Description of the UAM system for generating very short summaries at DUC-2003*

Enrique Alfonseca and Pilar Rodríguez
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 the techniques used for producing very short summaries (around ten words) of single documents. The processing has been divided in two separate steps: firstly, a sentence extractor selects the most relevant sentences from the document; secondly, portions of those sentences are selected in order to produce the final headline. The results obtained in the DUC-2003 evaluations show that this procedure has an average usefulness, as it was ranked the seventh out of thirteen systems (just in the middle).

1 Introduction

The UAM system has only participated in the first task in DUC 2003: the generation of very short summaries (around ten words). Our procedure uses extraction techniques: selecting and putting together several portions of the original documents for building the summary. In our approach, there are two separate extraction steps:

- The identification of the most relevant sentences.
- The extraction of relevant words and phrases from those sentences, while keeping the coherence of the output as much as possible.

This paper is structured as follows: Section 2 describes the procedures followed for extracting a few sentences from the document; Section 3 describes the procedures for actually producing the very short summaries; and Section 4 briefly summarises the approach followed.

2 Sentence extraction

In a previous paper [Alfonseca and Rodríguez, 2003] we described a procedure for speeding up the generation of extracts from single documents using a combination of heuristics to weight sentences and summaries. It is strongly influenced by Edmundson [1969]'s work, in which several papers about chemistry are summarised by ranking the sentences with a function that takes 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. Recently, Lin and Hovy [1997] and Mani and Bloedorn [1998], amongst others, further extend this set of features.

Below are listed a few features that have been already considered useful for creating a summary, and a way in which they can be represented with numerical values:

*

- Summaries that contain long sentences are better summaries than summaries that contain short sentences [Marcu and Gerber, 2001]. A numerical 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 summaries [Hovy and Lin, 1999, Mani, 2001]. Similarly, the position of the paragraph can also be encoded as a feature.
  $W(\mathcal{S}) = \sum_{i=0}^{N} 1/(\alpha + position\_in\_paragraph(s_i))$
  $T(\mathcal{S}) = \sum_{i=0}^{N} 1/(\beta + paragraph\_position\_in\_document(s_i))$

- Summaries that contain the sentences in the same order than in the original documents are better than otherwise [Marcu, 2001]. This function can be directly implemented by forcing the sentences to be always ordered.

- Summaries that contain sentences from all the paragraphs are better than summaries that focus only on a few paragraphs [Marcu, 2001]:
  $C(\mathcal{S}) = |\{p : paragraph(p) \wedge (\exists s \epsilon \mathcal{S} : s \epsilon p)\}|$.

- Summaries that only contain complete sentences (with subject and verb) are better than summaries that contain any incomplete sentence:
  $V(\mathcal{S}) = |\{s : s \epsilon \mathcal{S} \wedge has\_subject(s) \wedge has\_verb(s)\}|$

- Questions are usually low-informative sentences:
  $I(\mathcal{S}) = -|\{s : s \epsilon \mathcal{S} \wedge is\_a\_question(s)\}|$

- Sentences with a high degree of word overlapping probably convey the same information, so the summary is redundant. The word overlapping can be calculated, for instance, using the cosine similarity [Otterbacher et al., 2002]:
  $O(\mathcal{S}) = -|\{< s, t >: s, t \epsilon \mathcal{S} \wedge word\_overlapping(t, s) > \theta\}|$

Following the Edmundsonian paradigm, we defined the final score of a summary as a weighted sum of the different heuristics (with the weights initially set by hand with trial-and-error). However, this paradigm has often been criticised for the following three reasons [Mani, 2001]:

1. The linear model might not be powerful enough for summarisation. Other approaches have trained with different paradigms, such as Support Vector Machines, to discriminate more accurately between relevant and irrelevant sentences [Karamuftuoglu, 2002]. We should investigate more expressive representations.

2. It only uses morphological information; however, if found useful, the paradigm can be easily extended with features obtained by means of a syntactic analysis or sentence cohesion metrics.

3. 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. This problem can be overcome by adding features that weight whole summaries, rather than separate sentences; however, the addition of these features makes much more difficult the search for the best summary, as the possible number of extracts grows exponentially with the number of sentences.

As Marcu [2001] explains, when scoring summaries 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 proposal consists in using genetic algorithms for finding the summaries with high scores in a very short time.

0	0	0	0	1	1	0	1	0	1	0	1	1
0	0	1	0	0	1	0	0	0	1	0	1	1
1	0	0	0	0	1	0	1	0	0	0	0	1
0	0	1	0	0	0	1	0	0	1	0	1	0
0	0	1	1	0	0	1	0	0	1	0	0	1

Figure 1: Initial population of summaries. Each line is the genotype of the summary of a document with 13 sentences. For instance, the first summary consists of the sentences at the positions 5, 6, 8, 10, 12 and 13.

2.1 Summarisation with Genetic Algorithms

In our approach, the selection of the sentences that will appear in the extract is done using a simple genetic algorithm [Holland, 1975]. Let us suppose that the original document has N sentences: then, each summary is represented as a vector of N bits, where a 0 in the i^{th} position means that the i^{th} sentence will not been extracted, and a 1 means otherwise. Initially, the algorithm starts with a random set of vectors of bits, each of which is considered an individual in a population. For instance, let us suppose that a document contains 13 sentences. Figure 1 shows a possible initial set of five random summaries for that document.

The fitness function has been defined as a linear combination of the heuristics described above, together with an additional term that tries to produce summaries of the intended length. The objective is that the most informative summaries receive the highest fitness value:

$$\begin{aligned}F(\mathcal{S}) = & w_L \cdot L(\mathcal{S}) + w_W \cdot W(\mathcal{S}) + w_T \cdot T(\mathcal{S}) + w_C \cdot C(\mathcal{S}) + \\ & w_V \cdot V(\mathcal{S}) + w_I \cdot I(\mathcal{S}) + w_O \cdot O(\mathcal{S}) - \\ & (summary\_length - target\_length)^2\end{aligned} \quad (1)$$

The implementation of the genetic algorithms that has been used is the `PGAPack` library [Levine, 1996]. The process, described in a few words, is the following: we start with an initial population of summaries; at every generation, the less adapted individuals in the population die, and the most adapted replicate. The population varies randomly by means of the *mutation* operator, that changes randomly a bit in an individual, and the *crossover* operator, that interchanges random bits 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 value is generated.

The method is very easy to program, and the performance is good considering its complexity. Using a Pentium III 900MHz with Linux RedHat 7.2, it takes in average 20 seconds to summarise a document collection from DUC-2003. In contrast, Marcu [2001] states that an algorithm with a similar weight function but a different search procedure takes a couple of hours of computation for each document collection in DUC-2001.

2.2 Training the model weights

As stated before, the weights for the fitness function and the rest of the parameters have been tuned manually after several experiments. We decided to run an additional genetic algorithm on top of the previous one in order to discover automatically the weights that produce the best results. The procedure chosen was the following:

1. Initialise the population as a vector of weights (8-bit fixed point numbers with 4 bits for the integer part).

2. The fitness function for each vector of weights is the F-measure of the extract generator using those weights.

3. Execute a genetic algorithm in order to find the best weights.

w_L	w_W	α	w_T	β	w_C	w_V	w_I	w_O
0.0125	0.00625	8.25	0.01875	2	0	0.5	15.9375	0.5

Table 1: Weights obtained automatically with a genetic algorithm.

The weights obtained are listed in Table 1. As can be seen, the position of the paragraph was judged more relevant than the position of the sentence inside the paragraph; sentence length received a large weight considering that, when multiplied by the (typically high) number of words in the sentences, it becomes an important factor. Furthermore, the appearance of a question in a summary had a very negative impact on the fitness function.

3 Generating the very short summaries

The sentence extraction algorithm was executed on the original DUC documents for extracting three or four sentences from each. Next, the sentences were processed with the following tools:

- The TnT part-of-speech tagger [Brants, 2000].
- A stemmer written in flex based on the LaSIE stemmer [Gaizauskas et al., 1995].
- Three chunkers written in C++ and Java; the first one detects Complex Quantifiers; the second one detects base Noun Phrases, and the third one detects complex verbs [Manandhar and Alfonseca, 2000].
- A subject-verb and verb-object detector, written in Java *ad hoc* by the author with hand-crafted rules.

We considered that events are usually represented with verbs; therefore, we chose all the verb phrases from the extracted sentences together with their arguments. Next, the verb phrases were ranked using sentence cohesion as a weight. The procedure is the following:

1. Take each verb phrase and its arguments (subject and direct object). If the parser has not been able to identify either the subject or the direct object of a verb phrase, it will be ignored.
2. Remove all adjectives and adverbs (except when a noun phrase only contains adjectives and would be left empty).
3. Calculate the number of lexical links (after stemming) with all the other sentences in the original document (before producing the extract); normalise according to the word that produces the maximum number of lexical links.
4. Calculate the number of lexical links (after stemming) with all the other sentences in the same collection of documents; normalise according to the word that produces the maximum number of lexical links.
5. Rank the verb phrases according to the sum of those values.

Figure 2 shows a 4-sentence sample extract and the extracted verbs from it, and Table 2 shows the phrases that were extracted.

Finally, the phrases with the highest weights were chosen, ordered in the order in which they appeared in the extract and, when possible, connected by means of a conjunction (if there was any between both in the extract). We also added the additional heuristic of ignoring all the verb phrases whose verb is a hyponym of *communicate*

Schizophrenia patients whose medication could not stop the imaginary voices in their heads gained some relief after researchers repeatedly sent a magnetic field into a small area of their brains.

About half of 12 patients studied said their hallucinations became much less severe after the treatment , which feels like having a woodpecker knock on your head " once a second for up to 16 minutes , said researcher Dr. Ralph Hoffman.

The voices stopped completely in three of these patients.

The effect lasted for up to a few days for most participants , and one man reported that it lasted seven weeks after being treated daily for four days.

Figure 2: Example extract from the first document in collection d100a.

Weight	Phrase
11.736842	Schizophrenia patients gained some relief
9.431579	said researcher Dr. Ralph Hoffman
6.7157893	12 patients studied
3.1263158	The voices stopped
3.0842106	sent a magnetic field
1.2736843	The effect lasted
0.12631579	one man reported
0.08421053	said their hallucinations
0.06315789	lasted seven weeks

Table 2: Phrases extracted from the extract in Figure 2.

[treatment] Schizophrenia patients gained some relief, after sent a magnetic field; 12 patients studied; The voices stopped.

[schizophrenia] the disabling disorder is caused by gene abnormalities; their evidence linking that region.

[pathway] The study may help scientists; develop drugs; to treat schizophrenia, and stop smoking.

[brain] Exploring chaos, where perception is regulated; a region known a sort; it 's something.

[study] Treatments are woefully underused, of reviewing the literature and interviewing patients; approaches percent.

[rat] Cesarean babies may be susceptible to schizophrenia; Patricia Boksa and Bassem El-Khodor delivered their young.

influences and season are significant risk factors; The study published 's issue; to develop schizophrenia.

[study] lend support; doctors will eventually be able; to offer preventive treatment to people; took advantage.
[kety] A Massachusetts scientist has won an Albert Lasker Award.

[study] such early intervention can keep the illness; to begin his prevention trial, with patients suffering.

[schizophrenia] People display; to reveal a structural abnormality in more primitive brain areas.

palms and fingerprints may be used to diagnose schizophrenia and mental disorders; develop any abnormality.

Figure 3: Headlines produced for document collection d100a.

4 Summary

We have described an approach for automatic generation of very short summaries (around 10 words) from articles. It starts by selecting the most relevant sentences, using a genetic algorithm and a combination of well-known heuristics as the fitness function. The weights for each heuristic were obtained with another genetic algorithm built on top of them. Secondly, the verb phrases and their arguments were extracted from those sentences and ranked in terms of number of cohesive links to other words in the document and in the collection. The verb phrases were selected until the total number of words exceeded 10, and next they were connected with conjunctions whenever possible.

Measured in terms of usefulness, our system ranked in the 7th position, just in the middle of the 13 systems that have participated in the first task of DUC-2003.

References

E. Alfonseca and P. Rodríguez. Generating extracts with genetic algorithms, 2003. to appear in European Conference on Information Retrieval (ECIR-2003).

T. Brants. *TnT - A Statistical Part-of-Speech Tagger.* User manual, 2000.

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

R. Gaizauskas, T. Wakao, K. Humphreys, H. Cunningham, and Y. Wilks. University of sheffield: Description of the lasie system as used for muc-6. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, pages 207–220. Morgan Kauffmann, 1995.

J. Holland. *Adaptation In Natural and Artificial Systems.* University of Michigan Press, 1975.

E. Hovy and C-Y. Lin. *Automated Text Summarization in SUMMARIST*, pages 81–94. I. Mani and M. T. Maybury (eds.) Advances in Automatic Text Summarization. MIT Press, Cambridge, Massachusetts, 1999.

M. Karamuftuoglu. An approach to summarisation based on lexical bonds. In *Document Understanding Conference, DUC-2002*, 2002.

D. Levine. User guide to the pgapack parallel genetic algorithm library, 1996.

C-Y. Lin and E. Hovy. Identifying topics by position. In *Proceedings of the 5th Applied Natural Language Processing Conference*, pages 283–290, New Brunswick, New Jersey, 1997.

S. Manandhar and E. Alfonseca. Noun phrase chunking with apl2. In *Proceedings of the APL-Berlin-2000 Conference, Berlin. Also published as E. Alfonseca and S. Manandhar, Noun Phrase chunking with APL2, APL Quote Quad (ACM SIGAPL), Vol. 30:4, p. 135-143*, 2000.

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 the Fifteenth National Conference of Artificial Intelligence (AAAI'98)*, pages 821–826, Menlo Park, California, 1998.

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

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.

J. Otterbacher, A. J. Winkel, and D. R. Radev. The michigan single and multi-document summarizer for duc-2002. In *Document Understanding Conference, DUC-2002*, 2002.