Extract-based Summarization with Simplification

Partha Lal and Stefan Rüger
Department of Computing, Imperial College
180 Queen's Gate, London SW7 2BZ, England
s.rueger@ic.ac.uk

Abstract

We describe a single-document text summarizer using the Text Engineering framework GATE. The summarizer extracts sentences using a combination of simple Bayes classifiers, resolves anaphora using GATE's ANNIE module, simplifies words using the MRC psycho-linguistic database and WordNet, and supplies background information to named persons and places using internet resources.

1 Introduction

Our system performs two main functions. It first works as an extract-based single document summarizer. It then goes some way towards to customising the document for an audience with limited background knowledge or reading ability — a potential audience includes school children. Only the summarization part was used in the DUC evaluation.

The summarizer works as a Bayesian pattern classifier over sentences, similar to (Kupiec, Pedersen and Chen 1995) and (Sekine and Nobata 2001) trained from an annotated corpus. Features used to assign a score to were

- word count in a sentence
- XML element enclosing the sentence
- position of the enclosing paragraph within the document
- position of the sentence within the enclosing paragraph
- mean tf.idf of named-entities (NEs)
- level of co-reference with NEs in headline elements
- inclusion of highly co-refered NEs

Dangling anaphor repair is performed for some pronominal anaphora. This repair means the system deviates from being strictly extract-based, and so was evaluated in DUC 2002 as an abstract-based system.

The customisation of the summary addresses the importance of context factors described by Spärck-Jones (1998). The purpose factor of audience is considered to be school children intending to read, say, a newspaper article. The system applies lexical simplification — replacing difficult words with simpler ones — and background knowledge addition. The lexical simplification is inspired by PSET (Carroll, Minnen, Canning, Devlin and Tait 1998) and uses tools that were written for that project. Background knowledge includes information on people and maps of places, and is taken from sources on the web.

The remainder of the paper is structured as follows. Section 2 details the design of our system, Section 3 describes the training of the classifier on a training corpus. The results of the evaluation are in Section 4 and the work is then evaluated in Section 5. A demonstration of the system can be found at `http://km.doc.ic.ac.uk/pr-p.lal-2002/`. For GATE users, the system can be downloaded as a CREOLE Repositiory at the same URL — a screen shot is shown in Figure 7.

2 Design

The system is built within the GATE[1] framework (NLP group, University of Sheffield 2002). GATE is a modular architecture which provides a way in which text processing components can be built, combined and reused. GATE comes with ANNIE[2] — a series of modules that together provide named entity extraction and coreferencing. Pronominal anaphor resolution can also be performed by ANNIE (Dimitrov 2002).

Our system consists of ANNIE plus a series of modules written to do the actual summarization — see Figure 1.

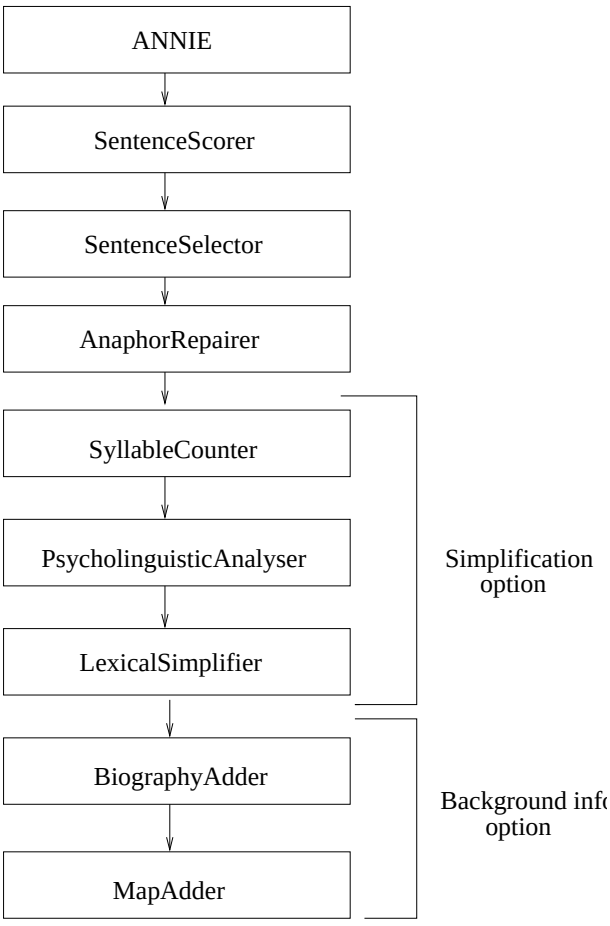

Figure 1: The overall structure of summarizer

[1]General Architecture for Text Engineering
[2]A Nearly New Information Extraction System

2.1 Summarization

The extract selection is performed by `SentenceScorer` and `SentenceSelector` — the interesting work is done in the former.

In `SentenceScorer` each sentence is assigned with a score shown in Eq (1).

$$\text{score}(s) = \sum_f w_f S_f(s), \qquad (1)$$

where f is a feature of the sentence s, each of the S_f is a conditional probability,

$$S_f(s) = P(s \text{ should be extracted} | f), \qquad (2)$$

and the w_f are coefficients for a convex combination, ie $w_f \geq 0$ and $\sum w_f = 1$.

Both the coefficients in Eq (1) and the probability model (2) used for most features are based on training data. The conditional probabilities are represented within XML files, and the coefficients are initialisation parameters of the module — this means the results of new training can be easily used.

The features used are listed above in Section 1, some perhaps need explanation. The "XML element enclosing the sentence" takes advantage of the TREC markup on the text, so headlines and lead paragraphs are indicated.

The document frequency data behind the tf.idf feature was gathered from the TREC corpus[3]. Separate df values were derived for each publication type in the corpus, since a term with a high document frequency in the Financial Times — a company perhaps — might not have a high document frequency in Associated Press articles.

The "level of co-reference with headline element" in a sentence is the number of NEs in it that refer to NEs that occurred within the headline. The XML markup for headlines is used for TREC data and the `HEAD` element is used when applied to HTML documents.

NEs in GATE have a feature called a matches list, listing which other NEs or pronouns co-refer to it. "Inclusion of highly co-refered NEs" is simply the length of that list, summed over each of the NEs in the sentence.

[3]Text REtrieval Conference, see http://trec.nist.gov

`SentenceSelector` then just annotates the highest scoring sentences as being extracts. The number of sentences used is dictated by the user through either a word count or a %-compression.

Once the sentences to be extracted are identified, anaphor repair is performed. The `AnaphorRepairer` module relies on ANNIE's anaphor resolution. The system was at first restricted to he/she/him/her/himself/herself anaphora since ANNIE performed well on them but not on others. Subsequently, improvements were made to justify repairing I/me/my/myself anaphora too.

2.2 Simplification

2.2.1 Lexical

Syllable counts are calculated for words by the `SyllableCounter`. The `PsycholinguisticAnalyser` accessed the MRC Psycholinguistic Database (Wilson 1988) and added all information found there to the document. The main work here is done within `LexicalSimplifier`. It iterates through words in decreasing order of difficulty — difficulty being measured with syllable count and frequency in the Kucera-Francis corpus — performing the following operation on each:

1. Analyse the morphology of the word. For example, `publicised=publicise+ed`.

2. The Part-of-Speech of the word will be known and so a query can be made of WordNet[4]. Here the query term is (`publicise`,`verb`).

3. The "difficulty" of each of the synonyms returned is evaluated, and the simplest one chosen. In our example, `air` would be picked.

4. The chosen word is given the inflection of the word it is to replace. So `air+ed` is calculated to be `aired`.

5. If applicable, the preceding "a"/"an" is corrected. So "*A* publicised event" becomes "*An* aired event" [5].

Extensive use is made of the tools produced for PSET (Carroll, Minnen and Pearce 2001), of the MRC Psycholinguistic Database and of WordNet (via the Java WordNet Library).

2.2.2 Background

The work done in this section is simple yet effective. Using the named entities found by ANNIE, queries are made to a biographical database[6] for the names of people and to an image search engine[7] for the names of places.

There are problems with the results of queries. Sometimes more than one result is returned when searching for biographies — some form of disambiguation is necessary. When map searches are performed for names of places, there is sometimes redundancy in the results, caused by close together places each having a map. This could be avoided by the use of a gazetteer, to find out where places are in relation to each other.

3 Training

Both the coefficients in Equation (1) and the probability model used for most features were based on the training data. The training data were provided by a fellow DUC participant[8]. They consisted of 150 documents from the DUC-2001 training data with sentences worthy of being extracted indicated — the extracts were manually chosen. Information about the extracts, and a great deal of other information about the sentences and words in the document, were appended to the original. Access to the data was via Perl, and so not ideal for our Java-based system.

Before any use of the data could be made, it had to be represented within GATE. Code was written to take the training corpus as given, and create a GATE corpus of documents.

[4] WordNet, see http://www.cogsci.princeton.edu/wn

[5] Publicised is an adjective in that phrase, so the example isn't strictly accurate.

[6] S9 Biographical Database, http://www.s9.com/biography/index.html

[7] http://www.picsearch.com

[8] John M. Conroy, Center for Computing Sciences, Institute for Defense Analyses

The seven coefficients w_f were trained using the AnnotationDiff tool of GATE. We used a simple grid search in the search space described by the convexity conditions $\sum w_f = 1$ and $w_f \geq 0$. Only half of the training corpus was used here. The system was set to produce summaries roughly equal in length to the reference summaries. The optimal coefficient arrived at, to the nearest 0.05 are described in Table 1.

Feature	Coefficient
Paragraph number	0.05
Sentence number	0.25
Overall co-reference	0.05
Length	0.35
tf.idf	0.0
XML source	0.2
Co-reference with headline	0.1

Table 1: Relative worth of features used

Repeating that training at a higher accuracy would be desirable. In the combination with other features tf.idf did not contribute to the best model, at least to the nearest 0.05. This does not necessarily mean that the tf.idf model in isolation is a weak model: It is a well-known effect from the fusion of models that the best combination model can disregard one or more of the underlying component models, even if these are competitive in isolation.

The probability model for each feature was found using all of the training corpus. The sentences within the corpus were iterated through and, for each feature value found,

$$\frac{count(f \wedge extract)}{count(f)}$$

was recorded. These values were stored as XML for use by the sentence scoring module.

The raw numbers taken for the corpus were modelled in a number of ways. First of all, some features were put into histograms, namely sentence length, mean tf.idf and overall co-reference.

Then the data were either modelled with a Gaussian model or through use of linear interpolation. To choose which model to use, a modified χ^2 test with the score

$$\sum_i \frac{((O_i + 1) - (E_i + 1))^2}{(E_i + 1)^2}$$

was used, and the model most closely fitting the original was selected. As with a usual χ^2 test, O represents the observed values — the raw data from the corpus in this case — and E is the expected values — here being the value under the model. The reason for the modifcation, simply adding one to everything, was the sparse nature of the observed values. An $O_i = 0$ would make that particular summed term independent of E_i, when ideally low E_i should cause a low contribution to the sum. Adding one to all values remedies this.

Some of the results of modelling were disregarded, either partly or entirely. For example, features like the inclusion of highly referred to terms is such that a greater value is always better. However, since the training data contained only so many values, it described a bell-shaped curve. The right hand side of the curve is just a symptom of the training data rather than indication of an underlying model. So, that part of the curve was ignored.

4 Results

The system produced summaries for all of the test documents provided. The results returned from DUC (NIST 2002) are as below.

Performance on each Quality Question The quality questions are described at (NIST 2002) — they check for mistakes in the language and presentation of the summary but not the information content. Results in Table 2 show where the mistakes were made, whilst Figure 2 takes the mean average error over all questions and compares systems.

Sentence Recall and Precision Figure 3 shows unit recall results and Figure 4 shows unit precision. Below MUs are units from the model summary, PUs are those from the system generated summary and marked PUs are those that match an MU.

$$recall = \frac{\#markedPUs}{\#MUs} \quad (3)$$

$$precision = \frac{\#markedPUs}{\#PUs} \quad (4)$$

Question #	avg. error	mean avg. error
1	0.138983	0.34878
2	0.0372881	0.0426321
3	0.0135593	0.0194625
4	0.125424	0.179487
5	0.0813559	0.0920605
6	0.0169492	0.0231696
7	0.122034	0.0898981
8	0.227119	0.189682
9	0.0101695	0.0139018
10	0.0305085	0.0278035
11	0.0440678	0.0651838
12	0.19322	0.247451

Table 2: Quality of summaries produced, by question

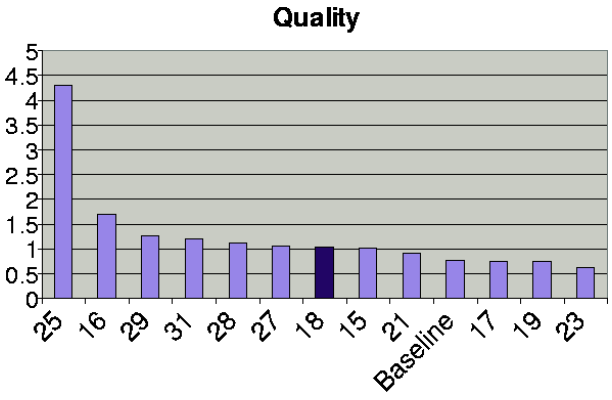

Figure 2: Quality

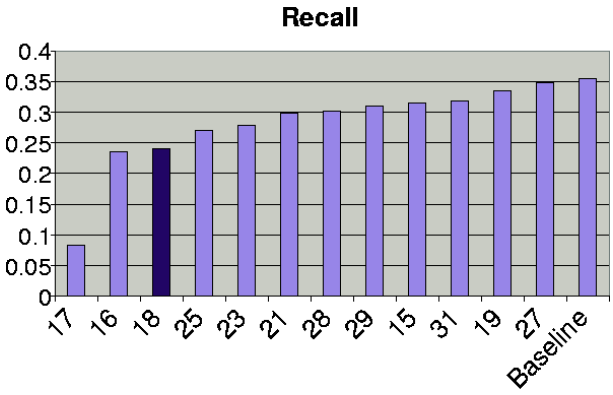

Figure 3: Recall

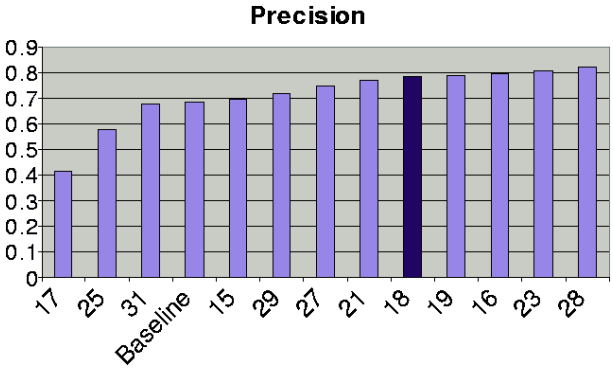

Figure 4: Precision

Unmarked Peer Unit Relevance Figure 5 states the fraction of PUs that are unmarked yet still relevant.

Coverage The mean length independent coverage is shown in Figure 6.

5 Evaluation

5.1 Summarization

The system gives good model unit precision. The quality of the summary is also good although this may be owing to the fact that our system is mainly extract-based apart from some anaphor repair.

The only three quality areas that were not performed so well on were:

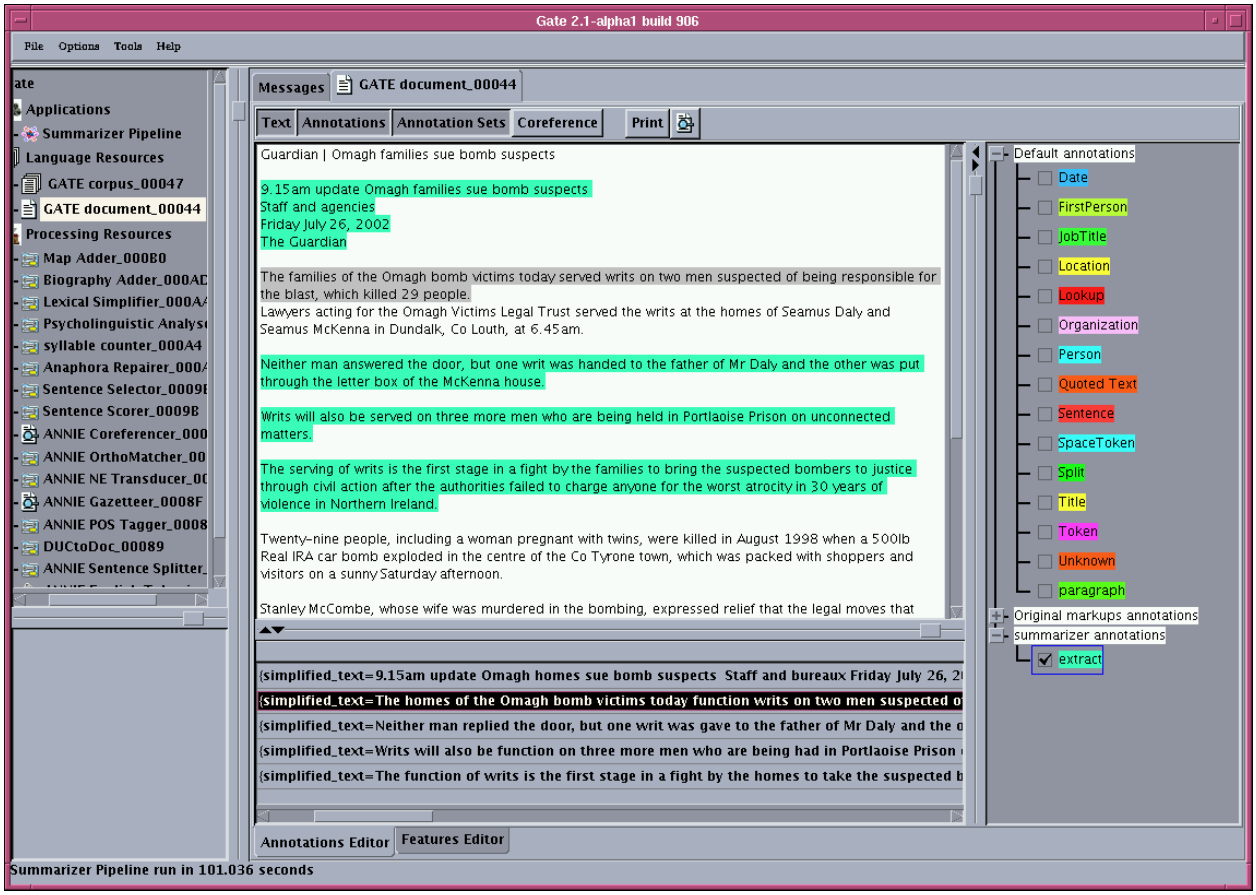

Figure 7: Screenshot of our summarizer in action

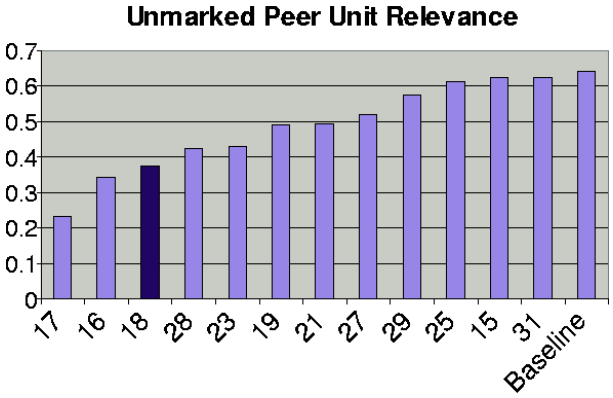

Figure 5: Unmarked PU Relevance

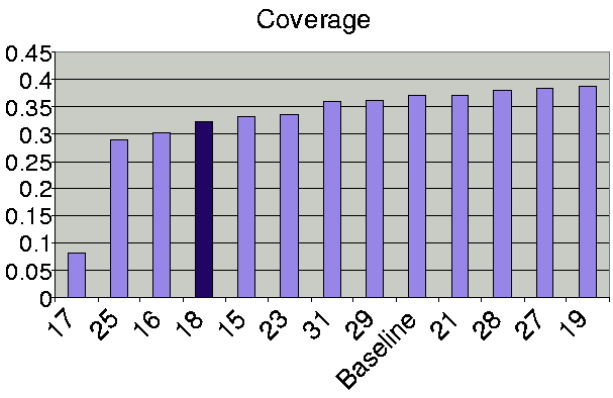

Figure 6: Coverage

Pronouns lacking antecedents (or having incorrect antecedents) Some effort was put into solving this problem (in the form of `AnaphorRepairer`). One possible reason it persisted is errors made by the GATE anaphor resolution module. Also, no attempt was made to repair "it" anaphora because of the prohibitively low precision with them.

Nouns with unclear referents Performance could be improved by replacing named entity string with the longest string that co-refers to it (derived from the NE's matches list).

Out of place "And", "However" etc. This could be solved in future by searching the summary for sentences beginning with such a word (a set would need to be defined) and including the preceding sentence in the summary if found.

5.2 Simplification

Informal trials of the lexical simplification showed some promise, although there were problems:

Incorrect sense used No word sense disambiguation was performed — the most frequent sense was always used. Disambiguating between word senses would improve performance.

Strange sounding text produced As described in (Pearce 2001), some strange sounding language can be produced, containing words that just do not sound right together. A collocation frequency table could be used here, so that only commonly used collocations are produced.

6 Conclusion

The generated summaries with our simple system seem effective and are competitive in performace with respect to other submitted systems.

The issue of directing summaries at the people who will read them is an important one — there may not be a general summary that works for all readers, on any text and in any situation. The simplification aspect is a step towards considering the reader in summary generation.

The decision to use GATE for this system appears to have been a good one. A significant amount of work was saved, by using the AnnotationDiff module. Other work was made possible where it otherwise wouldn't have been in the time available, due to ANNIE and its anaphor resolution module. Since the system produced is modular it can be reused, either wholly or in part, within other projects using GATE.

References

J Carroll, G Minnen, Y Canning, S Devlin and J Tait (1998). Practical Simplification of English Newspaper Text to Assist Aphasic Readers. In *AAAI-98 Workshop on Integrating Artificial Intelligence and Assistive Technology.*

J Carroll, G Minnen and D Pearce (2001). Applied morphological processing of English. In *Natural Language Engineering*, Volume 7, pp 207–223.

M Dimitrov (2002). A Light-weight Approach for Coreference Resolution for Named Entities in Text. Master's thesis, University of Sofia.

DUC 01 (2002). Workshop on Text Summarization. http://www-nlpir.nist.gov/projects/duc/2002.html.

J Kupiec, J O Pedersen and F Chen (1995). A Trainable Document Summarizer. In *Research and Development in Information Retrieval*, pp 68–73.

NLP group, University of Sheffield (1995–2002). GATE — General Architecture for Text Engineering. http://www.gate.ac.uk.

D Pearce (2001). Synonymy in Collocation Extraction. In *WordNet and Other Lexical Resources: Applications, Extensions and Customizations*, pp 41–46. Carnegie Mellon University, Pittsburgh.

S Sekine and C Nobata (2001). Sentence Extraction with Information Extraction

Technique. In *Workshop on Text Summarization.*

K Spärck-Jones (1998). Automatic summarising: factors and directions. In I Mani and M Maybury (Eds), *Advances in Automatic Text Summarization.* MIT press.

M D Wilson (1988). The MRC Psycholinguistic Database: Machine Readable Dictionary, Version 2. In *Behavioural Research Methods, Instruments and Computers*, Volume 20, pp 6–11. http://www.psych.rl.ac.uk/MRC_Psych_Db.html.

Acknowledgements: This work was partially supported by the EPSRC, UK.