Multi-document summarization by cluster/profile relevance and redundancy removal

Horacio Saggion and Robert Gaizauskas
Department of Computer Science
University of Sheffield
211 Portobello Street - Sheffield, England, UK, S1 4DP
Tel: +44-114-222-1947
Fax: +44-114-222-1810
saggion@dcs.shef.ac.uk

Abstract

We describe a sentence extraction system that produces two sorts of multi-document summaries: the first is a general-purpose summary of a cluster of related documents while the second is an entity-based summary of documents related to a particular person. The general-purpose summary is generated by a process that ranks sentences based on their document and cluster "worthiness". The personality-based summary is constructed by a process that ranks sentences according to a metric that uses coreference and lexical information in a person profile. In both cases, a process of redundancy removal is applied to exclude repeated information.

1 Introduction

A summary is a condensed version of a textual source having a recognisable genre and a very specific purpose: to give the reader an exact and concise idea of the contents of the source. The process of automatically producing a summary from a source text(s) consists of the following steps: (i) text interpretation; (ii) content selection; (iii) condensation ; and (iv) presentation of the summary in natural language. Our implementation of step (i) consists of the computation of a number of summarization features for each sentence in the input; step (ii) implements a sentence ranking function which is based on the combination of the features obtained in step (i); implementation of step (iii) consists of the use of an n-gram text similarity metric for detection of redundancy in text; and finally, step (iv) is a sorting function that uses document date and sentence position for deciding in which order sentences should be presented. In this work we do not use any natural language (re)generation techniques: our system is a sentence extractor.

The National Institute of Standards and Technology (NIST) with support from the Defense Advanced Research Projects Agency (DARPA) is conducting a series of evaluations in the area of text summarization, the Document Understanding Conferences (DUC), providing the appropriate framework for system-independent evaluation of text summarization systems. In DUC 2004 five summarization tasks were specified, we have concentrated on two of them: tasks 2 and 5 (refer to `http://www-nlpir.nist.gov/projects/duc/guidelines/2004.html` for a compete description of the tasks):

- task 2: Given a document cluster (Topic Detection and Tracking English clusters), create a short summary ($\leq$ 665 bytes) of it.
- task 5: Given a document cluster (TREC clusters) and a question of the form "Who is X?", where X is the name of a person, create a short summary ($\leq$ 665 bytes) of the cluster that responds to the question.

Taking as starting point our general purpose single document summarization system (Saggion, 2002; Saggion, H. and Bontcheva, K. and Cunningham, H., 2003), we have implemented a number of components to support multi-document summarization functionality.

2 Overview of the single-document summarization system

Our system is a pipeline of linguistic and statistical components, some of which are publicly available as part of GATE [1]. The system supports "generic", query-based, and multi-language (English, Finish, Swedish, Latvian, Lithuanian) summarization.
The input to the process is a single document, a compression rate specified as a percentage of the sentences of the document or as a number of words to extract, and an optional query. The document is automatically transformed by a text structure analyser into a representation containing the "text" of the original input and a number of annotation sets. Most linguistic analysis or summarization components add new information to the document in the form of new annotations or document features. Some summarisation components compute numerical features for the purpose of sentence scoring. These features are combined in order to produce the final sentence score. Sentences are output until the compression rate is reached.

2.1 Text analysis and summarization components

We perform text analysis of the document using GATE components: tokenisation, sentence splitting, gazeteer lookup, named entity recognition, part-of-speech tagging, morphological analysis, and coreference resolution.

A number of domain independent general purpose summarization components have been developed that allow the computation of several features identified as useful for content selection in text summarization (Mani, 2000).

The *corpus statistics module* computes token statistics including token frequency and lemma (or root) frequency.

The *vector space model* (Salton, 1988) module is used to create a vector representation of different text units. Each vector contains the tokens of the text unit and the value *token frequency * inverse document frequency*. For the inverse document frequencies (idf) two options are available: the system can either use a precompiled idf table (global idfs) where the frequencies may come from a huge text collection like the BNC, or use idfs values computed on-the-fly from the set of documents to be analysed (local idfs). Vector representations are produced for : (a) the whole document, (b) the lead-part of the document (the n% initial tokens of the document, where n is a parameter of the system), and (c) each sentence.

The *content-based analysis* module is used to establish the similarity between two textual units by computing the *cosine* between their vector representations (other similarity metrics will be incorporated in the future). The formula used is as follows:

$$cosine(x,y) = \frac{\sum_{i=1}^{n} w_{i,x} * w_{i,y}}{\sqrt{\sum_{i=1}^{n} (w_{i,x})^2} * \sqrt{\sum_{i=1}^{n} (w_{i,y})^2}}$$

where $w_{i,x}$ is the weight of term i in unit x and n is the numbers of terms. Different similarity values are computed for each sentence: (i) to the whole document (document sentence similarity feature); (ii) to the lead-part of the document (lead-document sentence similarity feature); (iii) to its preceding sentence (similarity forward); (iv) to its following sentence (similarity backwards); (v) to a query, and (vi) to the document title.

The *sentence position* module computes two values for each sentence. Absolute position: sentence i receives the value i^{-1}, and relative position: if the sentence is at the beginning of a paragraph, this value is set to *initial*, if the sentence is at the end of the paragraph (for paragraphs with more than one sentence), this value is set to *final*, if the sentence is in the middle of the paragraph (for paragraphs with more than

[1] `http://gate.ac.uk/`.

two sentences), this value is set to *middle*. These three values are parameters of the sentence position scorer.

Also available are modules for computing *term frequency* scores based on tf*idf, and *named entity (NE)* scores based on NE distribution in the text. All computed numeric values are normalised to a value between 0 and 1.

The final score for a sentence is computed using the following formula:

$$\sum_{i=1}^{n} feature_i * weight_i$$

where the weights are obtained experimentally and constitute parameters of the summarization process. The scores are used to produce a ranked list of sentences. Sentences in the ranked list are included in the summary until the compression rate is reached. A module is also available that allows the user to specify "text units" section headings, for example that should be excluded from the ranked list.

3 Multi-document summarization

In a multi-document situation, the question is how to measure not only the content of each sentence in relation to the other sentences in the same document but also across documents. We have followed two different approaches for addressing tasks 2 and 5 dictated by the tasks inherent differences.

- In task 2, apart from the input document cluster, no additional information ("topic") is given. This suggests a the use of general purpose technique to measure sentence worthiness.
- In task 5, apart from the input document cluster, we are given a *key entity* as constraint. This suggest that a more informed (or knowledge-based method) could be used to exploit that additional constraint.

In a general multi-document situation, we need to take into consideration the relationship each sentence has to the set of documents (cluster) that constitute the input to the process. Following a similar approach to the single document case, general purpose techniques are used. A *centroid* representation of the cluster of n related documents is constructed. It is a vector of pairs of terms and weights, where the weight w_i of term i in the centroid is obtained as follows:

$$w_i = \frac{\sum_{k=1}^{n} w_{i,k}}{n}$$

where $w_{i,k}$ is the weight of term i in document k. A cosine similarity value is computed between each sentence in the document set and the centroid. This is the only multi-document summarization feature that we have implemented for task 2.

Each sentence in the cluster is scored using the features: (i) sentence cluster similarity, (ii) sentence lead-document similarity, (iii) absolute document position. These values are combined with appropriate weights to produce the sentences final score which is used to rank them.

For task 5, we needed an approach that would take into account the input constraint: the fact that a summary about a particular person is to be produced. We have explored three ideas: identify and measure references of the key entity in a sentence; identify and measure if person facets or characteristics are referred to in a sentence; and identify and measure mention of information associated with the key entity. All this is explained in detail below:

- presence in a sentence of a key entity in full ("Robert Rubin"), alias ("Rubin") or pronoun ("him") coreferent with key entity is detected by string matching and coreference resolution. Using the distribution of male and female pronouns in the cluster the system guesses the gender of the key entity: male, female, or neutral and provides that information to the general purpose coreference resolution algorithm. This is done to boost the performance of the coreference resolution algorithm that makes

mistakes when named entities are not correctly classified by the named entity recogniser. During testing, this method identified the correct gender in 94% of the cases, assigned the "neutral" gender in 4% of the cases and made a mistake in 2% of the cases.

Mention	**Score for DUC 2004**
Key entity	$w_{key} = 1.00$
Alias entity	$w_{alias} = 0.50$
Coreferent pronoun	$w_{pronoun} = 0.25$

Table 1: Scores associated to entities in the text

A coreference score is associated with each sentence and is computed as:

$$CorefScore(S) = count_{key}(S) * w_{key} + count_{alias}(S) * w_{alias} + count_{pronoun}(S) * w_{pronoun}$$

where $count_{key}$, $count_{alias}$, and $count_{pronoun}$ give, respectively, the number of in-full, alias or coreferent mentions of the key entity, and w_{key}, w_{alias}, and $w_{pronoun}$ are weights associated with the mentions, set as in Table 1 for DUC 2004.

- presence of general profile words or expressions; a shallow representation of why a person might be important was manually created based on our intuitions. The representation is constructed around a number of "facets" of a person's life. In Table 2, we present the name of the facets we have specified and lexical information expressing the facets.

Facet	**Lexical information**
achievement	discover, invent, awarded, etc.
background	upper-class family, poor, etc.
education	college, university, etc.
ethnicity	latino, etc.
life event	born, die, marry, etc.
nationality	Argentinian, etc.
opinion	detested, loved, etc.
political thinking	facist, marxist, etc.
religion	muslim, catholic, etc.
sexual orientation	gay, bi-sexual, ect.
tragedy	drug addict, suicide, etc.
relevant person	writer, musician, etc.
temporal expression	in DATE, from DATE, etc.

Table 2: Facets in a person profile

A combined score is computed based on the presence of general profile terms:

$$GeneralProfileScore(S) = count_{facet}(S)$$

where $count_{facet}$ gives the number of faceted terms found in sentence S.

- presence of key entity related term. One of the ideas we wanted to explore was whether having "precompiled knowledge" about the key entity would lead to an increase performance compared with not having that knowledge. Knowledge about a particular person is implemented as a list of terms likely to co-occur in profiles of that person. The approach taken is based on our work on definitional QA (Saggion and Gaizauskas, 2004) where terms that co-occur with mentions of the key entity are mined

from Web pages. We used 14 patterns (see Table 3) to identify pages and sentences profiling the key entity from which we extract candidate terms (Table 4, shows part of "Robert Rubin" profile). A score is computed as follows:

$$InstanceProfileScore(S) = count_{terms}(S)$$

where $count_{terms}$ gives the number of entity related terms found in sentence S.

All scores are normalised on a second pass.

Person Pattern
KEY is a
KEY born in
KEY considered
KEY the first

Table 3: Some patterns to identify person profiles in the Web

Instance Profile
secretary; treasury; president; clinton; citigroup; chairman; ahtisaari; corzine; ...

Table 4: Some terms in "Robert Rubin" profile.

Since DUC 2004 allowed 3 runs per site, three configurations of the system were created. One of them called "Simple" was identical to the configuration used for task 2. The other two configurations used the following scoring formula to rank sentences in the cluster:

$$SentScore(S) =$$
$$w_{coref} * CorefScore(S) * (1 + w_{instance} * InstanceProfileScore(S) + w_{general} * GeneralProfileScore(S))$$

where w_{coref}, $w_{instance}$, and $w_{general}$ are weights. The configuration called knowledge-based, used $w_{instance} > 0$, while the configuration called generic used $w_{instance} = 0$.

For both tasks, sentences are examined in rank order and put in a candidate sentence set unless they are too similar to a sentences already in the candidate set. The process ends when the desired compression rate is reached.

System	R1		R2		R3		R4		RL		RW	
Centroid	0.3706	(5/35)	0.0829	(7/35)	0.0284	(5/35)	0.0121	(6/35)	0.3744	(8/35)	0.1293	(8/35)
Baseline	0.3242	(25/35)	0.0641	(24/35)	0.0199	(23/35)	0.0067	(26/35)	0.3459	(22/35)	0.1186	(21/35)
Best	0.3822	(1/35)	0.0922	(1/35)	0.0353	(1/35)	0.0166	(1/35)	0.3895	(1/35)	0.1338	(1/35)
Worst	0.2419	(35/35)	0.0188	(35/35)	0.0028	(35/35)	0.0008	(35/35)	0.2763	(35/35)	0.0936	(35/35)

Table 5: ROUGE for task 2. Results are presented for our configuration (Centroid), DUC baseline, best and worst results. (N/M) means the system obtained rank N among M configurations.

System	SEE coverage	
Centroid	0.2622	(2/17)
Baseline	0.1996	(12/17)
Best	0.3030	(1/17)
Worst	0.0490	(17/17)

Table 6: SEE-coverage for task 2. Results are presented for our configuration (Centroid), DUC baseline, best and worst results.(N/M) means the system obtained rank N among M configurations.

System	Q1	Q2	Q3	Q4	Q5	Q6	Q7
Centroid	2.82	2.56	2.26	1.64	1.58	1.42	1.44
Baseline	1.44	2.26	1.34	1.30	1.02	1.30	1.32
Best	1.44	2.08	1.34	1.20	1.02	1.22	1.20
Worst	4.82	4.54	2.26	4.66	1.62	4.30	2.70

Table 7: Linguistic questions evaluation for task 2. Results are presented for our configuration (Centroid), DUC baseline, best and worst results.

3.1 Similarity metric for redundancy detection

We take a shallow approach to the detection of similar information in texts by considering a metric that relies on n-gram overlap between text units. Having computed n-grams sets for each document in the input, the n-gram based similarity metric between two text fragments T_1 and T_2 is computed as follows:

$$NGramSim(T_1, T_2, n) = \sum_{k=1}^{n} w_k * \frac{|grams(T_1,k) \cap grams(T_2,k)|}{|grams(T_1,k) \cup grams(T_2,k)|}$$

where n means that n-grams 1, 2, ... n are to be considered, $grams(T, k)$ is the set of k-grams of fragment T, and w_k is the weight associated with the k-gram similarity of two sets. For DUC 2004, we chose n=4 and the arithmetic series $w_1 = 0.1$, $w_2 = 0.2$, $w_3 = 0.3$, and $w_4 = 0.4$ as weighting scheme. We consider two units T_1 and T_2 to be similar if $NGramSim(T_1, T_2, n) \geq \alpha$, α was set to 0.1 for DUC 2004.

Expecting to achieve a certain "degree of coherence", the candidate sentences are presented in a summary sorted by document date (least recent first) and by document position ("lead based").

While he was in an Atlanta jail, Kopp was given the nickname Atomic Dog, which investigators contend links him to the violent fringe of the anti-abortion movement, responsible for a series of bombings and arsons and seven murders of abortion providers like Slepian over the last five years. James Kopp, the man the FBI is seeking as a material witness in the sniper slaying of Dr. Barnett Slepian, is known to abortion rights leaders as an aggressive anti-abortion protester, and law enforcement officials say he has been arrested several times in demonstrations at abortion clinics. Investigators said Kopp's car was seen near the Amherst, N.Y., home of Dr. Barnett Slepian in the weeks before the doctor, whose work at an abortion clinic had long made him a target of harassment, was killed.

Figure 1: Multi-document summary for cluster d31013t (DUC2004/Task 2)

3.2 Evaluation, results and discussion

Figure 1 shows the summary generated by the system for cluster d31013t (task 2) and Figure 2 shows one of the summaries generated for cluster d132d about "Robert Rubin" (task 5).

Concerning content evaluation, summaries were assessed by human assessors using model (reference) summaries and the SEE evaluation tool (see `http://duc.nist.gov/duc2004/protocol.html`). The result of the evaluation is a coverage score for each summary. Summaries were also evaluated for content using the

System	R1		R2		R3		R4		RL		RW	
KB	0.3341	(8/23)	0.0723	(14/23)	0.0279	(13/23)	0.0131	(12/23)	0.3320	(17/23)	0.1130	(17/23)
Generic	0.3286	(13/23)	0.0734	(13/23)	0.0284	(9/23)	0.0143	(6/23)	0.3270	(18/23)	0.1117	(18/23)
Simple	0.3350	(6/23)	0.0738	(12/23)	0.0266	(14/23)	0.0128	(13/23)	0.3386	(14/23)	0.1149	(14/23)
Baseline	0.3136	(17/23)	0.0626	(19/23)	0.0209	(20/23)	0.0096	(19/23)	0.3395	(13/23)	0.1146	(15/23)
Best	0.3550	(1/23)	0.0857	(1/23)	0.0328	(1/23)	0.0164	(1/23)	0.3733	(1/23)	0.1267	(1/23)
Worst	0.2629	(23/23)	0.0487	(23/23)	0.0152	(23/23)	0.0067	(23/23)	0.2843	(23/23)	0.0984	(23/23)

Table 8: ROUGE evaluation for task 5. Results are presented for our three configurations (KB, Generic, and Simple), DUC baseline, best and worst results. (N/M) means the system obtained rank N among M configurations

System	SEE coverage	
KB	0.1891	(11/15)
Baseline	0.1903	(10/15)
Best	0.2414	(1/15)
Worst	0.1449	(15/15)

Table 9: SEE evaluation for task 5. Results are presented for our main configuration (KB), DUC baseline, best and worst results. (N/M) means the system obtained rank N among M configurations.

ROUGE package (see ROUGE working notes `http://www.isi.edu/~cyl/ROUGE`) which allows the computation of recall-based metrics using n-gram matching between a candidate summary and a reference set (ideal summaries). ROUGE-n (or Rn) is n-gram recall, ROUGE-L (or RL) is based on the longest common subsequence and ROUGE-W (RW) is a weighted longest common subsequence that takes into account distances when applying the longest common subsequence. Concerning content quality, summaries were assessed by humans who provided scores for 7 quality questions ("Does the summary build from sentence to sentence to a coherent body of information about the topic?", "To what degree does the summary say the same thing over again?", etc.). Scores awarded range from 1 (best) to 5 (worst).

For task 2, our results are presented in Table 5 for ROUGE, Table 6 for SEE-coverage, and Table 7 for question quality. All tables provide system ranks. Overall, the configuration for task 2 seems stable across ROUGE metrics and obtained a high rank in SEE human evaluation (2nd score). In what text quality is concerned, our scores for all questions but Q3, are somewhere in the middle always worst than the baseline.

For task 5, our results are presented in Table 8 for ROUGE, Table 9 for SEE-coverage, and Table 10 for question quality. In ROUGE evaluation configurations KB and Generic change from high to low ranks when a more informed metric (i.e., RW) is used. Configuration Simple (the same as in task 2) is more stable across metrics. All systems obtained ranks very close to the baseline, which is disappointing. Configuration KB performed just bellow the baseline in SEE-coverage evaluation. In what text quality is concerned, our scores

U.S. Treasury Secretary Robert Rubin arrived in Malaysia Sunday for a two-day visit to discuss the regional economic situation, the U.S. Embassy said. Robert Rubin, before joining the Clinton administration, built his career at Goldman, Sachs amp Co., trading stakes in companies as head of that Wall Street firm's arbitrage desk. A ledeall: Robert Rubin resigns as Treasury secretary; Clinton will name Lawrence Summers, Rubin's deputy, to succeed him. Robert Rubin's decision to step aside as Treasury Secretary was responsibly timed in that the major crises in the world now are not primarily economic. During the global financial crisis, Rubin has halfheartedly warned Americans not to invest cavalierly in weak foreign economies, but anytime foolhardy American investors have been threatened, Rubin has rushed to save them.

Figure 2: Knowledge-based multi-document summary for cluster d132d, topic="Robert Rubin" (DUC2004/Task 5)

System	Q1	Q2	Q3	Q4	Q5	Q6	Q7
KB	3.52	3.22	1.62	2.12	1.94	2.76	2.18
Baseline	1.62	2.20	1.44	1.44	1.10	1.40	1.82
Best	1.62	2.20	1.16	1.24	1.10	1.22	1.30
Worst	4.52	4.04	1.98	3.40	2.42	2.76	3.54

Table 10: Linguistic questions evaluation for task 5. Results are presented for our configuration (KB), DUC baseline, best and worst results.

for all questions but Q6, are somewhere in the middle always worst than the baseline. After submission, we discovered formatting errors in the KB and Generic configurations (initial parts of sentences as well as final full stop were chopped). This not only makes the summary difficult to read, but we think very difficult to evaluate. We are evaluating whether this has had an effect in the ROUGE numbers the systems obtained.

4 Future work

Our main objective for next year is to improve our personality-based multi-document summarizer by improving our specification of a profile. We would probably use machine learning techniques to assess the "value" of the profile components on training data. The human summaries produced by DUC provide an appropriate starting point for identifying what makes a good summary about an entity. We also plan to include other types of instances such as organisations or groups in our future system.

For DUC 2004, we were experimenting with a syntax-based headline generation system, unfortunately there was no time to test it in any appropriate way, but we plan to experiment with it for the next DUC.

References

Mani, I. (2000). *Automatic Text Summarization*. John Benjamins Publishing Company.

Saggion, H. (2002). Shallow-based Robust Summarization. In *Automatic Summarization: Solutions and Perspectives*, ATALA.

Saggion, H. and Gaizauskas, R. (2004). Mining on-line sources for definition knowledge. In *Proceedings of the 17th FLAIRS 2004*, Miami Bearch, Florida, USA. AAAI.

Saggion, H. and Bontcheva, K. and Cunningham, H. (2003). Generic and Query-based Summarization. In *European Conference of the Association for Computational Linguistics (EACL) Research Notes and Demos*, Budapest, Hungary. EACL.

Salton, G. (1988). *Automatic Text Processing*. Addison-Wesley Publishing Company.