4.2 for details).

Training SuPor-2 Fuzzy consisted of evolving the genetic algorithm steps up to achieving the fuzzy classification rules (see Figure 1). The main parameters and configurations of such an algorithm are listed in Table 3. Usually, the fuzzy literature (e.g., Eberhart and Shi (2007) recommends a high crossing rate (85-95%) and a low mutation rate (0.5-1%), as we adopted. However, there are no precise guidelines for this, making it difficult for us to determine a better recommendation for our domain. The total evolution time for training amounted to 168.0 hours, or 7 days. Figure 2 shows how the fitness function (ROUGE-2) progressed during evolution with TeMário. Besides, in our case, probably only c.a. 20 generations should have been enough without significant recall decrease.

Table 3. Main configurations of the genetic algorithm

Crossing Rate	90%
Mutation Rate	1%
Population size	200
Maximum number of generations	75

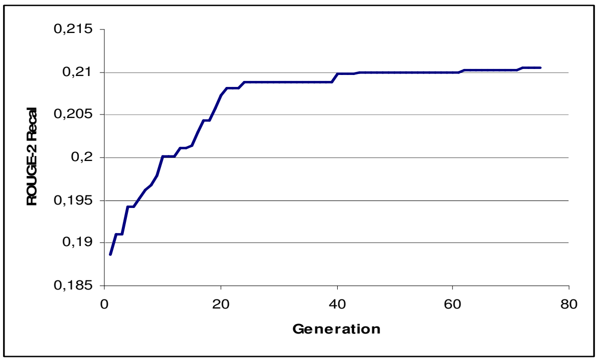

Figure 2. Genetic algorithm convergence for TeMário

Once the classification rules were obtained on the TeMário corpus, we run the three systems to produce extracts for a different corpus, the Summ-it one (Collovini et al., 2007). This also addresses news texts in BP and embeds manual summaries, but it is smaller than TeMário (50 texts, c.a. 390 words each).

We employed both ROUGE-1 and ROUGE-2 measures for evaluation, mirroring past DUCs (2003-2007). ROUGE was used without removing stems and stopwords because it does not provide those resources for BP. A 30% compression rate was adopted. Actually, it has been approximated to the length of the reference summaries for each extract, following previous approaches (Mihalcea, 2005). Table 4 shows the average recall rates for each summarizer. As it can be seen, SuPor-2 Fuzzy performed better than SuPor-2 modified and the other systems considering both ROUGE-1 and ROUGE-2 metrics. These results for the top two systems where analyzed with a t-student test of statistical significance: p-values of SuPor-2 modified and SuPor-2 Fuzzy are, respectively, 0.12014 for ROUGE-1 and 0.09466 for ROUGE-2. Considering ROUGE-2, the differences are statistically significant at a 90% confidence level. Both systems proved to differ from GistSumm at a 95% confidence level, and this result is also statistically significant.

Table 4. ROUGE recall metrics

System	ROUGE-1	ROUGE-2
SuPor-2 Fuzzy	0.74583	0.73859
SuPor-2 modified	0.73205	0.72323
GistSumm (Baseline)	0.57669	0.41821

Although the above results show that SuPor-2 Fuzzy outperforms the standard classifier, Bayesian-based, it should be noticed that its evolution phase aiming at generating rules takes much longer than the Bayesian method. Despite this, the effort may be worthwhile if we consider that the evolution phase is carried just once. Moreover, it may be promising under a large-scale summarization setting, a significant trend nowadays.

7. Final Remarks

Variations of the experiments described here shall be considered in the future. For instance, dynamic computing of ideal mutation and crossing rates in the evolution phase o the genetic algorithm may be employed (e.g., Eberhart and Shi, 2007). Since there are no guidelines for modeling fuzzy sets, other membership functions shall also be explored, aiming at devising anything but trial-and-error methods that may turn our systems scalable. As suggested in (Leite and Rino, 2006), automatic feature selection filter (Hall, 2000) can also be used along with any classifier for better accuracy.

References

Barzilay, R. and Elhadad, M. (1997). Using Lexical Chains for Text Summarization. In: Proc. of the Intelligent Scalable Text Summarization Workshop, Madri, Spain. Also In I. Mani and M.T. Maybury (eds.), Advances in Automatic Text Summarization, pp. 111-121. MIT Press, 1999.

Collovini, Sandra, Carbonel, T. I. , Fuchs, J. T. , Coelho, J. C. B. , Rino, L. H. M and Vieira, Renata. (2007). Summ-it: um corpus anotado com informações discursivas visando à sumarização automática. In: Proceedings do Congresso Nacional da SBC 2007.

Cordon O., Herrera F., Herrera-Viedma E. and Lozano F. (1996) Genetic Algorithms and Fuzzy Logic in Control Processes. Archives of Control Sciences.

Eberhart, Russell and Shi, Yuhiu. (2007). Computation Intelligence: Concepts to Implementations. Morgan Kaufman, New York.

Edmundson, H. P. (1969). New methods in automatic extracting. Journal of the ACM, 16(2):264-285.

Hall, M. A. (2000). Correlation-based Feature Selection for Discrete and Numeric Class Machine Learning. In: Langley, P. (ed.): Proc. of 17th International Conference on Machine Learning. Morgan Kaufmann, San Francisco, CA, 359-366.

Hearst, M. A. (1993). TextTiling: A Quantitative Approach to Discourse Segmentation. Technical Report 93/24. University of California, Berkeley.

Herrera, F., Lozano, M. and Verdegay, J.L. (1993). Genetic Algorithm Applications to Fuzzy Logic Based Systems. In: Proc. of the 9th Polish-Italian and 5th Polish-Finnish Symposium on Systems Analysis and Decision Support in Economics and Technology, pp. 125-134. Warsaw, 1993.

Herrera, F. and Lozano, M. (1999). Fuzzy genetic algorithms: issues and models. Dept. of Science and A. I., University of Granada, Technical Report No. 18071, Granada, Spain.

Kiani, A. and Akbarzadeh, M.R. (2006). Automatic Text Summarization Using Hybrid Fuzzy GA-GP Proceeding of the 11th IEEE International Conference on Fuzzy Systems, pp. 270-278, Vancouver Canada, 2006.

Kang, Bo-Yeong, Kim, Dae-Won and Li, Qing. (2006). Fuzzy Ranking Model Based on User Preference. IEICE Transactions 89-D(6): 1971-1974

Klir, George J. and Yuan, Bo (1995). Fuzzy sets and fuzzy logic: theory and applications. Upper Saddle River, NJ: Prentice Hall PTR.

Kupiec, Julian , Pedersen, Jan and Chen, Francine (1995). A trainable document summarizer. In: Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval, pp 68-73.

Larocca Neto, J., Santos, A. D., Kaestner, C. A. A. and Freitas, A. A. (2000). Generating Text Summaries through the Relative Importance of Topics. In: M. C. Monard and J. S. Sichman (Eds.), Proceedings of Iberamia-Sbia 2000, pp. 300-309. Springer-Verlag, Berlin, Heidelberg.

Leite, D. S. and Rino, L.H.M. (2006). Selecting a Feature Set to Summarize Texts in Brazilian Portuguese. In: J. S. Sichman et al. (eds.): Proc. of 18th. Brazilian Symposium on Artificial Intelligence (SBIA'06) and 10th. Ibero-American Artificial Intelligence Conference (IBERAMIA'06), pp. 462-471. Lecture Notes on Artificial Intelligence, No. 4140, Springer-Verlag.

Leite, D.S.; Rino, L.H.M.; Pardo. T.A.S. and Nunes, M.G.V. (2007). Extractive Automatic Summarization: Does more linguistic knowledge make a difference? In: Chris Biemann, Irina Matveeva, Rada Mihalcea, Dragomir Radev (eds.), Proc. of the Workshop on TextGraphs-2: Graph-Based Algorithms for Natural Language Processing, pp.17-24. 26 April, Rochester, NY, USA.

Luhn, H. P. (1958). The Automatic Creation of Literature Abstracts. IBM Journal of Research and Development, 2, 157-165.

Li, S., Ouyang, Y., Wang, W., and Sun, B. (2007). Multi-Document Summarization Using Support Vector Regression. In Proc. of Document Understanding Conference 2007. Rochester, New York USA, April.

Lin, C. Y. (2004). ROUGE: A package for automatic evaluation of summaries. In: Proc. of the Workshop on Text Summarization Branches Out (WAS-2004), pp. 74-81. Barcelona, Spain.

Mani, I. (2001). Automatic Summarization. John Benjamins Publishing Co., Amsterdam.

Mihalcea, R. (2005). Language Independent Extractive Summarization. In: Proc. of the 43th. Annual Meeting of the Association for Computational Linguistics, Companion Volume (ACL 2005), Ann Arbor, MI, June.

Pardo, T.A.S. , Rino, L.H.M. and Nunes, M.G.V. (2003). GistSumm: A Summarization Tool Based on a New Extractive Method. In N.J. Mamede, J. Baptista, I. Trancoso, M.G.V. Nunes (eds.), 6th Workshop on Computational Processing of the Portuguese Language - Written and Spoken – PROPOR (Lecture Notes in Artificial Intelligence 2721), pp. 210-218. Faro, Portugal. June 26-27.

Pardo, T.A.S. and Rino, L.H.M. (2003). TeMário: Um Corpus para a Sumarização Automática de Textos. NILC Tech. Report. NILC-TR-03-09. São Carlos, Outubro, 12p.

Pingali, P., K., R. and Varma, V. (2007). IIIT Hyderabad at DUC 2007. In Proc. of Document Understanding Conference 2007. Rochester, New York USA, April.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in Automatic Text Retrieval. Information Processing and Management 24, pp. 513-523.

Salton, G., Singhal, A., Mitra, M. and Buckley, C. (1997). Automatic Text Structuring and Summarization. Information Processing & Management, 33(2), pp. 193-207.

Wang, L. and Mendel, J. (1992). Generating fuzzy rules by learning from examples. IEEE Trans. on SMC, 22, pp. 414-1427.

Witten, Ian H. and Eibe, Frank (2005). Data Mining: Practical machine learning tools with Java implementations. Morgan Kaufmann, San Francisco