Columbia University at MSE 2005

Advaith Siddharthan and **Dave Evans**
Department of Computer Science
Columbia University
New York
{as372, devans}@cs.columbia.edu

Abstract

We describe our participation in the Multilingual Summarization Evaluation 2005. We describe the Columbia summarizers that were used in our submission and discuss the evaluation, drawing conclusions about the performance of our summarizers, discussing the state of multilingual summarization in general and also listing issues that need consideration for future evaluations.

1 Introduction

The Multilingual Summarization Evaluation 2005 aimed to evaluate multi-document summarizers on document sets containing a mixture of English and machine translated Arabic news reports. This differs from previous multilingual summarization evaluation efforts, such as the one in the Document Understanding Conference 2004, where the document sets consisted of two different machine translations into English of Arabic news reports. We have fine tuned our summarizer for this new task; in this paper we describe our summarizer and our experience with the evaluation effort.

The Columbia summarizers used for this evaluation have all been described elsewhere; in this paper we restrict ourselves to overviewing them and citing the papers where full details can be found (§2). An important aspect of our submission this year is that we prepared a training corpus, which we used to identify the configurations of our summarizers that performed best on the Rouge SU4 metric. We discuss our training corpus and show how it proved useful, despite it being appreciably different from the corpus used in MSE'05 (§3).

We then summarize the performance of our system in §4, on both the manual evaluation on ten sets using the pyramid approach (Nenkova and Passonneau, 2004) and the automatic evaluation using Rouge (Lin and Hovy, 2003). We discuss particular issues arising from these evaluations that need to be considered for future evaluations of this nature (§5) and present our main conclusions in §6.

2 The Columbia summarizers

We use a sentence-clustering approach to multi-document summarization (similar to *MultiGen* (Barzilay, 2003)), where sentences in the input documents are clustered according to their similarity. Larger clusters represent information that is repeated more often across input documents; hence the size of a cluster is indicative of the importance of that information. We use *SimFinder* (Hatzivassiloglou et al., 1999) to perform sentence clustering.

A problem with this approach is that the clustering is not always accurate. Clusters can contain spurious sentences, and a cluster's size might then exaggerate its importance. Improving the quality of the clustering can thus be expected to improve the content of the summary. Our summarizers perform two operations (1 and 3 in the pipeline below) to improve the quality of clustering. The 5 stages in our summarizers are:

1. Preprocess input documents by simplifying sentences
2. Perform sentence clustering over simplified sentences
3. Postprocess clusters by a pruning operation
4. Identify and rank important clusters

5. Generate a sentence from each of the identified clusters till 100 words are generated

Steps 1–3 are common to all our summarizers. We experimented with different strategies for steps 4 and 5 to arrive at our final submissions for MSE 2005. We now overview each step in the pipeline.

2.1 Sentence simplification

We have described elsewhere (Siddharthan et al., 2004) how simplifying text by removing parenthetical information (relative clauses and appositive phrases) results in significantly better sentence clustering by preventing clustering on the basis of background information present in these parentheticals. We use the simplification techniques described in Siddharthan (2003b) and Siddharthan (2003a) for this purpose. As an example of how clustering improves, our simplification routine simplifies:

> PAL, which has been unable to make payments on dlrs 2.1 billion in debt, was devastated by a pilots' strike in June and by the region's currency crisis, which reduced passenger numbers and inflated costs.

to:

> PAL was devastated by a pilots' strike in June and by the region's currency crisis.

Three other sentences also simplify to the extent that they represent PAL being hit by the June strike. The simplified sentences all share the common information about PAL being devastated by the strike, while other extraneous information not pertinent to the strike was removed. The resulting cluster is:

1. PAL was devastated by a pilots' strike in June and by the region's currency crisis.

2. In June, PAL was embroiled in a crippling three-week pilots' strike.

3. Tan wants to retain the 200 pilots because they stood by him when the majority of PAL's pilots staged a devastating strike in June.

4. In June, PAL was embroiled in a crippling three-week pilots' strike.

2.2 Sentence clustering

We cluster the simplified sentences in order to determine important concepts in the input documents. We used *SimFinder* (Hatzivassiloglou et al., 1999) for this purpose.

2.3 Cluster pruning

To further tighten the clusters and ensure that their size is representative of their importance, we post-process them as follows. *SimFinder* implements an incremental approach to clustering. At each incremental step, the similarity of a new sentence to an existing cluster is computed. If this is higher than a threshold, the sentence is added to the cluster. There is no backtracking; once a sentence is added to a cluster, it cannot be removed, even if it is dissimilar to all the sentences added to the cluster in the future. Hence, there are often one or two sentences that have low similarity with the final cluster. We remove these with a post-process that can be considered equivalent to a back-tracking step. We redefine the criteria for a sentence to be part of the final cluster such that it has to be similar (simval above the threshold) to *all* other sentences in the final cluster. We prune the cluster to remove sentences that do not satisfy this criterion. Consider the following cluster and a threshold of 0.65. Each line consists of two sentence ids (*P[sent_id]*) and their simval.

P37	P69	0.9999999999964279
P37	P160	0.8120098824183786
P37	P161	0.8910485867563762
P37	P176	0.8971370325713883
P69	P160	0.8120098824183786
P69	P161	0.8910485867563762
P69	P176	0.8971370325713883
P160	P161	**0.2333051325617611**
P160	P176	**0.0447901658343020**
P161	P176	0.7517636285580539

We mark all the lines with similarity values below the threshold (in bold font). We then remove as few sentences as possible such that these lines are excluded. In this example, it is sufficient to remove $P160$. The final cluster is then:

P37	P69	0.9999999999964279
P37	P161	0.8910485867563762
P37	P176	0.8971370325713883
P69	P161	0.8910485867563762
P69	P176	0.8971370325713883
P161	P176	0.7517636285580539

The result is a much tighter cluster with one sentence less than the original.

2.4 Cluster ranking

We explored the following options for ranking clusters by importance:

- **ImRk1.** Cluster size (number of sentences in the cluster) and TF*IDF to rank clusters of same size
- **ImRk2.** TF*IDF
- **ImRk3.** TF*IDF normalized by number of words in cluster
- **ImRk4.** TF*IDF weighted by cluster size

In addition, we explored multiple strategies for deciding the order in which to select these clusters, given their importance rankings. We partitioned the clusters into three:

- **ClPar1.** Clusters containing only English Sentences
- **ClPar2.** Clusters containing only Machine translated from Arabic Sentences
- **ClPar3.** Clusters contain both English and MT sentences

We then explored three cluster ordering strategies:

- **ClOrd1.** Ignore partitions
- **ClOrd2.** Round Robin
- **ClOrd2.** Proportional to partition sizes

ClOrd1 is the baseline cluster selection scheme (used in Columbia's monolingual summarizer) and selects clusters in the order specified by their importance rankings rankings, ignoring the partitioning into only-English, only-Arabic and mixed clusters. We also explored two cluster selection schemes that are specific to the multilingual task. *ClOrd2* alternately considers one cluster from each of the three partitions. *ClOrd3* considers clusters from the three partitions in proportion to number of clusters in each partition.

The motivation for partitioning the clusters into three is described in Evans and McKeown (2005). In short, the aim of this approach is to summarize changes in perspective between news report on the same events in two different languages. This partitioning allows for a three part summary consisting of 1) information common to reports in both languages 2) information only present in the English reports and 3) information only present in the foreign language. It is unclear how this aim of summarizing perspectives ties in with the MSE'05 task; we thus used a training phase (cf. §3.2) to optimize the various parameters described in this section.

2.5 Generating a sentence from a cluster

We only used extractive techniques at this stage; we explored three strategies for selecting one sentence from a cluster:

- **SntSel1** Most similar
- **SntSel2** Cluster centroid
- **SntSel3** TF*IDF

SntSel1 chooses the sentence that is most similar to every other sentence based on the SIMFINDER similarity score. *SntSel2* computes the vector space weight of all words in the cluster and then chooses the sentence that is closest to the centroid. *SntSel3* selects the sentence with the highest tf*idf score.

For *ClOrd2* and *ClOrd3* cluster ordering options, we had an added option *EngOnly* which only selected English sentences from the mixed partition (*ClPar3*).

We then used a training corpus to select the optimal configurations for importance rankings for clusters (*ImRk[1–4]*), cluster ordering (*ClOr[1–3]*) and sentence selection (*Sent[1–3]*). We describe this phase next.

3 The training phase

As this is the first evaluation for multilingual summarization where document sets are mixed (Some English and some MT Arabic in each set), we had to adapt the data available from the task 4 in DUC 2004. We now overview the DUC 2004 data and how we created our training corpus (§3.1) before describing the results of our training and the configuration of our submissions for MSE 2005 in §3.2.

3.1 DUC 2004 data

The Document Understanding Conference (http://duc.nist.gov) has been run annually since

2001 and is the biggest summarization evaluation effort, with participants from all over the world. In 2004, for the first time, there was a multilingual multi-document summarization task. There were 24 sets to be summarized. For each set consisting of 10 Arabic news reports, the participants were provided with 2 different machine translations into English (using translation software from ISI and IBM). The data provided under DUC includes 4 human summaries for each set for evaluation purposes, and a human translation into English of each of the Arabic news reports.

To mimic the MSE'05 summarization task, in which input document sets contain a mix of machine-translated Arabic text and English source texts, we created three sets from each DUC set by taking:

1. ISI translations of 3 reports and human translations for the other 7
2. 5 ISI translations and 5 human translations
3. 7 ISI translations and 3 human translations

The manual translations were meant to substitute for original English news reports. We ran all possible configurations of our summarizer and evaluated the summaries using Rouge SU4 Average Recall Metric using the Rouge parameters from the MSE 2005 evaluation.

3.2 Configuration obtained by training

We found that when the proportion of MT translations in a set was more than or equal to half, the configuration from the monolingual summarizer described in Siddharthan et al. (2004) gave the best results:

Config1=*ImRk1*, *ClOrd1*, *SntSel3*

When the proportion of MT translations in a set was less than half, the configuration from the multilingual summarizer described in Evans and McKeown (2005) gave the best results:

Config2=*ImRk4*, *ClOrd2*, *SntSel1*, *EngOnly=Y*

We also experimented with and without the preprocessing by sentence simplification, and found that results were significantly better *with* the simplification. These are the configurations we used in MSE 2005 for our submissions:

Priority	Run No.	Configuration
1	10	If %English<=50, **Config1** If %English>50 , **Config2**
2	11	**Config2**
3	12	**Config1**

As we did not have prior knowledge about the percentages of English and Arabic reports in the MSE 2005 evaluation sets, our additional runs were *Config1* and *Config2* individually, while our priority run was the combined configurations.

4 The MSE evaluation results

The MSE results reflected what our training phase had predicted. Our priority run (Run No. 10, which used *config1* or *config2* based on the proportion of English documents in the set) outperformed both the other runs that ran either in either *config1* or *config2* mode:

Run No.	Rouge Metric	Score
10	ROUGE-SU4 Average_R	0.16568
12	ROUGE-SU4 Average_R	0.16560
11	ROUGE-SU4 Average_R	0.14486

This shows the importance of automatic metrics; systems can be trained at little cost using them, even when the training data is not exactly equivalent to testing data. Only 7 out of the 25 test sets contained more than 50% English. This meant that run 11 did not perform particularly well, but its performance on those 7 sets was sufficient to help run 10 score marginally above run 11.

The flip side of using automatic metrics is that a system optimized on one metric needn't be the best when tested using another metric. On Rouge-2, system 12 outperformed our priority run:

Run No.	Rouge Metric	Score
12	ROUGE-2 Average_R	0.13231
10	ROUGE-2 Average_R	0.13038
11	ROUGE-2 Average_R	0.10838

This highlights the importance of identifying an automatic evaluation metric that is reliable — our experience shows that systems can be trained using an automatic metric, but training on a metric only makes sense when that metric can be trusted.

4.1 Relative performance: automatic evaluation

Our runs 10 and 12 performed creditably on SU4 Average Recall (the metric it was optimized for) in comparison to other summarizers at MSE 2005. Table 1 gives the top 10 systems according to this metric (there were 27 submissions in all from 10 different research groups).

Run No.	Rouge-SU4 Av. Recall	95%-conf. int.
28	0.186270	0.17272 - 0.19999)
29	0.169610	0.15889 - 0.18078)
30	0.169060	0.15866 - 0.17973)
10	0.165680	0.15532 - 0.17582)
12	0.165600	0.15716 - 0.17406)
16	0.161770	0.15162 - 0.17300)
17	0.161070	0.15101 - 0.17179)
8	0.159380	0.15103 - 0.16749)
18	0.157020	0.14664 - 0.16800)
1	0.156700	0.14055 - 0.17393)

Table 1: Top 10 systems on Rouge SU4 Average Recall (priority runs in bold, Columbia systems underlined)

28, 29 and 30 were the three runs from the group that performed best on this evaluation. 10 and 12 are our first and third runs. These make up the top 5 systems. However, we see again that testing using a different metric produces a different ranking, though 28, 29 and 30 stay on top and the top 10 systems stay the same despite the differences in relative rankings. Table 2 shows the top 10 systems on Rouge 2.

Table 3 shows the rankings of the top 10 systems according to each of these metrics. When confidence intervals are taken into account, system 28 is significantly better than the rest of the field. All we can say about our entries 10 and 12 are that they are significantly worse than 28, and significantly better than the bottom 17 systems. There is no significant difference between our systems 10 and 12 and 7 other systems.

Run No.	Rouge-2 Av. Recall	95%-conf. int.
28	0.160360	0.14537 - 0.17604
29	0.142570	0.13093 - 0.15489
30	0.139780	0.12816 - 0.15147
16	0.133550	0.12189 - 0.14649
12	0.132310	0.12295 - 0.14180
17	0.131970	0.12012 - 0.14495
1	0.130760	0.11183 - 0.15086
10	0.130380	0.11889 - 0.14141
8	0.126780	0.11640 - 0.13694
18	0.126630	0.11464 - 0.13952

Table 2: Top 10 systems on Rouge 2 Average Recall (priority runs in bold, Columbia systems underlined)

Metric	System Ranking
Rouge-2	**28**, 29, 30, **16**, 12, 17, **1**, **10**, **8**, 18
Rouge-SU4	**28**, 29, 30, **10**, 12, **16**, 17, **8**, 18, **1**

Table 3: Top 10 Systems according to rankings by average recall of different metrics (priority runs in bold, Columbia Summarizers underlined).

Metric (Average)	System Ranking
Pyramid-Precision	1, 28, 19, 8, 16, 10, 13, 25, 4, 7
Pyramid-Recall	1, 28, 8, 16, 19, 10, 4, 25, 7, 13
Rouge-2 Recall	28, 1, 8, 7, 16, 10, 19, 4, 25, 13
Rouge-SU4 Recall	28, 1, 8, 7, 10, 16, 4, 19, 25, 13

Table 4: Rankings of priority runs on 10 manually evaluated sets (Columbia Summarizer underlined).

4.2 Relative performance: manual evaluation

In addition to the automatic evaluation, there was a manual evaluation using the pyramid method (Nenkova and Passonneau, 2004) of 10 out of the 25 sets. Only the priority run of each participant was evaluated manually. On the manual evaluation of the priority runs of the ten participants, our priority run no. 10 came sixth. Table 4 gives the rankings of the 10 priority runs on the pyramid evaluation. We present two different pyramid metrics - the first (Pyramid-P) is a precision metric, where the overall pyramid score is normalized by the number of SCUs in the peer summary. The second (Pyramid-R) is a recall metric where the normalization factor is the average number of SCUs in the model summaries. The Rouge-2 and Rouge-SU4 rankings on these 10 sets are also provided for comparison. Of the 45 binary comparisons possible between systems, Rouge-2 agrees with the Pyramid-R scheme 71% of the time. Rouge-SU4 agrees with the Pyramid-R scheme 67% of the time. There are only two systems, 19 and 7, that move up or down drammatically depending on whether the evaluation is manual or automatic. Other systems only move up or down by one or two positions.

In particular, system 10, our priority run, ranks 5th or 6th irrespective of whether manual or automatic metrics are used on these 10 sets. However, as seen in Table 3, our systems perform quite well on the automatic metrics over all 25 sets. Thus, the MSE 2005 evaluation results can be interpreted in multiple ways, depending on:

1. Whether you consider the manual evaluation scheme reliable

2. Whether you consider the automatic evaluation metrics reliable

3. Whether you believe the 10 manually evaluated sets to be sufficiently representative of all 25

We address these three issues in the next section.

5 Evaluation issues: the MSE 2005 experience

We believe that the pyramid scheme is the most convincing manual evaluation method for content selection to have been used in summarization evaluation exercises to date. It offers significant benefits over the manual evaluations performed under previous DUC competitions — in particular, it compares information units in peer summaries against *all* human summaries. We thus believe that the pyramids accurate a means for comparing content selection as we can hope for at this point in time.

The major issues in interpreting the results are whether the pyramid results from 10 sets can be generalized to rank systems over all 25, whether the automatic evaluation metrics give a reliable indication of summary content, and indeed whether 25 sets are sufficient for an automatic evaluation.

5.1 How representative were the 10 manually evaluated sets?

In terms of size of input, on average the 25 sets contained 4167 words each. However the 10 manually evaluated sets contained the three shortest sets (449, 906 and 1068 words) and the two longest sets (9091 and 8727 words). Short sets are known to cause problems for clustering based summarizers — too few sentences in the input results in bad clustering, and clustering is the basis of our summarization strategy. The long sets require 100 word summaries to achieve a compression of 90:1, more than twice the average. This can also affect summarizer performance.

In terms of the proportion of English to MT documents in a set, there were 7 sets out of 25 (28%) where there was more English than MT. Out of these, four were present in the ten manually evaluated sets (40%). As we use different configurations for these two types of sets, it makes it harder to generalize results from the manual evaluation on 10 sets to system performance on all 25.

In future evaluations, when only a subset of sets can be evaluated manually, it might make sense to select these sets on the basis of how representative they are (in this evaluation, the first 10 sets were evaluated manually).

5.2 Is the average pyramid score a sufficient indicator?

One striking result from the manual evaluation was the variation in performance of every summarizer from set to set. In particular, three summarizers obtained a pyramid score of zero on set 33010 (including system 1 that recorded the highest average pyramid score across all ten sets).

When evaluating a summarizer, how important is robustness; is a summarizer that scores 0.9 and 0.0 on two sets (average=0.45) better than a summarizer that scores 0.5 and 0.3 (average=0.40)? Robustness can be measured by standard deviation — should this be incorporated into the final score, for example, by subtracting standard deviation from the average pyramid score? The three systems which exhibit the largest standard deviation on the pyramid evaluation are 7, 19 and 1; 7 and 19 were the two systems that the automatic and manual evaluations gave markedly different rankings for.

We performed an experiment where we adjusted the Pyramid and Rouge average recall scores by subtracting the standard deviation from the average — this has the effect of penalizing systems that are not robust from set to set. Interestingly, these adjusted scores resulted in better agreement between Pyramid and Rouge rankings on the 10 manual sets (cf. Table 5). The adjusted Rouge-2 Av. recall agrees with the adjusted Pyramid Recall on 84.4% of the 45 possible binary comparisons between systems (compare this with the 71% reported for unadjusted scores).

These adjusted scores also result in less variation

Metric	System Ranking
Pyramid-R Av.	1, 28, 8, 16, 19, 10, 4, 25, 7, 13
Rouge-2 Av. Recall	28, 1, 8, 7, 16, 10, 19, 4, 25, 13
Pyramid-Adjusted	1, 28, 16, 8, 10, 19, 25, 4, 7, 13
Rouge-2-Adjusted	28, 1, 8, 16, 10, 7, 4, 19, 25, 13

Table 5: Rankings of priority runs on 10 manually evaluated sets, showing adjusted scores (Av. minus standard deviation). Columbia Summarizer is underlined.

Metric	System Ranking
Ten Sets	
Rouge-2-Adjusted	28, 1, 8, 16, 10, 7, 4, 19, 25, 13
Pyramid-Adjusted	1, 28, 16, 8, 10, 19, 25, 4, 7, 13
All 25 Sets	
Rouge-2-Adjusted	28, 16, 10, 8, 1, 19, 4, 25, 7, 13

Table 6: Rankings of priority runs on 10 sets, compared to on all 25 sets (Av. minus standard deviation). Columbia Summarizer is underlined.

between rankings on ten sets and rankings on 25 (cf. Table 6) – Using the adjusted Rouge-2 scores, 80% of the 45 possible binary comparisons between systems give the same results on 10 sets and 25 sets. Further on binary comparisons between these 10 systems using adjusted Rouge-2 on all 25 sets and adjusted Pyramid on 10 sets, there is 86.7% agreement.

Using the adjusted Rouge-2 average recall metric, the top 10 systems (from all 27 submission) are shown in Table 7.

Metric	System Ranking
Rouge-2-Adj	**28**, 29, 30, 12, **16**, 17, **10**, **8**, 18, **1**,

Table 7: Top 10 Systems according to rankings by adjusted Rouge-2 average recall (priority runs in bold, Columbia Summarizers underlined).

5.3 How reliable is Rouge?

The various evaluation metrics used by the Rouge package have been tuned to maximize correlation with manual evaluations of DUC summarizers. However, there are known issues with the methodology used by DUC manual evaluations in the past — for example, peer summaries are only compared to one randomly chosen human summary, when it is known that there is variation between human summaries. The pyramid scheme provides us with a methodology manually evaluating peer summaries by comparison with multiple human models. If the pyramid scheme is accepted by the community, time and money would be well invested in creating a corpus of pyramid evaluations for past DUC competitions. This would allow for calibration of automatic metrics against a reliable manual metric, and hopefully make them more reliable.

As shown in the previous section, scores that penalize a system for high standard deviation across sets appear to result in better correlation between manual and automatic metrics. This is worth pursuing further; these results are preliminary, and a larger corpus is required for validating them.

6 Conclusion

In this paper, we have described our summarizer and overviewed the evaluation results. Our experience shows that it is possible to train the parameters of a summarizer to maximize scores on an automatic metric. However, optimizing on one metric does not guarantee a good performance on another metric. This highlights the need to find and agree on evaluation metrics that can be used in system development.

This is the first evaluation exercise that uses the Pyramid scheme for manual evaluation. The Rouge scores do not appear to correlate well with Pyramid scores in this evaluation. While we feel that this was a useful exercise in the sense that it has offered insights into evaluation issues, it is difficult to draw too many conclusions on the performance of different systems.

Part of the problem is due to the large standard deviation of some systems across sets. We have suggested a penalty for lack of robustness across sets. Our adjusted scores (average score - standard deviation) result in better agreement between automatic and manual rankings of summarizers, and less variation in rankings when the number of data sets is changed from 10 to 25. This is worth pursuing further; for validation, a larger data set is required. We believe it will be worthwhile for the community to prepare a larger corpus of summary pyramids that can be used to train automatic metrics. The biggest conclusion we can draw from this evaluation exercise is that the search for a reliable automatic evaluation metric is far from over.

References

R. Barzilay. 2003. *Information Fusion for Multidocument Summarization: Paraphrasing and Generation*. Ph.D. thesis, Columbia University, New York.

D. Evans and K. McKeown. 2005. Identifying similarities and differences across english and arabic news. In *Proceedings of International Conference on Intelligence Analysis*, pages 23–30, McLean, VA.

V. Hatzivassiloglou, J. Klavans, and E. Eskin. 1999. Detecting text similarity over short passages: exploring linguistic feature combinations via machine learning. In *Proceedings of EMNLP'99*, MD, USA.

C. Lin and E. Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of HLT-NAACL'03*, Edmonton.

Ani Nenkova and Rebecca Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In *HLT-NAACL 2004: Main Proceedings*, pages 145–152, Boston, MA, USA.

Advaith Siddharthan, Ani Nenkova, and Kathleen McKeown. 2004. Syntactic simplification for improving content selection in multi-document summarization. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*, pages 896–902, Geneva, Switzerland.

Advaith Siddharthan. 2003a. Resolving pronouns robustly: Plumbing the depths of shallowness. In *Proceedings of the Workshop on Computational Treatments of Anaphora, 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL'03)*, pages 7–14, Budapest, Hungary.

Advaith Siddharthan. 2003b. *Syntactic simplification and Text Cohesion*. Ph.D. thesis, University of Cambridge, UK.