Information Processing and Management 40 (2004) 919–938

INFORMATION PROCESSING & MANAGEMENT

www.elsevier.com/locate/infoproman

Centroid-based summarization of multiple documents

Dragomir R. Radev [a,*], Hongyan Jing [b], Małgorzata Styś [b], Daniel Tam [a]

[a] University of Michigan, Ann Arbor, MI 48109, USA
[b] IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, USA

Received 5 January 2003; accepted 24 October 2003
Available online 13 December 2003

Abstract

We present a multi-document summarizer, MEAD, which generates summaries using cluster centroids produced by a topic detection and tracking system. We describe two new techniques, a centroid-based summarizer, and an evaluation scheme based on sentence utility and subsumption. We have applied this evaluation to both single and multiple document summaries. Finally, we describe two user studies that test our models of multi-document summarization.

© 2003 Elsevier Ltd. All rights reserved.

Keywords: Multi-document summarization; Centroid-based summarization; Cluster-based relative utility; Cross-sentence informational subsumption

1. Introduction

On April 18, 2002, a small plane hit the 26th floor of the 30-story high Pirelli building in downtown Milan, Italy. The flight had originated in nearby Switzerland and was on a route to Italy. According to local hospitals, three people in the building and the pilot had died in the crash while 60 additional people were being treated for injuries. The cause of the crash was an apparent suicide attempt while Italian officials did not rule out the possibility that it was a terrorist act.

[*] Corresponding author.

E-mail addresses: radev@umich.edu (D.R. Radev), hjing@us.ibm.com (H. Jing), sm1@us.ibm.com (M. Styś), dtam@umich.edu (D. Tam).

0306-4573/$ - see front matter © 2003 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ipm.2003.10.006

The paragraph above summarizes a large amount of news from different sources. While it was not automatically generated, one can imagine the use of such automatically generated summaries. In this paper we will describe how multi-document summaries are built and evaluated.

1.1. Topic detection and multi-document summarization

To generate a summary, one must first start with relevant documents that one wishes to summarize. The process of identifying all articles on an emerging event is called *Topic Detection and Tracking* (TDT). A large body of research in TDT has been created over the past years (Allan, Papka, & Lavrenko, 1998). We will present an extension of our own research on TDT (Radev, Hatzivassiloglou, & McKeown, 1999) that we used in our summarization of multi-document clusters. The main concept we used to identify documents in TDT is also used to rank sentences for our summarizer.

Our entry in the official TDT evaluation, CIDR (Radev et al., 1999), uses modified TF * IDF to produce clusters of news articles on the same event ('TF' indicates how many times a word appears in a document while IDF measures what percentage of all documents in a collection contain a given word). An incoming document is grouped into an existing cluster, if the TF * IDF of the new document is close to the centroid of the cluster. A centroid is a group of words that statistically represent a cluster of documents. The idea of a centroid is described further in Section 3.

From a TDT system, an *event cluster* can be produced. An event cluster consists of chronologically ordered news articles from multiple sources. These articles describe an event as it develops over time. In our experiments, event clusters range from 2 to 10 documents. It is from these documents that summaries can be produced.

We developed a new technique for multi-document summarization, called *centroid-based summarization* (CBS). CBS uses the centroids of the clusters produced by CIDR to identify sentences central to the topic of the entire cluster. We have implemented CBS in MEAD, our publicly available multi-document summarizer.

A key feature of MEAD is its use of cluster centroids, which consist of words which are central not only to one article in a cluster, but to *all* the articles. While TF * IDF has been implemented in single-document summarizer, (e.g. Aone, Okurowski, Gorlinsky, & Larsen, 1997), ours is the first attempt to expand that idea to multi-document summarization.

MEAD is significantly different from previous work on multi-document summarization (Carbonell & Goldstein, 1998; Mani & Bloedorn, 2000; McKeown, Klavans, Hatzivassiloglou, Barzilay, & Eskin, 1999; Radev & McKeown, 1998), which use techniques such as graph matching, maximal marginal relevance, or language generation.

Finally, evaluation of multi-document summaries is a difficult problem. Currently, there is no widely accepted evaluation scheme. We propose a utility-based evaluation scheme, which can be used to evaluate both single-document and multi-document summaries.

The main contributions of this paper are: the use of *cluster-based relative utility* (CBRU) and *cross-sentence informational subsumption* (CSIS) for evaluation of single and multi-document summaries, the development of a centroid-based multi-document summarizer, two user studies that support our findings, and an evaluation of MEAD.

2. Informational content of sentences

2.1. Cluster-based relative utility (CBRU)

Cluster-based relative utility (CBRU, or relative utility, RU in short) refers to the degree of relevance (from 0 to 10) of a particular sentence to the general topic of the entire cluster (for a discussion of what is a topic, see Allan, Carbonell, Doddington, Yamron, & Yang, 1998). A utility of 0 means that the sentence is not relevant to the cluster and a 10 marks an essential sentence. Evaluation systems could be built based on RU and thus provide a more quantifiable measure of sentences.

2.2. Cross-sentence informational subsumption (CSIS)

A related notion to RU is *cross-sentence informational subsumption* (CSIS, or subsumption). CSIS reflects that certain sentences repeat some of the information present in other sentences and may, therefore, be omitted during summarization. If the information content of sentence **a** (denoted as $\mathbf{i}(\mathbf{a})$) is contained within sentence **b**, then **a** becomes informationally redundant and the content of **b** is said to *subsume* that of **a**:

$$\mathbf{i}(\mathbf{a}) \subset \mathbf{i}(\mathbf{b})$$

In the example below, (2) subsumes (1) because the crucial information in (1) is also included in (2) which presents additional content: “the court”, “last August”, and “sentenced him to life”.

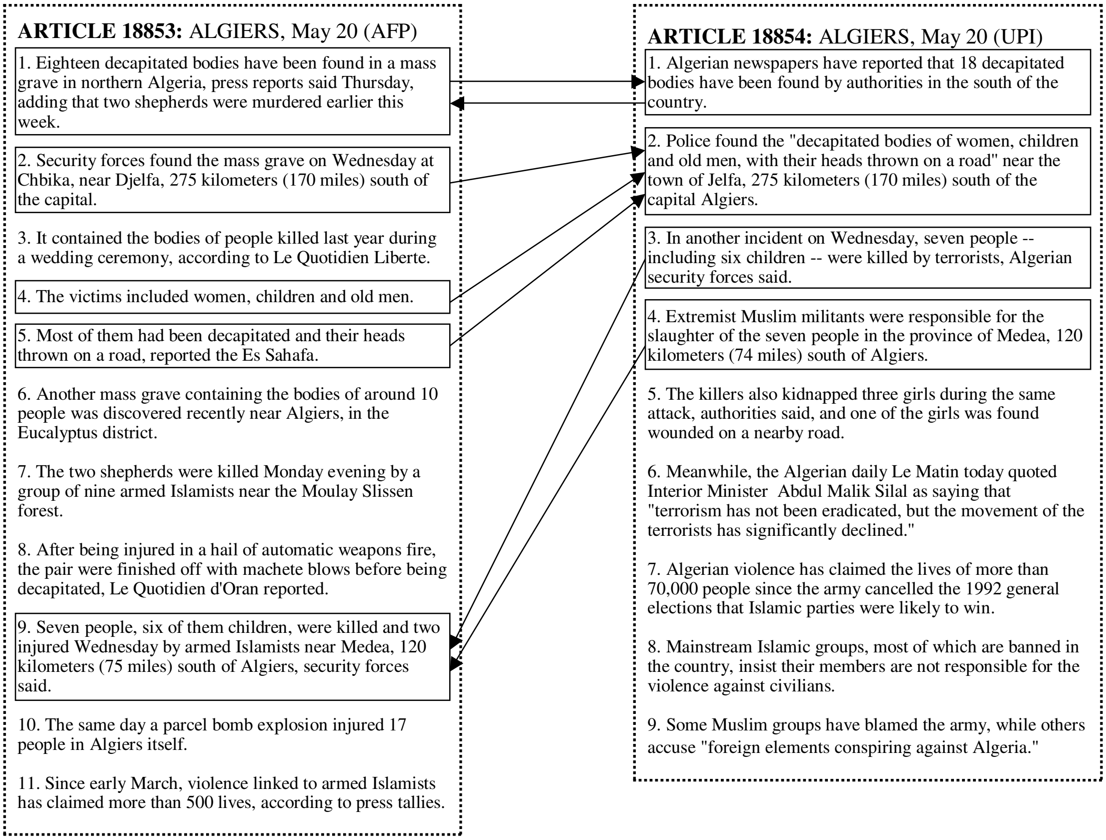
(1) John Doe was found guilty of the murder.
(2) The court found John Doe guilty of the murder of Jane Doe last August and sentenced him to life.

The cluster shown in Fig. 1 shows subsumption links across two articles about terrorist activities in Algeria (ALG 18853 and ALG 18854).

An arrow from sentence A to sentence B indicates that the information content of A is subsumed by the information content of B. Sentences 2, 4, and 5 from the first article repeat the information from sentence 2 in the second article, while sentence 9 from the former article is later repeated in sentences 3 and 4 of the latter article.

2.3. Equivalence classes of sentences

Sentences subsuming each other are said to belong to the same *equivalence class*. An equivalence class may contain more than two sentences within the same or different articles. In the following example, although sentences (3) and (4) are not exact paraphrases of each other, they can be substituted for each other without crucial loss of information and therefore belong to the



ARTICLE 18853: ALGIERS, May 20 (AFP)

1. Eighteen decapitated bodies have been found in a mass grave in northern Algeria, press reports said Thursday, adding that two shepherds were murdered earlier this week.

2. Security forces found the mass grave on Wednesday at Chbika, near Djelfa, 275 kilometers (170 miles) south of the capital.

3. It contained the bodies of people killed last year during a wedding ceremony, according to Le Quotidien Liberte.

4. The victims included women, children and old men.

5. Most of them had been decapitated and their heads thrown on a road, reported the Es Sahafa.

6. Another mass grave containing the bodies of around 10 people was discovered recently near Algiers, in the Eucalyptus district.

7. The two shepherds were killed Monday evening by a group of nine armed Islamists near the Moulay Slissen forest.

8. After being injured in a hail of automatic weapons fire, the pair were finished off with machete blows before being decapitated, Le Quotidien d'Oran reported.

9. Seven people, six of them children, were killed and two injured Wednesday by armed Islamists near Medea, 120 kilometers (75 miles) south of Algiers, security forces said.

10. The same day a parcel bomb explosion injured 17 people in Algiers itself.

11. Since early March, violence linked to armed Islamists has claimed more than 500 lives, according to press tallies.

ARTICLE 18854: ALGIERS, May 20 (UPI)

1. Algerian newspapers have reported that 18 decapitated bodies have been found by authorities in the south of the country.

2. Police found the "decapitated bodies of women, children and old men, with their heads thrown on a road" near the town of Jelfa, 275 kilometers (170 miles) south of the capital Algiers.

3. In another incident on Wednesday, seven people -- including six children -- were killed by terrorists, Algerian security forces said.

4. Extremist Muslim militants were responsible for the slaughter of the seven people in the province of Medea, 120 kilometers (74 miles) south of Algiers.

5. The killers also kidnapped three girls during the same attack, authorities said, and one of the girls was found wounded on a nearby road.

6. Meanwhile, the Algerian daily Le Matin today quoted Interior Minister Abdul Malik Silal as saying that "terrorism has not been eradicated, but the movement of the terrorists has significantly declined."

7. Algerian violence has claimed the lives of more than 70,000 people since the army cancelled the 1992 general elections that Islamic parties were likely to win.

8. Mainstream Islamic groups, most of which are banned in the country, insist their members are not responsible for the violence against civilians.

9. Some Muslim groups have blamed the army, while others accuse "foreign elements conspiring against Algeria."

Fig. 1. Cross-sentence informational subsumption (example).

same equivalence class, i.e. $\mathbf{i}(\mathbf{3}) \subset \mathbf{i}(\mathbf{4})$ and $\mathbf{i}(\mathbf{4}) \subset \mathbf{i}(\mathbf{3})$. In the user study section (Section 5) we will take a look at the way humans perceive CSIS and equivalence class.

(3) Eighteen decapitated bodies have been found in a mass grave in northern Algeria, press reports said Thursday.
(4) Algerian newspapers have reported on Thursday that 18 decapitated bodies have been found by the authorities.

2.4. Comparison with MMR

Maximal marginal relevance (or MMR) is a technique similar to CSIS and was introduced in (Carbonell & Goldstein, 1998). In that paper, MMR is used to produce summaries of single documents that avoid redundancy. The authors mention that their preliminary results indicate that multiple documents on the same topic also contain redundancy but they fall short of using MMR for multi-document summarization. Their metric is used as an enhancement to a query-based summary; CSIS is designed for query-independent and therefore generic summaries.

3. MEAD—a centroid-based multi-document summarizer

3.1. What is a centroid

A centroid is a set of words that are statistically important to a cluster of documents. As such, centroids could be used both to classify relevant documents and to identify salient sentences in a cluster. We will explain how we utilized centroids in both our TDT algorithm and our multi-document summarizer, MEAD.

3.2. Centroid-based clustering

Relative documents are grouped together into clusters by the algorithm described in detail in (Radev et al., 1999). Each document is represented as a weighted vector of TF $*$ IDF. CIDR first generates a centroid by using only the first document in the cluster. As new documents are processed, their TF $*$ IDF values are compared with the centroid using the formula described below. If the similarity measure $\mathrm{sim}(D, C)$ is within a threshold, the new document is included in the cluster (Radev et al., 1999).

$$\mathrm{sim}(D,C) = \frac{\sum_k (d_k * c_k * \mathrm{idf}(k))}{\sqrt{\sum_k (d_k)^2}\sqrt{\sum_k (c_k)^2}}$$

Fig. 2 gives a pictorial explanation of the algorithm: suppose cosine α is within a threshold, then document 1 is included in the cluster. The "terms" on the axis are the words that make up the centroid. See Table 2 for the top 10 words in the centroid for cluster A.

3.3. Description of the corpus

To better illustrate the centroid-based algorithms, we will use examples from our experiments. We prepared a small corpus consisting of a total of 558 sentences in 27 documents, organized in 6 clusters (Table 1), all extracted by CIDR. Four of the clusters are from Usenet newsgroups. The remaining two clusters are from the official TDT corpus. [1] Among the factors for our selection of clusters are: coverage of as many news sources as possible, coverage of both TDT and non-TDT data, coverage of different types of news (e.g., terrorism, internal affairs, and environment), and diversity in cluster sizes (in our case, from 2 to 10 articles). The test corpus is used in the evaluation in such a way that each cluster is summarized at nine different compression rates, thus giving nine times as many sample points as one would expect from the size of the corpus.

[1] The selection of cluster E is due to an idea by the participants in the Novelty Detection Workshop, led by James Allan.

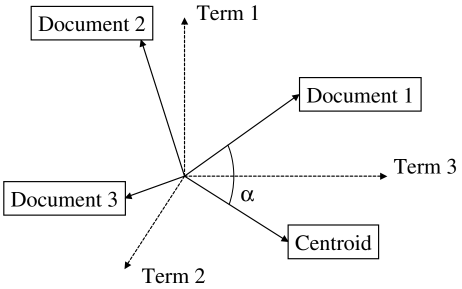


Fig. 2. Conceptual centroid representation.

Table 1
Corpus composition

Cluster	#Docs	#Sent	Source	News sources	Topic
A	2	25	clari.world.africa.northwestern	AFP, UPI	Algerian terrorists threaten Belgium
B	3	45	clari.world.terrorism	AFP, UPI	The FBI puts Osama bin Laden on the most wanted list
C	2	65	clari.world.europe.russia	AP, AFP	Explosion in a Moscow apartment building (09/09/1999)
D	7	189	clari.world.europe.russia	AP, AFP, UPI	Explosion in a Moscow apartment building (09/13/1999)
E	10	151	TDT-3 corpus topic 78	AP, PRI, VOA	General strike in Denmark
F	3	83	TDT-3 corpus topic 67	AP, NYT	Toxic spill in Spain

3.4. Cluster centroids

Table 2 shows a sample centroid, produced by CIDR (Radev et al., 1999) from cluster A. The “TF” column indicates the term frequency, or average number of occurrences of a word across the entire cluster. For example, a TF value of 5.5 for two documents indicates that the term (“Belgium” in Table 2) appears 11 times in the cluster. The IDF (Invert Document Frequency) values were computed from the TDT corpus. A centroid, in this context, is a pseudo-document which consists of words which have Count ∗ IDF scores above a pre-defined threshold in the documents that constitute the cluster. CIDR computes Count ∗ IDF in an iterative fashion, updating its values as more articles are inserted in a given cluster. We hypothesize that sentences containing words from the centroid are more indicative of the topic of the cluster.

Table 2
Cluster centroid for cluster A

Term	TF	IDF	TF * IDF
Belgium	5.5	5.60	30.81
Islamic	3.0	9.80	29.42
GIA	7.0	3.00	21.00
Arabic	1.50	9.11	13.67
Jailed	2.00	6.76	13.52
Al	1.50	7.17	10.75
Hardline	1.00	9.81	9.81
Statement	2.50	3.84	9.61
Torture	1.00	8.42	8.42
Threat	1.50	5.44	8.15

3.5. MEAD extraction algorithm

MEAD decides which sentences to include in the extract by ranking them according to a set of parameters. The input to MEAD is a cluster of articles (e.g., extracted by CIDR), segmented into sentences and a value for the compression rate R. The output is a sequence of $n * r$ sentences from the original documents presented in the same order as the input documents. For example, if the cluster contains a total of 50 sentences ($n = 50$) and the value of R is 20%, the output of MEAD will contain 10 sentences. Sentences appear in the extract in the same order as the original documents are ordered chronologically. We benefit here from the time stamps associated with each document.

We used three features to compute the salience of a sentence: Centroid value, Positional value, and First-sentence overlap. These are described in full below.

3.5.1. Centroid value

The centroid value C_i for sentence S_i is computed as the sum of the centroid values $C_{w,i}$ of all words in the sentence. For example, the sentence "President Clinton met with Vernon Jordan in January" would get a score of 243.34 which is the sum of the individual centroid values of the words (clinton = 36.39; vernon = 47.54; jordan = 75.81; january = 83.60).

$$C_i = \sum_w C_{w,i}$$

3.5.2. Positional value

The positional value is computed as follows: the first sentence in a document gets the same score $C_{\max}$ as the highest-ranking sentence in the document according to the centroid value. The score for all sentences within a document is computed according to the following formula:

$$P_i = \frac{(n - i + 1)}{n} * C_{\max}$$

3.5.3. First-sentence overlap

The overlap value is computed as the inner product of the sentence vectors for the current sentence i and the first sentence of the document. The sentence vectors are the n-dimensional representations of the words in each sentence, whereby the value at position i of a sentence vector indicates the number of occurrences of that word in the sentence.

$$F_i = \overrightarrow{S_1}\,\overrightarrow{S_i}$$

3.5.4. Combining the three parameters

We tested several sentence weighting techniques using linear combinations of three parameters: words in centroid (C), sentence position (P), and words in title or first sentence (F). The score of a sentence is the weighted sum of the scores for all words in it. Since we have not incorporated learning the weights automatically, in this paper we used an equal weight for all three parameters. Thus, we use the following SCORE values to approximate cluster-based relative utility, where i is the sentence number within the cluster.

$\mathrm{SCORE}(s_i) = w_c C_i + w_p P_i + w_f F_i$
INPUT: Cluster of d documents with n sentences (compression rate $= r$)
OUTPUT: $(n * r)$ sentences from the cluster with the highest values of SCORE.

The current paper evaluates various SCORE functions. These are discussed in Section 5.3.

3.6. Redundancy-based algorithm

We try to approximate CSIS by identifying sentence similarity across sentences. Its effect on MEAD is the subtraction of a *redundancy penalty* (R_s) for each sentence which overlaps with sentences that have higher SCORE values. The redundancy penalty is similar to the negative factor in the MMR formula (Carbonell & Goldstein, 1998).

$$\mathrm{SCORE}(s_i) = w_c C_i + w_p P_i + w_f F_i - w_R R_s$$

For each pair of sentences extracted by MEAD, we compute the cross-sentence word overlap according to the following formula: [2]

$$R_s = 2 * (\#\text{overlapping words})/(\#\text{words in sentence1} + \#\text{words in sentence2})$$
$$w_R = \mathrm{Max}_s(\mathrm{SCORE}(s))$$

where SCORE(s) is computed according to the formula in Section 3.5.4.

As an example, since sentence (3) has 6 words common to sentence (4) in Section 2.3, $R_3 = 2 * 6(18 + 16) = 0.35$.

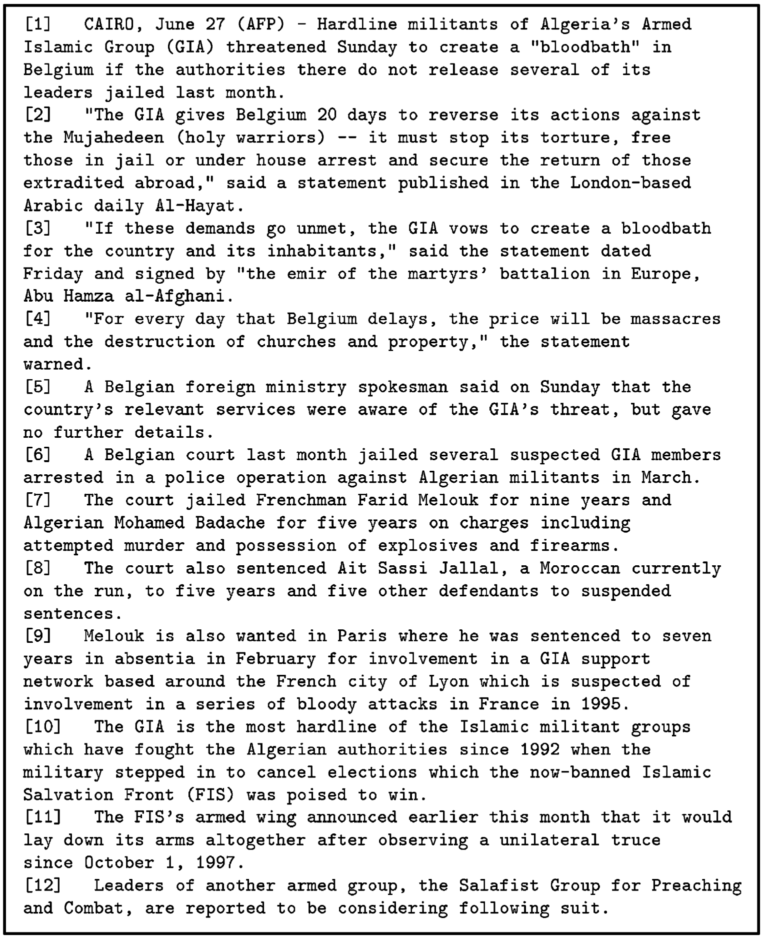
[2] Stop words are also counted.

Note that $R_s = 1$ when the sentences are identical and $R_s = 0$ when they have no words in common. After deducting R_s, we rerank all sentences and possibly create a new sentence extract. We repeat this process until reranking does not result in a different extract.

The number of overlapping words in the formula is computed in such a way that if a word appears m times in one sentence and n times in another, only $\min(m, n)$ of these occurrences will be considered overlapping.

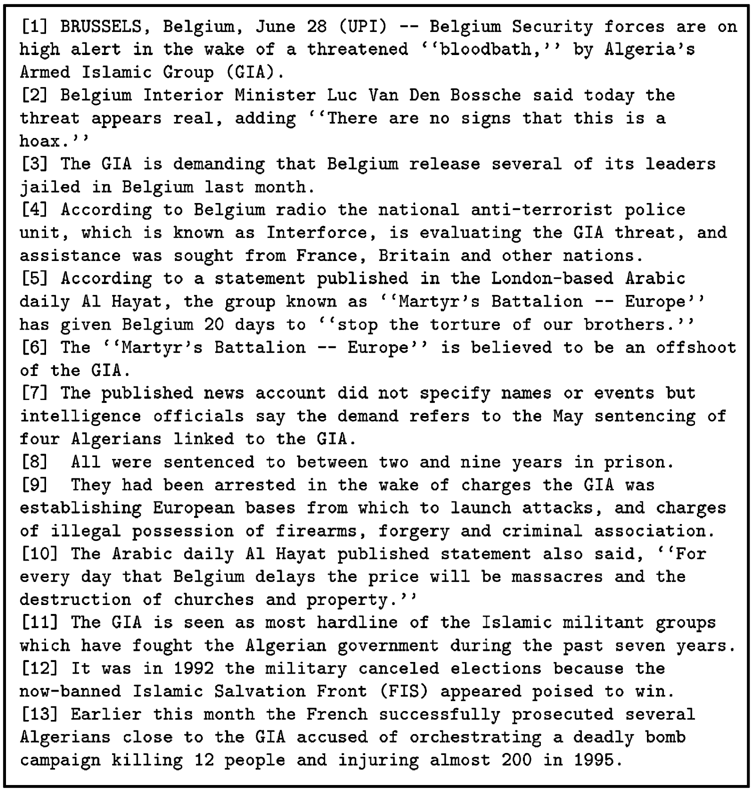
Here we present an example summary produced from 2 news documents: A1 (shown in Fig. 3) and A2 (shown in Fig. 4).

Using the algorithm described above we extracted summaries of various compression rates. A 10% summary of cluster A is shown in Fig. 5.



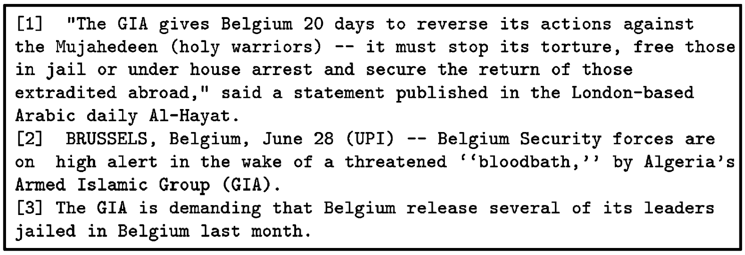
```
[1]   CAIRO, June 27 (AFP) - Hardline militants of Algeria's Armed
Islamic Group (GIA) threatened Sunday to create a "bloodbath" in
Belgium if the authorities there do not release several of its
leaders jailed last month.
[2]   "The GIA gives Belgium 20 days to reverse its actions against
the Mujahedeen (holy warriors) -- it must stop its torture, free
those in jail or under house arrest and secure the return of those
extradited abroad," said a statement published in the London-based
Arabic daily Al-Hayat.
[3]   "If these demands go unmet, the GIA vows to create a bloodbath
for the country and its inhabitants," said the statement dated
Friday and signed by "the emir of the martyrs' battalion in Europe,
Abu Hamza al-Afghani.
[4]   "For every day that Belgium delays, the price will be massacres
and the destruction of churches and property," the statement
warned.
[5]   A Belgian foreign ministry spokesman said on Sunday that the
country's relevant services were aware of the GIA's threat, but gave
no further details.
[6]   A Belgian court last month jailed several suspected GIA members
arrested in a police operation against Algerian militants in March.
[7]   The court jailed Frenchman Farid Melouk for nine years and
Algerian Mohamed Badache for five years on charges including
attempted murder and possession of explosives and firearms.
[8]   The court also sentenced Ait Sassi Jallal, a Moroccan currently
on the run, to five years and five other defendants to suspended
sentences.
[9]   Melouk is also wanted in Paris where he was sentenced to seven
years in absentia in February for involvement in a GIA support
network based around the French city of Lyon which is suspected of
involvement in a series of bloody attacks in France in 1995.
[10]   The GIA is the most hardline of the Islamic militant groups
which have fought the Algerian authorities since 1992 when the
military stepped in to cancel elections which the now-banned Islamic
Salvation Front (FIS) was poised to win.
[11]   The FIS's armed wing announced earlier this month that it would
lay down its arms altogether after observing a unilateral truce
since October 1, 1997.
[12]   Leaders of another armed group, the Salafist Group for Preaching
and Combat, are reported to be considering following suit.
```

Fig. 3. Input document A1.



[1] BRUSSELS, Belgium, June 28 (UPI) -- Belgium Security forces are on high alert in the wake of a threatened ‘‘bloodbath,’’ by Algeria’s Armed Islamic Group (GIA).
[2] Belgium Interior Minister Luc Van Den Bossche said today the threat appears real, adding ‘‘There are no signs that this is a hoax.’’
[3] The GIA is demanding that Belgium release several of its leaders jailed in Belgium last month.
[4] According to Belgium radio the national anti-terrorist police unit, which is known as Interforce, is evaluating the GIA threat, and assistance was sought from France, Britain and other nations.
[5] According to a statement published in the London-based Arabic daily Al Hayat, the group known as ‘‘Martyr’s Battalion -- Europe’’ has given Belgium 20 days to ‘‘stop the torture of our brothers.’’
[6] The ‘‘Martyr’s Battalion -- Europe’’ is believed to be an offshoot of the GIA.
[7] The published news account did not specify names or events but intelligence officials say the demand refers to the May sentencing of four Algerians linked to the GIA.
[8] All were sentenced to between two and nine years in prison.
[9] They had been arrested in the wake of charges the GIA was establishing European bases from which to launch attacks, and charges of illegal possession of firearms, forgery and criminal association.
[10] The Arabic daily Al Hayat published statement also said, ‘‘For every day that Belgium delays the price will be massacres and the destruction of churches and property.’’
[11] The GIA is seen as most hardline of the Islamic militant groups which have fought the Algerian government during the past seven years.
[12] It was in 1992 the military canceled elections because the now-banned Islamic Salvation Front (FIS) appeared poised to win.
[13] Earlier this month the French successfully prosecuted several Algerians close to the GIA accused of orchestrating a deadly bomb campaign killing 12 people and injuring almost 200 in 1995.

Fig. 4. Input document A2.



[1] "The GIA gives Belgium 20 days to reverse its actions against the Mujahedeen (holy warriors) -- it must stop its torture, free those in jail or under house arrest and secure the return of those extradited abroad," said a statement published in the London-based Arabic daily Al-Hayat.
[2] BRUSSELS, Belgium, June 28 (UPI) -- Belgium Security forces are on high alert in the wake of a threatened ‘‘bloodbath,’’ by Algeria’s Armed Islamic Group (GIA).
[3] The GIA is demanding that Belgium release several of its leaders jailed in Belgium last month.

Fig. 5. Sample output summary.

3.8. Example score table

Table 3 shows one way how MEAD could rank sentences. It combines Position, Length, SimWithFirst (first-sentence overlap), and Centroid scores for each sentence for all documents in a cluster.

Table 3
Score table for cluster A

DID	SNO	Length	Position	SimWithFirst	Centroid	Score
A1	1	34	1.000000	1.000000	0.939058	2.939058
A1	2	49	0.707107	0.023176	0.898288	1.628571
A1	3	39	0.577350	0.033781	0.342458	0.953589
A1	4	22	0.500000	0.026229	0.425512	0.951741
A1	5	24	0.447214	0.026288	0.123643	0.597145
A1	6	20	0.408248	0.199493	0.457915	1.065656
A1	7	27	0.377964	0.122771	0.405956	0.906692
A1	8	23	0.353553	0.000242	0.173832	0.527628
A1	9	46	0.333333	0.000605	0.481800	0.815738
A1	10	39	0.316228	0.233644	1.000000	1.549872
A1	11	25	0.301511	0.079100	0.125394	0.506005
A1	12	19	0.288675	0.193060	0.110614	0.592349
A2	1	26	1.000000	1.000000	0.962489	2.962489
A2	2	24	0.707107	0.048180	0.477059	1.232346
A2	3	16	0.577350	0.098166	0.762541	1.438057
A2	4	30	0.500000	0.049614	0.545765	1.095379
A2	5	33	0.447214	0.125150	0.858023	1.430387
A2	6	13	0.408248	0.109683	0.181009	0.698940
A2	7	28	0.377964	0.000035	0.188283	0.566282
A2	8	11	0.353553	0.000103	0.099978	0.453634
A2	9	31	0.333333	0.082990	0.382250	0.798573
A2	10	28	0.316228	0.047719	0.784451	1.148398
A2	11	23	0.301511	0.022082	0.511821	0.835414
A2	12	20	0.288675	0.026089	0.507750	0.822514
A2	13	29	0.277350	0.000176	0.396122	0.673648

4. Techniques for evaluating summaries

Summarization evaluation methods can be divided into two categories: intrinsic and extrinsic (Mani & Maybury, 1999). Intrinsic evaluation measures the quality of summaries directly (e.g., by comparing them to ideal summaries). Extrinsic methods measure how well the summaries help in performing a particular task (e.g., classification). Extrinsic evaluation, also called task-based evaluation, has received more attention recently at the Document Understanding Conference (DUC, http://duc.nist.gov).

4.1. Single-document summaries

Two techniques commonly used to measure interjudge agreement and to evaluate extracts are (A) precision and recall, and (B) percent agreement. In both cases, an automatically generated summary is compared against an "ideal" summary. To construct the ideal summary, a group of human subjects are asked to extract sentences. Then, the sentences chosen by a majority of humans are included in the ideal summary. The precision and recall indicate the overlap between the ideal summary and the automatic summary.

We should note that (Jing, McKeown, Barzilay, & Elhadad, 1998) pointed out that the cut-off summary length can affect results significantly, and the assumption of a single ideal summary is problematic.

We will illustrate why these two methods are not satisfactory. Suppose we want to determine which of the two systems that selected summary sentences at a compression rate of 20% (Table 4) is better.

Using precision and recall indicates that the performance of Systems 1 and 2 is 50% and 0%, respectively. System 2 appears to have to worst possible performance, since precision and recall treat sentences S3 to S10 as equally bad. Using percent agreement, the performance is 80% and 60%, respectively. However, percent agreement is highly dependent on the compression rate.

4.2. Utility-based evaluation of both single and multiple document summaries

Instead of P&R or percent agreement, one can measure the coverage of the ideal summary's utility. In the example in Table 5, using both evaluation methods A and B, System 1 achieves 50%, whereas System 2 achieves 0%. If we look at relative utility, System 1 matches 18 out of 19 utility points in the ideal summary and System 2 gets 15 out of 19. In this case, the performance of system 2 is not as low as when using methods A and B.

We therefore propose to model both interjudge agreement and system evaluation as real-valued vector matching and not as boolean (methods A and B). By giving credit for "less than ideal" sentences and distinguishing the degree of importance between sentences, the utility-based scheme is a more natural model to evaluate summaries.

Table 4
Comparing systems without utility metrics

	Ideal	System 1	System 2
Sent1	+	+	–
Sent2	+	–	–
Sent3	–	+	+
Sent4	–	–	+
Sent5	–	–	–
Sent6	–	–	–
Sent7	–	–	–
Sent8	–	–	–
Sent9	–	–	–
Sent10	–	–	–

Table 5
Comparing systems with utility metrics

	Ideal	System 1	System 2
Sent1	10 (+)	+	–
Sent2	9 (+)	–	–
Sent3	8	+	+
Sent4	7	–	+

Table 6
Illustrative example

	Judge 1	Judge 2	Judge 3
Sent1	10	10	5
Sent2	8	9	8
Sent3	2	3	4
Sent4	5	6	9

Other researchers have also suggested improvements on the precision and recall measure for summarization. Jing et al. (1998) proposed to use fractional P/R. Goldstein, Kantrowitz, Mittal, and Carbonell (1999) used 11-point average precision.

4.2.1. Interjudge agreement (J)

Without loss of generality, suppose that three judges are asked to build extracts of a single article. [3] As an example, Table 6 shows the weights of the different sentences (note that no compression rate needs to be specified; from the data in the table, one can generate summaries at arbitrary compression rates).

The interjudge agreement measures to what extent each judge satisfies the utility of the other judges by picking the right sentences.

In the example, with a 50% summary, Judge 1 would pick sentences 1 and 2 because they have the maximum utility as far as he is concerned. Judge 2 would select the same two sentences, while Judge 3 would pick 2 and 4. [4] The maximum utilities for each judge are as follows: 18 ($=10+8$), 19, and 17.

How well does Judge 1's utility assignment satisfy Judge 2's utility need? Since they have both selected the same sentences, Judge 1 achieves 19/19 (1.00) of Judge 2's utility. However, Judge 1 only achieves 13/17 (0.765) of Judge 3's utility.

We can therefore represent the cross-judge utility agreement $J_{i,j}$ as an asymmetric matrix (e.g., the value of $J_{1,2}$ is 0.765 while the value of $J_{2,1}$ is 13/18 or 0.722). The values $J_{i,j}$ of the cross-judge utility matrix for $r = 50\%$ are shown in Table 7.

We can also compute the performance of each judge (J_i) against all other judges by averaging for each Judge i all values in the matrix $J_{i,j}$ where $i \neq j$. These numbers indicate that Judge 3 is the outlier.

Finally, the mean cross-judge agreement J is the average of J_i for $i = 1, \dots, 3$. In the example, $J = 0.841$.

J is like an upper bound on the performance of a summarizer (it can achieve a score higher than J only when it can do a better job than the judges).

4.2.2. Random performance (R)

We can also similarly define a lower bound on the summarizer performance.

[3] We concatenate all documents in a cluster in a chronological order.

[4] In case of ties, we arbitrarily pick the sentence that occurs earlier in the cluster.

Table 7
Cross-judge utility agreement (J)

	Judge 1	Judge 2	Judge 3	Overall
Judge 1	1.000	1.000	0.765	0.883
Judge 2	1.000	1.000	0.765	0.883
Judge 3	0.722	0.789	1.000	0.756

The random performance R is the average of all possible system outputs at a given compression rate, r. For example, with 4 sentences and a $r = 50\%$, the set of all possible system outputs is {12,13,14,23,24,34}. [5] For each of them, we can compute a system performance. For example, the system that selects sentences 1 and 4 (we label this system as {14}) performs at 15/18 (or 0.833) against Judge 1, at 16/19 against Judge 2 (or 0.842), and at 14/17 against Judge 3 (or 0.824). On average, the performance of {14} is the average of the three numbers, or 0.833.

We can compute the performance of *all* possible systems. The six numbers (in the order {12,13,14,23,24,34}) are 0.922, 0.627, 0.833, 0.631, 0.837, and 0.543. Their average becomes the random performance (R) of all possible systems; in this example, $R = 0.732$.

4.2.3. System performance (S)

The system performance S is one of the numbers described in the previous subsection. For {13}, the value of S is 0.627 (which is lower than random). For {14}, S is 0.833, which is between R and J. In the example, only two of the six possible sentence selections, {14} and {24} are between R and J. Three others, {13}, {23}, and {34} are below R, while {12} is better than J.

4.2.4. Normalized system performance (D)

To restrict system performance (mostly) between 0 and 1, we use a mapping between R and J in such a way that when $S = R$, the normalized system performance, D, is equal to 0 and when $S = J$, D becomes 1. The corresponding linear function [6] is: $D = (S - R)/(J - R)$.

Fig. 6 shows the mapping between system performance S on the left (a) and normalized system performance D on the right (b). A small part of the 0–1 segment is mapped to the entire 0–1 segment; therefore the difference between two systems, performing at e.g., 0.785 and 0.812 can be significant! [7]

Example: the normalized system performance for the {14} system then becomes $(0.833 - 0.732)/(0.841 - 0.732)$ or 0.927. Since the score is close to 1, the {14} system is almost as good as the interjudge agreement. The normalized system performance for the {24} system is similarly $(0.837 - 0.732)/(0.841 - 0.732)$ or 0.963. Of the two systems, {24} outperforms {14}.

[5] There are a total of $(n!)/(n(1 - r))!(r * n)!$ system outputs.

[6] The formula is valid when $J > R$ (that is, the judges agree among each other better than randomly).

[7] This normalization example describes an ideal situation; in cases where random agreement among judges is high, normalization is not possible.

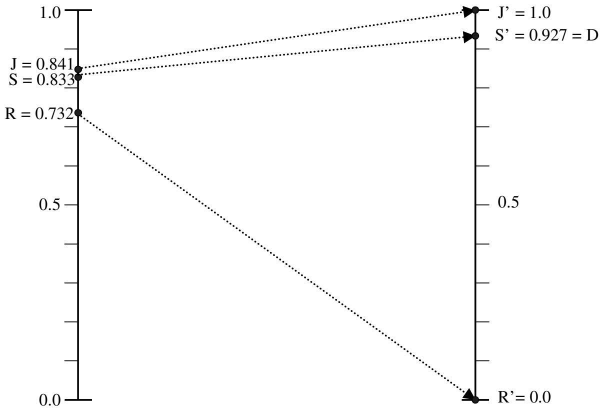
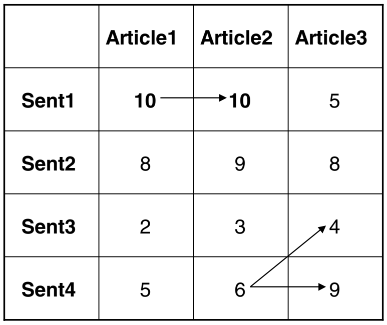


Fig. 6. Normalizing relative utility.

4.3. Using CSIS to evaluate multi-document summaries

To use CSIS in the evaluation, we introduce a new parameter, E, which tells us how much to penalize a system that includes redundant information. In the example from Fig. 7 (arrows indicate subsumption), a summarizer with $r = 20\%$ needs to pick 2 out of 12 sentences. Suppose that it picks 1/1 and 2/1 (in bold). If $E = 1$, it should get full credit of 20 utility points. If $E = 0$, it should get no credit for the second sentence as it is subsumed by the first sentence. By varying E between 0 and 1, the evaluation may favor or ignore subsumption.

When CSIS information is incorporated into relative utility, interjudge agreement, random performance, as well as system performance are affected. A heuristic is used to select sentences for a judge's extract, and sentences are penalized only when they are subsumed by other sentences in an extract. Since subsumption and the related penalties were only used tangentially in the experiments in this paper, we will not go into detail describing the exact algorithms used.



	Article1	**Article2**	**Article3**
Sent1	**10**	**10**	5
Sent2	8	9	8
Sent3	2	3	4
Sent4	5	6	9

Fig. 7. Examples of sentence-based subsumption.

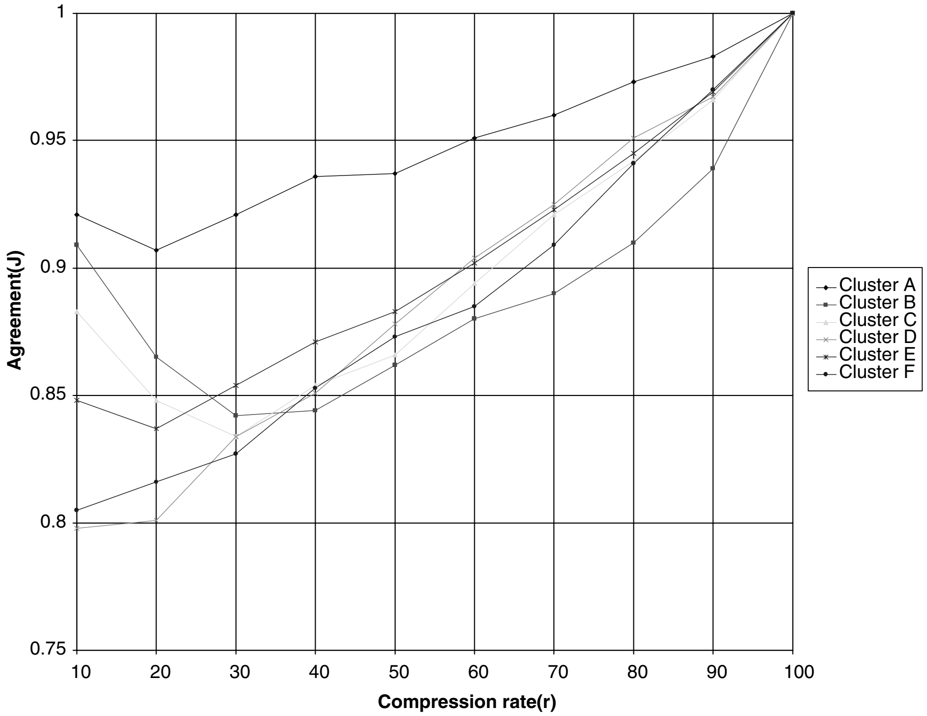


Fig. 8. Interjudge agreement across various compression rates.

5. User studies and system evaluation

We ran two user experiments. First, six judges were each given six clusters and asked to ascribe an importance score from 0 to 10 to each sentence within a particular cluster. Next, five judges had to indicate for each sentence which other sentence(s) in the entire cluster, if any, it subsumes. [8]

For a comparative study of different evaluation schemes, as well as how human evaluations correlate with such automatic schemes, we refer the reader to Radev et al. (2003) and Radev and Tam (2003).

5.1. RU: interjudge agreement

Using the techniques described in Section 4.2.1, we computed the cross-judge agreement (J) for the six clusters for various R (Fig. 8). Overall, interjudge agreement was quite high. An interesting drop in interjudge agreement occurs for 20–30% summaries. The drop most likely results from the fact that 10% summaries are typically easier to produce because the few most important sentences in a cluster are easier to identify.

[8] We should note that both annotation tasks were quite time consuming and frustrating for the users who took anywhere from 6 to 10 h each to complete their part.

5.2. CSIS: interjudge agreement

In the second experiment, we asked users to indicate all cases when within a cluster, a sentence is subsumed by another. The judges' data on the first seven sentences of cluster A are shown in Table 8.

The "+ score" indicates the number of judges who agree on the most frequent subsumption. The "− score" indicates that the consensus was "no subsumption". We found relatively low interjudge agreement on the cases in which at least one judge indicated evidence of subsumption. Overall, out of 558 sentences, there was full agreement (five judges) on 292 sentences (Table 9). Unfortunately, in 291 of these 292 sentences the agreement was "no subsumption". When the bar of agreement was lowered to four judges, 23 out of 406 agreements are on sentences with subsumption. Overall, out of 80 sentences with subsumption, only 24 had an agreement of four or more judges. However, in 54 cases at least three judges agreed on the presence of a particular instance of subsumption.

In conclusion, we found very high interjudge agreement in the first experiment and moderately low agreement in the second experiment. We concede that the time necessary to do a proper job at the second task is partly to blame.

5.3. Evaluation of MEAD

We used a number of SCORE functions to assign various weights to the features describe above, namely, Position, Centroid, and Overlap with First Sentence. Then we extracted

Table 8
Judges' indication for subsumption for the first seven sentences in cluster A

Sentence	Judge 1	Judge 2	Judge 3	Judge 4	Judge 5	+ Score	− Score
A1-1	–	A2-1	A2-1	–	A2-1	3	–
A1-2	A2-5	A2-5	–	–	A2-5	3	–
A1-3	–	–	–	–	A2-10	–	4
A1-4	A2-10	A2-10	A2-10	–	A2-10	4	–
A1-5	–	A2-1	–	A2-2	A2-4	–	2
A1-6	–	–	–	–	A2-7	4	–
A1-7	–	–	–	–	A2-8	–	4

Table 9
Interjudge CSIS agreement

# Judges agreeing	Cluster A		Cluster B		Cluster C		Cluster D		Cluster E		Cluster F	
	+	−	+	−	+	−	+	−	+	−	+	−
5	0	7	0	24	0	45	0	88	1	73	0	61
4	1	6	3	6	1	10	9	37	8	35	0	11
3	3	6	4	5	4	4	28	20	5	23	3	7
2	1	1	2	1	1	0	7	0	7	0	1	0

summaries using these SCORE functions and compared them with Lead-based summaries. We find that including these features on top of Lead generates more human-like extracts. In large clusters in particular, the benefit of adding the Centroid feature is most demonstrable, see Table 10. Here we used equal weights for each feature. The "–" indicates that no normalized relative utility can be computed as the average judge score is the same as the random score.

Averaged over all six clusters, Lead appears to perform quite well. This is due to the fact that a few of the clusters are rather small. Table 11 shows the average relative utility matrix for the six clusters.

We have experimented with unequal weights for different features. It appears that for our experiment, the best SCORE function is: $\text{SCORE}(s_i) = C_i + 2 * P_i + F_i$.

The summaries produced by this function consistently outperforms Lead for compression rates 20–30%. Table 12 contains the normalized relative utility for all clusters using both Lead and our best score function.

Table 10
Performance of various summarizers for a large cluster D (L = LEAD, C = CENTROID, P = POSITION, O = OVERLAP)

Compression (%)			L		C		C + P		C + P + O	
	J	R	S	D	S	D	S	D	S	D
10	0.80	0.62	0.86	1.34	0.70	0.46	0.86	1.33	0.84	1.25
20	0.80	0.66	0.82	1.11	0.77	0.79	0.82	1.12	0.83	1.23
30	0.83	0.70	0.82	0.90	0.76	0.46	0.84	1.05	0.84	1.03
40	0.85	0.73	0.84	0.90	0.79	0.47	0.85	1.01	0.85	1.03
50	0.88	0.77	0.87	0.94	0.82	0.41	0.86	0.80	0.87	0.94
60	0.90	0.81	0.88	0.76	0.85	0.44	0.87	0.66	0.89	0.84
70	0.93	0.85	0.91	0.74	0.89	0.45	0.91	0.80	0.91	0.79
80	0.94	0.90	0.94	1.10	0.92	0.65	0.94	1.11	0.94	1.12
90	0.94	0.94	0.97	–	0.97	–	0.96	–	0.97	–

Table 11
Performance of various SCORE summarizers for all clusters

Compression (%)			L		C		C + P		C + P + O	
	J	R	S	D	S	D	S	D	S	D
10	0.86	0.60	0.89	1.10	0.77	0.65	0.88	1.07	0.87	1.06
20	0.86	0.64	0.84	0.95	0.79	0.73	0.85	0.98	0.84	0.95
30	0.86	0.69	0.85	0.91	0.80	0.63	0.84	0.90	0.84	0.90
40	0.88	0.72	0.87	0.93	0.82	0.63	0.86	0.86	0.86	0.86
50	0.89	0.76	0.90	1.01	0.85	0.64	0.87	0.84	0.87	0.81
60	0.91	0.80	0.92	1.04	0.86	0.60	0.88	0.76	0.89	0.79
70	0.93	0.84	0.93	0.99	0.89	0.56	0.91	0.76	0.91	0.76
80	0.94	0.88	0.95	1.28	0.92	0.78	0.94	1.15	0.94	1.23
90	0.95	0.94	0.96	1.26	0.96	1.08	0.97	1.44	0.97	1.14

Table 12
Normalized relative utility

Compression (%)	10	20	30	40	50	60	70	80	90
Lead	1.10	0.95	0.91	0.93	1.01	1.04	0.99	1.28	1.26
C1P2F1	1.08	1.02	0.95	0.90	0.91	0.80	0.90	1.24	1.55

5.4. Discussion

It may seem that utility-based evaluation requires too much effort and is prone to low interjudge agreement. We believe that our results show that interjudge agreement is quite high. As far as the amount of effort required, we believe that the larger effort on the part of the judges is more or less compensated with the ability to evaluate summaries off-line and at variable compression rates. Alternative evaluations don't make such evaluations possible. We should concede that a utility-based approach is probably not feasible for query-based summaries as these are typically done only on-line.

We discussed the possibility of a sentence contributing negatively to the utility of another sentence due to redundancy. Moreover, we should point out that sentences can also reinforce one another positively. For example, if a sentence mentioning a new entity is included in a summary, one might also want to include a sentence that puts the entity in the context of the rest of the article or cluster.

6. Contributions and future work

We presented a new multi-document summarizer, MEAD. It summarizes clusters of news articles automatically grouped by a topic detection system. MEAD uses information from the centroids of the clusters to select sentences that are most likely to be relevant to the cluster topic.

We used a new utility-based technique, RU, for the evaluation of MEAD and of summarizers in general. We found that MEAD produces summaries that are similar in quality to the ones produced by humans. We also compared MEAD's performance to an alternative method, multi-document lead, and showed how MEAD's sentence scoring weights can be modified to produce summaries significantly better than the alternatives.

We also looked at a property of multi-document clusters, namely cross-sentence information subsumption (which is related to the MMR metric proposed in Carbonell & Goldstein (1998)), and showed how it can be used in evaluating multi-document summaries.

All our findings are backed by the analysis of two experiments that we performed with human subjects. We found that the interjudge agreement on relative utility is very high while the agreement on cross-sentence subsumption is moderately low, although promising. With the limited data set however, we could not claim any statistical significance on our findings.

In the future, we would like to test our multi-document summarizer on a larger corpus and improve the summarization algorithm. More experiments using CSIS would be done to validate our new evaluation scheme. We would also like to explore how the techniques we proposed here can be used for multilingual multi-document summarization.

Acknowledgements

We would like to thank Inderjeet Mani, Wlodek Zadrozny, Rie Kubota Ando, Joyce Chai, and Nanda Kambhatla for their valuable feedback. We would also like to thank Carl Sable, Min-Yen Kan, Dave Evans, Adam Budzikowski, Barry Schiffman, and Veronika Horvath for their help with the evaluation.

References

Allan, J., Carbonell, J., Doddington, G., Yamron, J., & Yang, Y. (1998a). In *Proceedings of the broadcast news understanding and transcription workshop*. In *Topic detection and tracking pilot study: final report*.

Allan, J., Papka, R., & Lavrenko, V. (1998b). On-line new event detection and tracking. In *Proceedings of the 21st annual international ACM 57-GIR conference on research and development in information retrieval* (pp. 37–45). Melbourne, Australia.

Aone, C., Okurowski, M. E., Gorlinsky, J., & Larsen, B. (1997). A scalable summarization system using robust NLP. In *Proceedings of the workshop on intelligent scalable text summarization at the 35th meeting of the association for computational linguistics, and the 8th conference of the European chapter of the assocation for computational linguistics* (pp. 66–73).

Carbonell, J. G., & Goldstein, J. (1998). The use of MMR, diversity-based reranking for reordering documents and producing summaries. In A. Mof-fat, & J. Zobel (Eds.), *Proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval* (pp. 335–336). Melbourne, Australia.

Goldstein, J., Kantrowitz, M., Mittal, V. O., & Carbonell, J. G. (1999). Summarizing text documents: sentence selection and evaluation metrics. In *Research and development in information retrieval* (pp. 121–128). Berkeley, California.

Jing, H., McKeown, K. R., Barzilay, R., & Elhadad, M. (1998). Summarization evaluation methods: experiments and analysis. In E. Hovy, D. R. Radev (Eds.), *Intelligent Text Summarization. Papers from the 1998 AAAI Spring Symposium. Technical Report SS-98-06* (pp. 60–68). The AAAI Press, Stanford, California, USA.

Mani, I., & Bloedorn, E. (2000). Summarizing similarities and differences among related documents. *Information Retrieval, 1*(1).

Mani, I.& Maybury, M. T. (Eds.). (1999). *Advances in Automatic Text Summarization*. Cambridge, MA: MIT Press.

McKeown, K. R., Klavans, J., Hatzivassiloglou, V., Barzilay, R., & Eskin, E. (1999). Towards multidocument summarization by reformulation: progress and prospects. In *Proceeding of the 16th national conference of the American association for artificial intelligence (AAAI-1999)* (pp. 453–460).

Radev, D. R., Hatzivassiloglou, V., & McKeown, K. R. (1999). A description of the CIDR system as used for TDT-2. In *DARPA broadcast news workshop*. Herndon, Virginia.

Radev, D. R., & McKeown, K. R. (1998). Generating natural language summaries from multiple on-line sources. *Computational Linguistics, 4*, 469–500.

Radev, D. R., & Tam, D. (2003). Single-document and multi-document summary evaluation via relative utility. In *Poster Session, Proceedings of the ACM CIKM conference*. New Orleans, LA.

Radev, D. R., Teufel, S., Saggion, H., Lam, W., Blitzer, J. H. Q., Çelebi, A., Liu, D., & Drabek, E. (2003). Evaluation challenges in large-scale multi-document summarization: the mead project. In *Proceedings of ACL 2003*. Sapporo, Japan.