Back to Basics: CLASSY 2006

John M. Conroy
IDA/Center for Computing Sciences
conroy@super.org

Judith D. Schlesinger
IDA/Center for Computing Sciences
judith@super.org

Dianne P. O'Leary
University of Maryland
oleary@cs.umd.edu

Jade Goldstein
Department of Defense
jade44@gmail.com

Abstract

The IDA/CCS summarization system, CLASSY, underwent significant change for this year's DUC. Two changes made processing simpler and faster: 1) we eliminated the use of a POS (part of speech) tagger for sentence splitting and to assist sentence trimming, and 2) we simplified the scoring of sentences for inclusion in the summary by introducing a new "approximate oracle" score. An additional change introduced a modest amount of extra computation: we ordered sentences in the summary using a new Traveling Salesperson (TSP) formulation. These changes improved ROUGE scores on the DUC 2005 data from last year and gave strong performance in the DUC 2006 competition.

1 Introduction

The IDA/CCS summarization system, CLASSY, has performed very well in recent DUC and MSE evaluations. That not withstanding, CLASSY underwent a major overhaul between last year and this which, we believe, led to improved summarization.

The system had major changes in all of its major components:

1. Sentence splitting and trimming (shallow processing);
2. Sentence scoring for initial summary consideration; and
3. Sentence selection for redundancy removal and to improve flow.

For the most part, the changes made actually made processing simpler and, in some cases, faster.

In addition to these changes, the topic descriptions were also processed to generate query terms for the scoring algorithm.

2 Document Preparation

Prior to scoring sentences for selection, each document is split into individual sentences which are marked as 1) candidates for inclusion in a summary (1), 2) non-candidates that might provide useful terms (0),

or 3) non-candidates to be ignored (-1). Each document then is sentence split and processed on a sentence-by-sentence basis to trim "unnecessary" parts of the sentence before the documents are given to the sentence scorer. In parallel with this processing, the topic descriptions are analyzed and query terms are selected for the sentence scorer to use.

The following sections describe the document preparation processes which have been modified since the DUC 2005 submission. Additional details can be found in earlier DUC papers [8, 4].

2.1 Sentence Splitting

The sentence splitter we were using was a "by-product" of a commercial product for which we did not have source code[1] It proved ineffective in our system because we could not make changes in response to splitting errors that were identified.

We wrote a new JAVA-based sentence splitter which can be updated as needed. While errors continue to be made, we are now able to correct for many of them. In addition, during the sentence trimming process that follows sentence splitting, we have been able to correct for many errors that are beyond the capability of a simple sentence splitter. These errors include:

- erroneous splits on foreign words, especially names, that appear to be English abbreviations;
- erroneous splits on less commonly used abbreviations due to their omission from the sentence splitters abbreviation list as well as to "unexpected" use;
- erroneous splits due to missing or bad punctuation; and
- erroneous splits due to ellipsis at sentence end (our sentence splitter does not look far enough ahead).

These fixes may either split a sentence that should have been and wasn't or combine two (or more) "sentences" that should not have been split.

We believe that with this post-sentence-splitting processing, we have some of the most accurate sentence splits possible. Unfortunately, errors do remain that would require full parsing (which we do not do) to detect.

2.2 Sentence Trimming

Prior to DUC 2006, we had been using a POS-tagger to tag words in a sentence. We then used these POS-tags to help identify individual words, phrases, and/or clauses to be trimmed. As with the sentence splitter, we experienced problems with this in our production system. In about 10% of the documents processed, the POS-tagger failed, i.e., no tagged file was returned.

This problem alone was enough to make us find another solution. In addition, the POS-tagger, while not slower than any comparable system, was a slow "cog" in our system. We also found that the errors made by the POS-tagger were leading to errors in the subsequent trimming that caused grammatical sentences to be made ungrammatical.

We undertook a major re-write of all trimming code to eliminate dependence on POS tags. At the onset, it was not clear that we could succeed, but our DUC 2006 submission demonstrates that we did.

[1] We choose not to identify the product since problems mentioned herein are not unique to this product.

The new algorithms rely on lists of "function" words, i.e., those words that play critical roles such as prepositions, conjunctions, determiners, etc., on lists of words that play a major role in a specific trim, i.e., adverbs, gerunds[2], and on punctuation. The word lists were created ad hoc for the code and can easily have omissions. They are simply extended when new words are "discovered".

Our sentence trimming tasks basically remain as they have been.

1. We remove extraneous words that appear in a sentence, including date lines, editor's comments, and so on.
2. We remove many adverbs and all conjunctions, including phrases such as "As a matter of fact," and "At this point," that occur at the start of a sentence.
3. We remove a small selections of words that occur in the middle of a sentence, such as ", however," and ", also," (not always requiring the commas).
4. For DUC 2006, we added the removal of ages such as ", 51," or ", aged 24,".
5. We remove gerund phrases (phrases starting with the -ing form of a word) from the start, middle, or end of a sentence when possible.
6. We remove relative clause attributives (clauses beginning with "who(m)", "which", "when", and "where") wherever possible.
7. We remove attributions, such as "police said", at the start or end of sentences when the text is not a quote.

Commas, periods, and sentence start are used in identifying most of these items to remove. There are only a few exceptions where these are not required.

It should be noted that we are very conservative with our trims, electing to miss something rather than risk creating an ungrammatical sentence due to bad trimming. Analysis of this trimming task, applied to the DUC 2005 data, showed an error rate of less than 3%, i.e., fewer than 3% of the trimmed sentences were made ungrammatical (or worse, semantically wrong). This compares to an error rate of approximately 25% for the sentences trimmed using the POS tagger.

2.3 Query Term Selection

Query terms are extracted from the topic description for each document cluster by excluding all words occurring in the function word lists mentioned in Section 2.2 along with a list of stop words consisting of generic words that may appear in topic descriptions like "organization", "background", and "explain".

Previously, CLASSY did not use stemming. To test the effects of stemming, we used the dictionary list for the Porter Stemmer (snowball.tartarus.org/algorithms/english/diffs.txt) to expand the query term list to include full words that correspond to the stemmed version of each query term. We also added some common endings (such as -s, -ed, -ing) to the query terms, to further expand the list from those terms which the "reverse" Porter stemming list provided.

When evaluated on DUC 2005 summaries, the ROUGE scores slightly increased, therefore we incorporated the expanded query term algorithm into our DUC 2006 runs.

[2]Actually, we use a list of "non-gerunds" rather than trying to list the gerund form of all verbs.

3 Scoring Sentences

Until this year, we've used an HMM to do the initial scoring and selection of sentences. We were motivated by SumBasic [11] to take a fresh look at scoring sentences. SumBasic uses the high frequency content words that occur in the relevant documents to yield a simple and powerful summarization method. SumBasic produced extract summaries which performed nearly as well as the best machine systems for generic 100 word summaries, as evaluated in DUC 2003 and 2004, as well as the Multi-lingual Summarization Evaluation in 2005.

Instead of using term frequencies of the corpus to infer highly likely terms in human summaries, we propose to directly model the *set* of terms (vocabulary) that is likely to occur in a sample of human summaries. We model human variation in summary generation with a unigram bag-of-words model on the terms. In particular, let $P(t|\tau)$ be the probability that a human will select term t in a summary given a topic τ. We define the *score* for a sentence x to be

$$\omega(x) = \frac{1}{|x|} \sum_{t \in T} x(t) P(t|\tau)$$

where $|x|$ is the number of distinct terms sentence x contains, T is the universal set of all terms used in the topic τ and $x(t) = 1$ if the sentence x contains the term t and 0 otherwise. We produce a computable *approximate oracle score* [7] to substitute for this score.

If we were given a set of human abstracts for a topic τ, we could readily compute the maximum-likelihood estimate of $P(t|\tau)$. Suppose we are given h sample summaries generated independently. Let $c_{it}(\tau) = 1$ if the i-th summary contains the term t and 0 otherwise. Then the maximum-likelihood estimate of $P(t\tau)$ is given by

$$\hat{P}(t|\tau) = \frac{1}{h} \sum_{i=1}^{h} c_{it}(\tau).$$

We define $\hat{\omega}$ by replacing P with $\hat{P}$ in the definition of ω. Thus, $\hat{\omega}$ is the maximum-likelihood estimate for ω, given a set of h human summaries. This *oracle score*, which allows us to compute the expected number of abstract terms in a sentence, was shown to achieve ROUGE-2 performance exceeding that of the humans on DUC 2005 data [7]. We use this oracle score as a guide to develop approximate scores when the human abstracts are not known.

3.1 An Estimate of $P(t|\tau)$

To estimate $P(t|\tau)$, we view both the query terms and the signature terms as "samples" from idealized human summaries. Query terms are extracted from the topic description (see Section 2.3) while signature terms are extracted from the set of documents. Loosely, a signature term is a term which occurs significantly more than expected (see [10, 6]). Both query terms and signature terms are likely to appear in a human summary. As such, we expect that the set of these terms may approximate the underlying set of human summary terms. Given a collection of query terms and signature terms, we can readily estimate our target objective, $P(t|\tau)$ by the following:

$$P_{qs}(t|\tau) = \frac{1}{2} q_t(\tau) + \frac{1}{2} s_t(\tau)$$

where $s_t(\tau)$=1 if t is a signature term for topic τ and 0 otherwise, and $q_t(\tau) = 1$ if t is a query term for topic τ and 0 otherwise.

More sophisticated weighting of the query and signature have been considered. We discuss two approaches below in Section 3.2. section which explains the change to this.

Similarly, we define the query term and signature term approximation of the oracle score of a sentence's expected number of human abstract terms as

$$\omega_{qs}(x) = \frac{1}{|x|} \sum_{t \in T} x(t) P_{qs}(t|\tau)$$

where $|x|$ is the number of distinct terms sentence x contains, T is the universal set of all terms and $x(t) = 1$ if the sentence x contains the term t and 0 otherwise.

3.2 Improving the Estimate of $P(t|\tau)$

We considered two approaches to improve the estimates for the probability that a term would be included in a human summary. The first approach was to include entities as a third set of terms used to estimate $P(t|\tau)$. We used BBN IndentiFinder to extract PERSONS, LOCATIONS, and ORGANIZATIONS. Two methods were used to fold entities into the estimate for $P(t|\tau)$. The first approach was to simply give them some proportional weight relative to their occurrence based on a subset of the document sets. This approach did not yield an improvement in ROUGE scores. We conjectured that entities were only relevant when the topic description asked for information regarding an entity. To this end, we conditionally included entities based upon the topic description. Unfortunately, this approach also did not yield higher ROUGE scores. We found that when entities were relevant to a document set, they were often covered by the signature terms. Thus, the added information of entities was redundant.

The second attempt to improve the estimates was to return to using just query terms and signature terms and to use them in a pseudo-relevance feedback method (see, for example, [1] or [9]) to reestimate $P(t|\tau)$. In particular, we first compute the score ω_{qs} for each sentence of the document cluster. Then we select the top k scoring sentences. If the term-sentence incidence matrix is given by A and the top k scoring sentences correspond to columns $j_1, j_2, ..., j_k$, we then compute $\rho(t)$, the expectation that a term t is used in the extract of these k sentences. That is,

$$\rho(t) = \frac{1}{k} \sum_{i=1}^{k} A(t, j_i).$$

Our updated estimate for $P(t|\tau)$ is simply the average of the $P_{qs}(t|\tau)$ and $\rho(t)$,

$$P_{qs\rho}(t|\tau) = \frac{1}{2} P(t|\tau) + \frac{1}{2} \rho(t).$$

The updated score for a sentence x is given by

$$\omega_{qs\rho}(x) = \frac{1}{|x|} \sum_{t \in T} x(t) P_{qs\rho}(t|\tau)$$

3.3 Reducing Redundancy of the Selected Sentences

We form the summary by taking the top scoring sentences from those sentences with at least 8 distinct terms. (We found the length of 8 empirically using the DUC 2005 data.) To minimize redundancy, we first select enough sentences to give a summary of length 500, i.e., twice the target length. A pivoted-QR is then used to select the subset of these top scoring sentences [3].

3.4 Ordering of Sentences for the Summary

The lead sentence of the summary is the highest scoring sentence as calculated by our score $\omega_{qs\rho}$. Given this lead sentence and a set of additional sentences selected by the pivoted-QR to include in the summary, we determined the order for the additional sentences using a Traveling Salesperson (TSP) formulation. We defined a *distance* between each pair of sentences and then determined an ordering that minimized the sum of the distances between adjacent sentences. Since the summaries were limited to 250 words, it is not too difficult to solve the TSP. For example, there are only 3,628,800 ways to order 10 sentences plus a lead sentence, so exhaustive search is feasible. For our purposes, though, we chose the best of a large sample of orderings, some random and some determined by single-swap changes on a previous candidate ordering. 50,000 samples seemed to be enough to consistently generate a minimum distance ordering.

Althaus, Karamanis, and Koller [2] also use the TSP formulation to determine sentence ordering. Their distance function is based on a combination of *coherence* and *salience*. They also survey other distance functions from the literature.

The choice of distance function is, of course, critical to making this method succeed. Guided by our term-based algorithm for choosing the sentences, we chose to compute it as follows:

1. For each pair of sentences (j, k), we computed the *similarity measure* b_{jk} to be the number of terms they have in common.

2. If the two sentences come from the same document, we multiplied b_{jk} by 1.6, to increase their similarity measure.

3. We then normalized these measures so the similarity between each sentence and itself is 1. We accomplish this by computing
$$c_{jk} = \frac{b_{jk}}{\sqrt{b_{jj} b_{kk}}}.$$

4. The distance between sentence j and sentence k is then defined to be $-c_{jk}$, or, if you prefer nonnegative distances, $1 - c_{jk}$.

4 Results

Based on the ROUGE scores and pyramid evaluation, shown below, we are pleased to see that the fairly dramatic changes we made to CLASSY did not diminish its performance, especially since most of our changes simplified computation, yielding a faster summarization system.

Our system was system 15. It faired well in terms of ROUGE scores. For ROUGE-1, while it finished fifth, it was within the 95% confidence interval of the second highest system. On ROUGE-2, system 15 placed second, and for ROUGE-SU4 it was very close in score to the second highest scoring system. For ROUGE-BE System, system 15 scored within the 95% confidence interval of the best scoring system. In the pyramid evaluation, system 15 ranked 4th; however, many systems (including ours) scored within the confidence interval of the top scoring system. Our system was slightly handicapped in this evaluation; due to a glitch in the DUC scoring software, the summaries for our system (and also system 8) were truncated to fewer than the 250 words specified in the rules, with one word truncated for every sentence in the summary, so our summaries evaluated by the pyramid had as few as 230 words.

After the competition, we developed an enhanced system for the MSE competition [5]. In this system, labeled 15E in the tables, we removed pseudo-relevance feedback, used stemming to form the term-sentence matrices, and modified our redundancy removal algorithm. On ROUGE-1, system 15E would have placed first, and its score is within the 95% confidence interval of one human. On both ROUGE-2 and ROUGE-SU4, it would have scored second, within the confidence interval for the top scorer and 2 humans.

Submission	Mean	95% CI Lower	95% CI Upper
C	0.47856	0.46005	0.49867
B	0.47284	0.45479	0.49179
D	0.46873	0.45129	0.48687
G	0.45909	0.44700	0.47187
J	0.45892	0.43983	0.47905
A	0.45816	0.44963	0.46821
I	0.45347	0.43629	0.46972
H	0.44834	0.43036	0.46385
E	0.44316	0.42297	0.46286
F	0.43578	0.41054	0.45968
15E	0.42282	0.41675	0.42845
24	0.41108	0.40488	0.41706
12	0.40488	0.39919	0.41051
23	0.40440	0.39820	0.40967
10	0.40369	0.39852	0.40874
15	0.40279	0.39649	0.40839
33	0.40206	0.39684	0.40754
8	0.40010	0.39370	0.40597
28	0.39922	0.39364	0.40460

Table 1: Average F score of ROUGE 1 Scores

Acknowledgement

The authors thank Michael Graham for the work he did in developing the sentence splitter that we are now using. His work is greatly appreciated by us all.

References

[1] James Allan. Relevance feedback with too much data. In *In Proc. of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, WA*, pages 337–343, 1995.

[2] Ernst Althaus, Nikiforos Karamanis, and Alexander Koller. Computing locally coherent discourses. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 399–406, July 2004.

[3] J.M. Conroy and D.P. O'Leary. "Text Summarization via Hidden Markov Models and Pivoted QR Matrix Decomposition". Technical report, University of Maryland, College Park, Maryland, March, 2001.

[4] J.M. Conroy, J.D. Schlesinger, J. Goldstein, and D.P. O'Leary. Left-brain right-brain multi-document summarization. In *DUC 04 Conference Proceedings*, 2004. `http://duc.nist.gov/`.

Submission	Mean	95% CI Lower	95% CI Upper
C	0.13260	0.11596	0.15197
D	0.12380	0.10751	0.14003
B	0.11788	0.10501	0.13351
G	0.11324	0.10195	0.12366
F	0.10893	0.09310	0.12780
H	0.10777	0.09833	0.11746
J	0.10717	0.09293	0.12460
I	0.10634	0.09632	0.11628
E	0.10365	0.08935	0.11926
A	0.10361	0.09260	0.11617
24	0.09558	0.09144	0.09977
15E	0.09539	0.09145	0.09922
15	0.09097	0.08671	0.09478
12	0.08987	0.08583	0.09385
8	0.08954	0.08540	0.09338

Table 2: Average F score of ROUGE 2 Scores

[5] John M. Conroy, Dianne P. O'Leary, and Judith D. Schlesinger. CLASSY Arabic and English multi-document summarization. In *Multi-Lingual Summarization Evaluation 2006*, 2006. `http://www.isi.edu/~cyl/MTSE2006/MSE2006/papers/index.html`.

[6] John M. Conroy, Judith D. Schlesinger, and Jade Goldstein. Three CLASSY ways to perform Arabic and English multi-document summarization. In *Multi-Lingual Summarization Evaluation*, 2005.

[7] John M. Conroy, Judith D. Schlesinger, and Dianne P. O'Leary. Topic-focused multi-document summarization using an approximate oracle score. In *Proceedings of the ACL'06/COLING'06*, 2006.

[8] John M. Conroy, Judith D. Schlesinger, and Jade Goldstein Stewart. CLASSY query-based multi-document summarization. In *DUC 05 Conference Proceedings*, 2005.

[9] K. L. Kwok, L. Grunfeld, and D. D. Lewis. Trec-3 ad-hoc, routing retrieval and thresholding experiments using pircs. In *Proc. of the Text REtrieval Conference, Gaithersburg, MD*, pages 247–256, 1995.

[10] Chin-Yew Lin and Eduard Hovy. The automated acquisition of topic signatures for text summarization. In *Proceedings of the 18th conference on Computational linguistics*, pages 495–501, Morristown, NJ, USA, 2000. Association for Computational Linguistics.

[11] Ani Nenkova and Lucy Vanderwende. The impact of frequency on summarization. In *MSR-TR-2005-101*, 2005.

Submission	Mean	95% CI Lower	95% CI Upper
C	0.18385	0.17012	0.19878
D	0.17814	0.16527	0.19094
B	0.17665	0.16356	0.19080
G	0.17121	0.16301	0.17952
J	0.16934	0.15716	0.18319
I	0.16843	0.15828	0.17851
A	0.16829	0.16042	0.17730
H	0.16665	0.15627	0.17668
E	0.16298	0.15012	0.17606
F	0.16043	0.14518	0.17771
24	0.15529	0.15126	0.15906
15E	0.15311	0.14934	0.15659
12	0.14755	0.14360	0.15142
15	0.14733	0.14373	0.15069
8	0.14607	0.14252	0.14943

Table 3: Average F score of ROUGE-SU4 Scores

Submission	Mean	95% CI Lower	95% CI Upper
C	0.09905	0.08125	0.11911
B	0.07847	0.06844	0.09050
D	0.07466	0.05807	0.09283
G	0.06756	0.05750	0.07974
F	0.06729	0.05480	0.08255
H	0.06625	0.06047	0.07252
J	0.06601	0.05666	0.07616
A	0.06600	0.05422	0.08011
E	0.06073	0.05201	0.06982
I	0.05767	0.04997	0.06561
24	0.05107	0.04766	0.05436
23	0.05049	0.04728	0.05370
15	0.04852	0.04558	0.05177
2	0.04797	0.04461	0.05127

Table 4: Average F score of ROUGE-BE Scores