IIIT Hyderabad at DUC 2007

Prasad Pingali, Rahul K and Vasudeva Varma
LTRC, IIIT, Hyderabad, India
pvvpr@iiit.ac.in, rahul_k@research.iiit.ac.in and vv@iiit.ac.in

Abstract

In this paper we report our performance at DUC 2007 summarization tasks. We participated both in the query-focused multi-document summarization main task and in a pilot update summary generation tasks. This year we used a term clustering approach to better estimate a sentence prior. We used only the sentence prior which is query independent, in the update summarization task and found that it's performance is comparable with the top performing systems. In the main task our system ranked 1 in ROUGE-2, ROUGE-SU4 and ROUGE-BE scores as well as in pyramid scores.

1 Introduction

The query focused summarization track of DUC[1] is designed to take a step closer to the true "information" retrieval rather than "document" retrieval. Traditionally, the notion of information retrieval was limited to locate documents that might contain the relevant information, and it is left to the user to extract any useful information from a ranked list of documents. This leaves the user with a relatively large amount of text to manually process and consume the relevant pieces of text from within these documents. There is an urgent need for tools that would reduce the amount of text one might have to read in order to obtain the desired information. The query focused summarization track at DUC aims at doing exactly that for a special class of information seeking behavior, where an information need is posed using a set of questions. The motivation behind having a query focused summarization application is that people usually have questions and they need answers, as opposed to a set of documents as output. In the rest of the paper we discuss our participation in DUC 2007 in both main and update summarization tasks.

[1] Document Understanding Conference, http://duc.nist.gov

2 Problem Definition

DUC 2007 has conducted summarization evaluation in one main task and one pilot task.

The main task problem is defined as to synthesize from a small set of 25-50 documents $D = \{d_1, d_2, d_3...d_i\}$ that are related to a given topic or query $Q = \{q_1, q_2, q_3, ...q_k\}$, a brief, well-organized, fluent answer to a need for information given, that cannot be met by just stating a name, date, quantity, etc.

For the pilot task, an update summarization task was considered. An update summary is a summary, that assumes that the user has already read previous documents related to a given topic, and the summary only provides new or update information.

The main task comprised of 45 topics which had to be summarized. The update summary task was evaluated on a subset of the main task topics. Ten topics were chosen from the 45 topics of the main task. The document collection of each of these 10 topics were divided into three subsets, A, B and C based on the time period of their publication. An initial summary has be generated for the document

set from A and update summaries for document sets B and C have to be generated assuming document sets A and A, B were already read by the user.

3 Our Algorithm

We have addressed both the main and pilot summarization tasks using the same summarization algorithm. We use a sentence extraction based approach, where we extract sentences verbatim from the given set of documents and concatenate them as a summary. Hence, a sentence boundary identification program is executed on the document set D to obtain a set of sentences, $S = \{s_1, s_2, s_3...s_m\}$.

Our summarization algorithm can be outlined as following.

1. Identify sentence boundaries from the given set of documents.
2. Reduce Sentences.
3. Score and rank the sentences.
4. Pick the top ranking sentences and check for redundancy with previously selected sentences.
5. Concatenate the sentences in the order found in the source documents to generate a summary of the given length.
6. Post-process the summary to de-reference any entities.

3.1 Sentence Reduction

We have manually identified a set of patterns in sentences which may not be much informative and are usually added to provide some meta-information to the actual information being discussed. For example, in a sentence like *" President Clinton, however, is seeking a major increase in spending for national missile defense"*, the word *however* is usually not adding much information and therefore can be dropped without losing much information. We have manually hand-crafted about a hundred such patterns which can help reduce one or few words from the original sentences without losing much information. Each sentence from the set S is passed through these set of rules which may result in reducing the length of the given sentences. While this technique helps in compressing the input source, in some cases such rules may also affect readability. Clearly, the focus of our algorithm has been to pump in as much information as possible into the summary while not worrying too much about readability.

3.2 Sentence Scoring

The score of each sentence s is computed as

$$Score(s) = \alpha \cdot QIScore(s) + (1 - \alpha) \cdot QFocus(s, Q) \quad (1)$$

where α is a weighting factor and is experimentally computed using previous years' summarization datasets. $QIScore(s)$ acts as a sentence prior score (or query independent score), while $QFocus(s, Q)$ gives the score of a sentence answering the given query Q (query dependent score).

3.2.1 Query-Independent Score (QIScore)

We compute a query-independent score of a sentence using a contrastive analysis of the given document set D with a randomly chosen document set. A set of random documents from various topics $\hat{D} = \{\hat{d_1}, \hat{d_2}, \hat{d_3}...\hat{d_j}\}$ are chosen. Sentences extracted from this document set $\hat{D}$ are $\hat{S} = \{\hat{s_1}, \hat{s_2}, \hat{s_3}...\hat{s_n}\}$. Words belonging to D and $\hat{D}$ are clustered using their distribution in both the document sets as described in (Baker and McCallum, 1998). If a Term's probability distribution in both the document sets D and $\hat{D}$ is very similar to another term, such terms are clustered together. Similarity of the probability distributions is computed using the KL-Divergence between the two term distributions. Clustering of words helps achieve a better estimate of a prior score of a sentence, since sentences are sparse and clustering of features provides more information based on topically similar words.

Since sentences are typically short in length, we make use of feature clustering along with naïve bayes model of term distributions to learn a better model. Moreover, this model was shown to work very well when the size of training data and number of features are very small (Baker and McCallum, 1998).

After building the term clusters, score for each sentence s from S is computed using all the words occurring in the sentence as

$$QIScore(s) = \frac{P(D)\prod_{t=1}^{|s|} P(w_t|D)}{P(D)\prod_{t=1}^{|s|} P(w_t|D) + P(\hat{D})\prod_{t=1}^{|s|} P(w_t|\hat{D})} \quad (2)$$

For the update task, $\hat{D}$ was chosen to be the set of documents which the user has already read instead of any random documents. Therefore, while generating the update summaries for document clusters B and C, $\hat{D}$ is A and $A + B$ respectively.

3.2.2 Query-Dependent Score (QFocus)

We compute the query dependent score of a given sentence using co-occurrence statistics of terms in the given document set D. The joint probability distribution of every term co-occurring with every other term in a fixed window of length k words is estimated from the given document set. The resulting distribution would indicate features that are topically similar. The intuition behind such a computation is that topically similar terms tend to co-occur together more frequently. We used a similar approach in DUC 2005 (Jagarlamudi et al., 2005) and DUC 2006 (Jagarlamudi et al., 2006) previously, however, we did not treat all co-occurring terms occurring at different distances in a window as equally related in our previous approaches. Once such a distribution is estimated, all the words co-occurring with a word occurring in a sentence and with a joint probability above a threshold are included as part of the sentence. In a way we add more features in order to better estimate the probability of a sentence emitting a query, accounting for the sparseness in a given sentence. Therefore the query focused score of a sentence is computed as,

$$QFocus(s) = \prod_{w_j \in s} P(w_j|s) \sum_{w_i \in vocab} P(w_i|w_j, D).P(w_j|D) \quad (3)$$

3.3 Entity Deferencing

Once sentences are scored and top sentences are picked after eliminating redundancy, we de-reference repeating entities in the summaries. Repeated mention of an entity may not be reader friendly and hence needs to be de-referenced. This year, we attempted de-referencing of person names and organization names. The de-referencing task is achieved in the following manner.

1. Identify potential named entities (person names and organization names in our case)

2. Search the document set D for acronymized version of organization names and partial person names (i.e. first name or last name).

3. Replace repeated occurrences of a named entity with it's shorter name.

This technique resulted in very good readability and compression of the summaries. For example, an excerpt from the summary for topic 'D0701A' is shown here.

The Southern Poverty Law Center, *which was founded in the 1970s ... against the* **Ku Klux Klan** *and other white supremacist groups. A recent report from the* **SPLC** *... The* **SPLC** *previously recorded ... Triggs called* **Morris Dees**, *... the first municipality to designate the* **Klan** *a terrorist group. ... attorney* **Dees** *the ....*

The underlined items 'SPLC', 'Klan' and 'Dees' were automatic replacements by the system in place of their complete name occurrences. It can be noted that we do not replace the first occurrence of any entity.

4 Evaluation and Discussion

A set of 45 topics along with clusters of 25-50 documents relevant to each topic were provided for the main task. Using these inputs, systems were expected to generate a summary of 250 words. Similarly, a set of 10 topics along with three document clusters per topic were provided to generate an update summary. The number of documents to be summarized in an update cluster is about 10. We restricted the length of summary to 250 words (whitespace-delimited tokens) and 100 words for main and update tasks respectively. Summaries over the size limit were truncated.

The documents from which summary was to be generated were news articles and reports chosen from ACQUAINT corpus. We have observed that this year's corpus had many noisy terms, which is why some of our summaries had extraneous words in them. We report the official scores using ROUGE (Lin and Hovy, 2003) evaluation framework and the pyramid scores. In table 1 we show the average ROUGE scores over 45 topics in the main task, and in table 3 we show the average ROUGE scores in update task. In both these tables, the scores are arranged in the descending order of ROUGE-2 and the

rank of a system-ID is shown in parantheses along with it's score. The last column in table 1 and table 3 show the average content responsiveness from the manual evaluation. As noted in (Vanderwende et al., 2006) in DUC 2006 we observe this year that the content responsiveness does not correlate with the ROUGE and pyramid scores. Table 2 shows the average pyramid scores of various systems that participated in the pyramid evaluation. Our system performance is shown in bold in all these tables.

System ID	ROUGE-2	ROUGE-SU4	ROUGE-BE	C Resp.
Human Mean	0.14099	0.19158	0.08856	4.712
15	**0.12448 (1)**	**0.17711 (1)**	**0.06632 (1)**	**2.844 (13)**
29	0.12028 (2)	0.17074 (3)	0.06458 (3)	3.000 (5)
4	0.11887 (3)	0.16999 (4)	0.06388 (4)	3.400 (1)
24	0.11793 (4)	0.17593 (2)	0.06577 (2)	3.000 (5)
13	0.11172 (5)	0.16446 (5)	0.06230 (5)	2.933 (8)

Table 1: Main Task, ROUGE and Content Responsiveness scores

System ID	Pyramid Score
15	**0.348700**
29	0.340030
13	0.327952
24	0.327443
23	0.306265
30	0.277130
14	0.267183
9	0.258843
2	0.252752
17	0.251513
5	0.245783
6	0.154078
1	0.138748

Table 2: Average Pyramid scores

System ID	ROUGE-2	ROUGE-SU4	ROUGE-BE	C Resp.
Human Mean	0.12642	0.16169	0.09027	3.975
40	0.11189 (1)	0.14306 (1)	0.07219 (1)	2.967 (1)
55	**0.09851 (2)**	**0.13509 (3)**	**0.05223 (7)**	**2.700 (4)**
45	0.09622 (3)	0.13245 (4)	0.05542 (3)	2.533 (9)
47	0.09387 (4)	0.13052 (5)	0.05458 (4)	2.633 (7)
44	0.09370 (5)	0.13607 (2)	0.05544 (2)	2.600 (8)

Table 3: Pilot Update Task, Average ROUGE and Content Responsiveness

In the update task this year, we find that our system could have performed better if both query independent and query focused scoring was included. However, it can be observed that despite not using any query while generating the update summary, our summaries are comparable with the top performing summarizers.

5 Conclusion

In this paper we reported our experiments in DUC 2007 main and update tasks. We participated in pyramid evaluations this year and we find that the pyramid scores of our summaries correlate well with the ROUGE evaluation. However, we found that the manual evaluation did not correlate well with the ROUGE and pyramid scores. This year we experimented with sentence reduction and entity dereferencing as part of our summarization algorithm. We also experimented with a term clustering technique to generate a query-independent score or a prior of a sentence. We found that this technique is able to achieve a comparable performance with other top performing summarizers in the update summarization task.

References

L. Douglas Baker and Andrew Kachites McCallum. 1998. Distributional Clustering of Words for Text Classification. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 96–103, New York, NY, USA. ACM Press.

Jagadeesh Jagarlamudi, Prasad Pingali, and Vasudeva Varma. 2005. A Relevance-Based Language Modeling Approach to DUC 2005. In *Document Understanding Conference, October 2005 at Annual meeting of HLT/EMNLP*.

Jagadeesh Jagarlamudi, Prasad Pingali, and Vasudeva Varma. 2006. Query Independent Sentence Scoring approach to DUC 2006. In *Document Understanding Conference, June 2006 at Annual meeting of HLT/NAACL*.

Chin-Yew Lin and E.H. Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of 2003 Language Technology Conference (HLT-NAACL 2003), Edmonton, Canada*.

Lucy Vanderwende, Hisami Suzuki, and Chris Brockett. 2006. Microsoft Research at DUC2006: Task-Focused Summarization with Sentence Simplification and Lexical Expansion. In *Document Understanding Conference 2006, at HLT/NAACL 2006*. Association for Computational Linguistics.