Generating extracts with genetic algorithms*

Enrique Alfonseca[1] and Pilar Rodríguez[1]

Computer Science Department, Universidad Autónoma de Madrid,
28049 Madrid, Spain
{Enrique.Alfonseca, Pilar.Rodriguez}@ii.uam.es
http://www.ii.uam.es/~ealfon

Abstract. This paper describes an application of genetic algorithms for text summarisation. We have built a sentence extraction algorithm that overcomes some of the drawbacks of traditional sentence extractors, and takes into consideration different features of the summaries. The fitness function can be easily modified in order to incorporate features such as user modelling and adaptation. The system has been evaluated with standard procedures, and the obtained results are very good.

1 Introduction

Automatic Text Summarisation is the task that consists in finding, in a textual source, which information is more relevant for a user or an application, and presenting it in a condensed way. It has received much attention lately, due to its many potential applications. In a broad sense, summarisation systems refer to different problems and can vary largely depending on their aim. For instance, a system that sends summaries of e-mails to mobile phones will be very different from a system that generates summaries of various newswire articles about the same topic. Each of these will have to process different language styles and different domains, and the requirements about the readability and size of the generated summaries will not be the same.

There have been several competitions for automatic text summarisation systems, that have encouraged research in the area, and established common evaluation procedures and metrics. The first one was the TIPSTER SUMMAC evaluation, and the two Document Understanding Conferences [DUC, Marcu, 2001b] in the years 2001 and 2002. The third DUC takes place in 2003.

In this paper, we propose a new procedure for generating summaries from single documents that addresses some of the drawbacks of existing approaches. The algorithm proposed is a sentence extraction procedure that makes use of genetic algorithms. This has several advantages: firstly, Marcu [2001a] indicates that, when scoring sentences for generating an extract, current search procedures are slow if the weight of a sentence depends on whether other sentences have been selected or not; our approach with genetic algorithms is capable of finding summaries with high scores in a very short time. Secondly, with this approach,

* This work has been sponsored by CICYT, project number TIC2001-0685-C02-01.

it is very easy to incorporate in the scheme factors such as the suitability of a summary to individual users, and to evaluate the relevance of a sentence in function of the remaining sentences that have been selected for the extract. The system has been evaluated with standard procedures, and the obtained results are very good.

1.1 Text extraction

Probably the most comprehensive review on Automatic Text Summarisation systems is the one provided by Mani [2001]. In it, text summarisation systems are classified in two different kinds: text abstraction and text extraction systems.

Text abstraction consists in summarising the original material, including at least some information that was not present in the original document. Text abstraction usually needs some kind of semantic processing of a text.

Text extraction systems, on the other hand, cite, literally, fragments from the original text. These fragments may be whole paragraphs, sentences, clauses, or words; it may consist in removing the closed class words (prepositions, determiners, conjunctions, etc.), or in extracting the sentences that are judged more relevant. As the compression level increases, more information is discarded from the original documents. Text extraction systems usually suffer from the following problems:

1. They extract single discourse units (e.g. sentences). This may lead to incoherence, for instance, when the extracted sentences contain conjunctions at the beginning of sentences, dangling anaphoras, etc.
2. Sentences should not be evaluated independently beforehand, and next extracted; quite on the contrary, the choice of a sentence should, for example, block the choice of a different sentence that describes the same idea. It may be the case that the top two sentences are each a paraphrase of the other.

In order to improve the readability of the extract, and to mitigate these problems, extracts are usually post-processed. The following are some typical problems:

1. Lack of conjunctions between sentences, or dangling conjunctions at the beginning of the sentences (e.g. an extracted sentence starting with *However*).
2. Lack of adverbial particles (e.g. it would be desirable the addition of *too* to the extract *"John likes Mary. Peter likes Mary"*)
3. Syntactically complex sentences.
4. Redundant repetition (e.g. repetition of proper names, which can be substituted by pronouns).
5. Lack of information (e.g. complex dangling anaphoras, such as *in such a situation*; or incomplete phrases).

1.2 Related work

Many extraction systems on use nowadays are based on the work done by Edmundson [1969], which described a procedure that has later received the name

of *Edmundsonian paradigm* [Mani, 2001]. Edmundson summarised papers about chemistry, by ranking the sentences with a function that took into account four variables: the position of the sentence in the document; the number of words from the title that appear in the sentence; the number of words in the sentence that are considered relevant or irrelevant for the domain of chemistry; and the number of words in the sentence that are very frequent in the document being analysed. The weighting function is a linear combination of the values of these variables. Lin and Hovy [1997] and Mani and Bloedorn [1998], amongst others, further extended this set of features.

This paradigm has often been criticised for the following reasons [Mani, 2001]:

- It does not take into account for the extraction process the compression rate. For instance, if the top two sentences, s_1 and s_2, provide together an idea; and the third sentence in the ranking, s_3, provides alone other important idea, then a 1-sentence summary should select s_3, because it includes a complete idea, than either s_1 or s_2 alone.
- The linear model might not be powerful enough for summarisation.
- Finally, it only uses morphological information.

Other approaches to extraction that try to capture discourse structure in the original texts try to prevent this problem. Mani [2001] distinguishes two ways in which discourse can be studied: **text cohesion** [Halliday and Hasan, 1996], studying relations between units in a text, and **text coherence**, represents the overall structure of a multi-sentence text in terms of macro-level relations between clauses or sentences [Marcu, 1999].

2 Summarisation with Genetic Algorithms

We have built a *sentence extraction* procedure for producing summaries from single documents. Sentence extractors usually make use of heuristics in order to weight the sentences, and then select the sentences with the higher scores. However, when the score of a sentence depends on whether other sentence has been selected for the summary, then it becomes more difficult to choose the best combination.

In our approach, the selection of the sentences that will appear in the extract is done using a new summarisation method that is based on genetic programming [Holland, 1975]. The procedure is the following: initially, the algorithm starts with a set of random summaries, each of which is considered an individual in a population. The *genotype* of a summary would be the sentences that it has selected to create an extract. For instance, let us suppose that a document contains 47 sentences, and we need a summary with only 13 sentences. Figure 1 shows a possible initial set of five random summaries.

A fitness function has been defined based upon some heuristics. Some of them were previously known to indicate that particular sentences are relevant, and we have added some others that help produce summaries adapted to the

0	2	6	7	9	16	22	24	26	30	38	43	44
0	5	6	12	18	19	21	26	28	31	38	43	46
1	5	6	9	18	24	31	33	35	36	42	44	45
2	3	4	7	20	24	27	29	31	34	38	41	43
2	5	6	8	17	20	25	27	29	32	34	43	44

Fig. 1. Initial population of summaries. Each line is the genotype of a summary, and contains the numbers of the sentences that will be selected for that summary.

user's profiles. The objective is that the most informative summaries receive the highest fitness value.

The following are some characteristics of summaries that had been already observed when designing summarisation algorithms:

- Summaries that contain long sentences are better summaries than summaries that contain short sentences [Marcu and Gerber, 2001]. A partial fitness function can be defined as the sum of the lengths of all the sentences in the extract (measured in number of words): $L(\mathcal{S}) = \sum_{i=0}^{N} length(s_i)$.
- Summaries that contain sentences that occur in the beginning of a paragraph in the original documents are better than summaries that contain sentences that occur toward the end [Hovy and Lin, 1999, Mani, 2001]: $W(\mathcal{S}) = \sum_{i=0}^{N} 1/position(s_i)$.
- Summaries that contain the sentences in the same order than in the original documents are better than otherwise [Marcu, 2001a]. This function can be directly implemented in the genetic algorithm, by forcing the sentences to be always ordered: $O(\mathcal{S}) = 1$ if the sentences are ordered, 0 otherwise.
- Summaries that contain sentences from all the paragraphs are better than summaries that focus only on a few paragraphs [Marcu, 2001a]: $C(\mathcal{S}) = |\{p : paragraph(p) \wedge (\exists s \epsilon \mathcal{S} : s \epsilon p)\}|$.

The following heuristics were also used for the fitness function. Some of them are there for adapting the summary to a possible user profile, and last two try to rule out uninformative sentences.

- $P(\mathcal{S})$: summaries that contain sentences that are more relevant according to the user profile are better than summaries that don't. In our approach, the user profile is defined as a set of documents that the user considers interesting, and a similarity between the sentences and those documents is calculated using the vector model (e.g. the tf·idf metric).
- $Q(\mathcal{S})$: in case that the user has specified a query, then the summaries that contain sentences with any of the user's query keywords are better than summaries that only contain general-purpose terms.
- $V(\mathcal{S})$: summaries that contain complete sentences (with subject and verb) are better than summaries that contain any sentence with any of those constituents.
- $I(\mathcal{S})$: questions are usually low-informative sentences.

Generation	Sentences	Fitness
0	1 6 8 13 20 28 29 33 35 41 42 44 46	46.501553
5	1 6 13 20 22 28 29 33 35 41 42 44 46	47.385387
10	1 3 4 6 13 22 29 33 35 41 42 44 46	49.599186
20	1 3 4 6 13 26 29 33 35 39 41 42 44	51.74695
50	3 4 19 24 25 26 29 39 40 41 42 43 44	54.43973

Table 1. Summary with the best score at different generations.

The final score of a summary is a weighted sum of the different fitting functions. We chose a linear combination for our experiments because it has already been used in all the previous approaches that follow the Edmundsonian paradigm, but other approaches should also be investigated. As in his case, the weights for each of the partial fitness functions were set by hand, with the feedback from several experiments.

Once the fitness function has been decided, we applied a standard genetic algorithms, as described here: we start with an initial population of summaries; at every generation, the two less adapted individuals in the population die, and the two most adapted have children. Population changes by means of the *mutation* operator, that changes randomly a sentence number in an individual, and the *crossover* operator, that interchanges a random portion of the genotype of the two parents. After a certain number of iterations, the population becomes homogeneous and the best score does not vary for a certain number of steps. At that point, the evolution stops and the summary with the best fitness function is generated. Table 1 shows the best summary at different stages of evolution for a summary of 13 sentences from a total of 47.

3 Evaluation

The summarisation system has been tested on an adaptive on-line information system about Darwin's *Voyages of the Beagle*. The information shown to the users is dependent on their interest profiles, and they may indicate a compression rate for the texts in the adaptive site. We built a test set of thirty documents from the site, with lengths ranging from 6 to 59 sentences. It consisted of three subsets:

1. A set with 10 documents for a user whose general interest is biology, needing a compression of roughly 29% (the summaries had to be 29% of the original text). Two of the summaries had an additional constraint: that the user wanted specifically to have information about a *carrancha* and about a *bizcacha*, respectively.
2. A set with 10 documents for a user interested in geography, with a compression rate of roughly 39%. Again, some summaries have additional keywords, such as *Siberia*.
3. A set with 10 documents for a user interested in history, with a compression rate of roughly 46%. As before, some summaries will be general, and others have to be focused on specific topics.

Compression rate	Judges pair	Agreement
0.29 (10 docs.)	1,2	63.11%
	1,3	62.14%
	2,3	62.14%
0.39 (10 docs.)	1,2	61.26%
	1,3	63.39%
	2,3	62.16%
0.46 (10 docs.)	1,2	65.57%
	1,3	72.95%
	2,3	72.13%

Table 2. Agreement between pairs of judges.

In order to create human-created summaries for testing purposes, each of the three sets of documents was given to three different human judges. Every document carried an explanatory note indicating the preferences of the user, and the number of sentences that had to be selected. The agreement of every pair of judges was calculated for every set of ten documents, as shown in Table 2.

The creation of a summary is something that is not totally objective: different people will probably produce different summaries from the same text, even though they do not have to reformulate the information but only extract sentences. Therefore, for a proper evaluation of the system it is equally important to know how much humans agree on the same task. A widely used metric to measure judge agreement is the Kappa statistic [Carletta, 1996], defined as

$$K = \frac{P(A) - P(E)}{1 - P(E)} \tag{1}$$

where P(A) is the proportion of times that the judges agree, and P(E) is the proportion of times that we would expect them to agree by chance. If all the judges agree, $P(A)$ is 1 and the value of Kappa is 1; on the other hand, if the agreement of the judges is the one that could be expected by mere chance, Kappa takes the value of 0. According to Carletta [1996], citing other research works, a value of $K > .8$ can be considered good, and a value between 0.67 and .8 "allows tentative conclusions to be drawn". However, in medical literature values of K between 0.21 and 0.4 are considered "fair".

The values of Kappa obtained from the judges' answers are listed in Table 3. $P(E)$ was calculated in the following way: with a compression rate of 0.46, given a sentence, the probability that three judges randomly choose it as summary-worthy is $0.46^3 = 0.097$. The values obtained show that there is some agreement amongst the judges, as the level of agreement was always substantially higher than the one that could be expected with random summaries, although they are below 0.67. In fact, as [Mani, 2001, pg. 226] notes, in the SUMMAC evaluation four subjects were asked to evaluate some summaries, without explicit criteria. There was unanimous agreement only for 36% of the sentences, leading to a Kappa of 0.24.

Compression rate	P(A)	P(E)	Kappa
0.29 (10 docs.)	0.46	0.024	0.447
0.39 (10 docs.)	0.43	0.059	0.435
0.46 (10 docs.)	0.59	0.097	0.546

Table 3. Level of agreement between judges.

The conclusions we can derive from the fact that inter-judge agreement is not very high is that the task was not defined in a very precise way. Indeed, many choices of sentences were left to the personal choice of the judges, as there were not very specific guidelines. In the case of summarisation, even if we know that the user is interested on a topic, such as *biology*, there might be many sentences referring to that topic, and different judges use their own criteria. In any case, the value of Kappa was always well above that of the SUMMAC competition.

After collecting the information from the judges, the *target summaries* used to evaluate the algorithm were calculated with the sentences that had received more votes from them. The agreement between the genetic algorithm and the hand-made summaries was 49.51% for the 29% summaries, 54.95% for the 46% summaries, and 67.21% for the 46%. Considering that the agreement between human judges was between 60% and 70% (c.f. Table 2), the machine-generated summaries can be considered to be at worst around 15% less accurate than human-made extracts, and at best around 5% less accurate.

The readability of the summaries was evaluated in a controlled experiment with 12 people. They were asked to read the generated summaries, and to mark the readability and coherence of the summaries from 1 to 5 (1 meaning very low, and 5 very high). The mean of the answers was 3.42, with a standard deviation of 2.63 (which means that there was substantial disagreement between them).

4 Conclusions and future work

This work includes a new approach for text summarisation based on generating extracts of texts with genetic algorithms. The method is very easy to program, and the performance is good considering its complexity, as it takes in average 250 milliseconds to summarise a 20-sentence text in a Pentium III 900MHz machine. In contrast, Marcu [2001a] states that an algorithm with similar weight functions takes a couple of hours of computation for each document collection in the Document Understanding Conference.

Most procedures for generating a summary by using extraction have one important drawback in that they do not take into account the compression rate for choosing the sentences. The weight of the extract should take into consideration all the sentences that have been selected and the relationships between them. This is specially relevant for multi-document summarisation, when there are sentences with shared meanings. Also, the algorithm does not even restrict the length of the summary. We can easily define the genotype as a boolean vector

with zeros and ones, and define that the sentences extracted are those such that the corresponding gene has a value of 1. In this case, we can *evolve* a population of summaries whose length is not fixed beforehand.

The proposed fitness function is only experimental, and most of the future research we are planning concerns it. Apart from introducing new heuristics, we are aware that a linear combination of the different values might not be the best solution, so other possibilities should be explored. The approach presented here is unsupervised, but a future objective is to make the fitness function evolve as well as the summaries, in a supervised environment. The summarisation procedure should also be extended with some kind of linguistic post-processing, such as resolving the antecedents of the pronouns, so that a pronoun whose antecedent has been removed can be replaced by it.

Measures of sentence cohesion and meaning overlapping can be easily encoded in the fitness function, which will guide the evolution of the population of summaries. Genetic algorithms also have the advantage that the search performed is nonlinear, but they can be programmed so the performance is not slow, and they can be built for practical applications. It will be computationally tractable because that function is not called exhaustively for every possible summary, but only on those that belong to the population of summaries at each iteration.

We would finally like to thank Alejandro Sierra and Pablo Castells for the fruitful discussions about future work.

References

J. Carletta. Assessing agreement on classification tasks: the kappa statstic. *Computational Linguistics*, 22(2):249–254, 1996.

H. P. Edmundson. New methods in automatic abstracting. *Journal of the Association for Computational Machinery*, 16(2):264–286, 1969.

M. A. K. Halliday and R. Hasan. *Cohesion in Text.* Longmans, London, 1996.

J. Holland. *Adaptation in natural and artificial systems.* University of Michigan, 1975.

E. Hovy and C-Y. Lin. *Automated Text Summarization in SUMMARIST.* I. Mani and M. Maybury (eds.) Advances in Automatic Text Summarization. MIT Press, 1999.

C-Y. Lin and E. Hovy. Identifying topics by position. In *Proceedings of the 5th Applied Natural Language Processing Conference*, New Brunswick, New Jersey, 1997.

I. Mani. *Automatic Summarization.* John Benjamins Publishing Company, 2001.

I. Mani and E. Bloedorn. Machine learning of generic and user-focused summarization. In *Proceedings of AAAI'98*, 1998.

D. Marcu. *Discourse Trees are good indicators of importance in text. In I. Mani and M. T. Maybury (eds.), Advances in Automatic Text Summarisation.* MIT Press, 1999.

D. Marcu. Discourse-based sumarization in duc-2001. In *Proceedings of Document Undestanding Conference, DUC-2001*, 2001a.

D. Marcu. The document understanding conference: A new forum for summarization research and evaluation. In *Aut. Summarization Workshop, NAACL-2001*, 2001b.

D. Marcu and L. Gerber. An inquiry into the nature of multidocument abstract. In *Proceedings of the NAACL'01 workshop on text summarisation*, Pittburgh, PA, 2001.