Automation of Summary Evaluation by the Pyramid Method

Aaron Harnly* **Ani Nenkova*** **Rebecca Passonneau*** **Owen Rambow†**
∗ Department of Computer Science, † Center for Computational Learning Systems
Columbia University
New York, NY, USA
{aaron, ani, becky, rambow}@cs.columbia.edu

Abstract

The manual Pyramid method for summary evaluation, which focuses on the task of determining if a summary expresses the same content as a set of manual models, has shown sufficient promise that the Document Understanding Conference 2005 effort will make use of it. However, an automated approach would make the method far more useful for developers and evaluators of automated summarization systems. We present an experimental environment for testing automated evaluation of summaries, pre-annotated for shared information. We reduce the problem to a combination of similarity measure computation and clustering. The best results are achieved with a unigram overlap similarity measure and single-link clustering, which yields high correlation to manual pyramid scores (r=0.942, p=0.01), and shows better correlation than the n-gram overlap automatic approaches of the ROUGE system.

1 Introduction

Automatic summarization is usually evaluated through comparison to human summarization choices for the same texts.[1] Traditionally, the comparison is done through eliciting human judgments on content. When humans write short, abstractive summaries based on their reading of multiple documents, they select content they think belongs in a summary, and put it in their own words. While many words and phrases may be similar to those another human summarizer would employ, people can use different forms of the same words (inflectional or derivational variants), different word order, syntactic structure, and paraphrases. See for example the spans of words in bold below, coming from five different summaries of the same set of documents[2] about a Swissair crash off of Nova Scotia in 1998, all expressing the fact that the cause of the crash has not been determined.

S1 The cause of the Sept. 2, 1998 crash **has not been determined**.

S2 Investigators of a Swissair crash that killed 229 people off the coast of Nova Scotia searched for clues as to a cause but **but refrained from naming one**.

S3 **The cause has not been determined**, but there was extreme heat damage to the front of the aircraft and it is suspected that an in-flight entertainment system had electrical problems.

S4 **The specific cause of the tragedy was never determined**, but suspicions are that an electrical short caused a fire.

S5 Wreckage showed evidence of high heat and heat damaged wiring above the cockpit area but **investigators remain unsure of its cause**.

Note that while this example illustrates some overlap of 4-grams (*has not been determined*), much of the semantic similarity is obscured by alternate phrasings (*was never determined*, *remain unsure*) or by various forms of explicit anaphora (*the tragedy* instead of *the crash*, *its cause* instead of *the cause of the crash*, *naming one* instead of *naming a cause*).

A set of word spans which express similar meaning (such as those in bold in the example above) is referred to as a *Summary Content Unit* (SCU). After similar manual annotation of a complete set of reference summaries, the resulting set of SCUs is called a *pyramid*. A pyramid can be used to evaluate new summaries, following a method proposed by Nenkova & Passonneau (04). Each span of words in an SCU or in a summary to be evaluated is referred to as a *contributor* (and may have discontinuities). A new summary that is to be evaluated against the pyramid (or *peer summary*) will have some contributors that express content already represented in a pyramid, and perhaps some spans that do not. The Pyramid evaluation consists in identifying relevant contributors in the peer summary and matching them against SCUs in the pyramid. This match is used to assign a score, with SCUs that have more contributors providing a higher score. But the Pyramid method goes beyond telling us a score: because of the matching process, we also know which key ideas from the source documents the summary has chosen to include.

In this paper, we explore the automation of this evaluation approach. Since the number of possible

[1]We would like to thank Chin-Yew Lin for helpful comments on an earlier version of this paper. This work was supported by the National Science Foundation under the KDD program. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation.

[2]These sentences are from summaries written by university students for DUC 2003 set *D30016*.

candidate contributor sets is exponential in the number of words in the sentence, we use dynamic programming to find an optimal candidate contributor set of a summary based on different clustering methods and similarity metrics. Our results indicate that using automatic Pyramid scoring leads to better correlation with human Pyramid scoring over the use of the n-gram overlap automatic evaluation metric ROUGE.

2 Related Work

The development of automated or semi-automated methods for evaluating content selection in summarization has recently been an area of active research. A completely manual evaluation method was used in the Document Understanding Conferences (DUC) in 2001–2003. The method involved human judgments about how much of the content of a *single* model summary is expressed in a new peer summary. Analysis of the DUC evaluations results revealed some weaknesses— the stability of human judgments of "information overlap" (Lin & Hovy 02), the coarse-grained and subjective nature of the judgments required (Halteren & Teufel 03; Nenkova & Passonneau 04) , and the use of single reference summaries, despite the observation that summaries with different content can be equally good (Nenkova & Passonneau 04). The "factoid" (Halteren & Teufel 03) and manual Pyramid annotation methods have been proposed to address these limitations.

At the same time, several automated methods have been proposed to address the cost/time issues imposed by manual annotation, most notably the ROUGE family of ngram-overlap measures (Saggion *et al.* 02; Lin & Hovy 03; Pastra & Saggion 03). All of these methods rely on the comparison of peer summaries to one or more human-written reference summaries. The summarization task, by definition, demands high compactness relative to its source documents. Paraphrase and synonymy are expected to be used to achieve the desired compactness, and indeed we find mostly 1- or 2-grams matching between source text and abstractive multi-document summaries (Banko & Vanderwende 04).

3 The Pyramid Method

The pyramid method addresses the following characteristics of abstractive summaries that present a challenge for evaluation: that summaries written by equally skilled writers are highly likely to have some overlap in content, and highly likely to have some content that is unique to each summary; and that when different summaries express the same content, the wording can vary in unpredictable ways. The pyramid method adopts the following strategies:

- We explicitly assume that multiple reference summaries are required to evaluate a peer summary.
- A pyramid is created by identifying SCUs, i.e., sets of contributors (text fragments) in the reference summaries that express approximately the same meaning.
- The number of contributors in an SCU is the frequency with which an SCU was expressed in the pool of model summaries. This frequency is used to weight the importance of the SCU.

A pyramid, or set of SCUs, tends to have very few SCUs with high weights, increasing numbers of SCUs as the weights decrease, and finally, a very large number of SCUs with weights of one or two. It is this fact that gives the method its name.

When a peer summary is evaluated against the pyramid, its content is matched against SCUS to identify *candidate contributors*, which are fragments of text that express roughly the same meaning as an SCU in the pyramid, and there will typically be remaining fragments that have no match. A candidate contributor which has the same meaning as the contributors in an SCU in the pyramid are rewarded with the score n, where n is the weight of the matching SCU in the pyramid. Candidate contributors with no match are assigned weight zero. The score of the peer summary is the ratio between the sum of weights of its candidate contributors and the sum of weights of a optimal summary of the same size. The optimal summary is defined as the informationally ideal summary, that expresses the most highly weighted pyramid SCUs.

4 Automation: Motivation and Algorithms

There are two tasks involved in pyramid evaluation: creating a pyramid by annotating model summaries, and evaluating a new summary (peer) against a pyramid. Ideally, an automated evaluation component would address both tasks. However, the task of creating a pyramid is far more complex than the task of scoring a new summary against existing (hand-created) pyramid, and the automated scoring component is useful when doing a large amount of evaluation (of multiple summarizers, or different versions of the same summarizer). Therefore, we decided to explore first the automation of scoring a new summary

against an extant, human-produced pyramid. We anticipate that what we learn in this process will apply when we turn to automating pyramid construction.

Our algorithm consists of four steps.

Enumerate Enumerate all candidate contributors (contiguous phrases) in each sentence of the peer summary.

Match For each candidate contributor, find the most similar SCU in the pyramid. In the process, the similarity between the candidate contributor and all pyramid SCUs is computed.

Select From the set of candidate contributors, find a covering, disjoint set of contributors that have maximum overall similarity with the pyramid.

Score Calculate the pyramid score for the summary, using the chosen contributors and their SCU weights.

For example, the enumeration of all candidate contributors for a peer summary sentence ABC might be $\{A, B, C, AB, AC, BC, ABC\}$, where A, B, and C are words. In the **Match** step, each member of this set will be assigned a score, based on its similarity with pyramid SCUs. In the **Select** step, the overall optimal subset of candidates will be chosen, for example $\{A, BC\}$ and A and BC will also be mapped to SCUs in the pyramid. In the **Score** step, the pyramid summary score for the peer based on the SCU assignment from the previous step will be computed. We next discuss the four steps in detail.

4.1 Enumeration of candidate contributors

What set of text fragments could be contributors in an SCU? We have chosen to consider all contiguous spans of words that do not cross sentence boundaries. Without the restriction that the candidate contributor spans be contiguous spans of words, an n-word sentence would yield 2^n possible candidate contributors consisting of all possible subsets of words from the original sentence. But imposing the contiguity requirement on candidate contributors, the size of the set of all candidate contributors is reduced to $\frac{n(n-1)}{2} = 1+2+...+n$ since there are $(n+1-k)$ contributors of length k. Note that this restriction to contiguous spans of words is a departure from the manual pyramid method, which permits, in limited circumstances, noncontiguous words to comprise a contributor.

4.2 Matching of contributors to SCUs

Next, we match each candidate contributor $\hat{c}$ to the SCU $C = \{c_1, c_2, ..., c_n\}$ with which it shares the most meaning (c_i are the contributors of C and express the same meaning, possibly with a different wording). The degree of shared meaning is measured using a similarity metric `set_sim` between the candidate contributor and a pyramid SCU:

$$\texttt{set\_sim}(\hat{c}, C) = \texttt{combine}_{c_i \in C}(\texttt{span\_sim}(\hat{c}, c_i))$$

`set_sim` is defined in terms of a function `span_sim` which expresses the similarity between two text spans, and the function `combine`, which, given scores for the similarity between the candidate contributor and the contributors from the pyramid, returns a single score. Thus, we must choose the two functions `span_sim` and `combine`, and these choices represent an important part of our research. Note that the **Matching** step can be seen as a clustering problem. The SCUs in the gold-standard pyramid can be viewed as clusters of contributors. The task is to merge the candidate contributor (viewed as a cluster with a single element) to the most appropriate SCU cluster in the pyramid.

We explore several choices for `combine`. In the single-link method, the overall similarity between the candidate contributor and an SCU is the maximum of the pairwise span similarity between their contributors, i.e., $\texttt{combine} = \texttt{max}$. In the average-link method, the overall similarity is the mean of pairwise similarity, and $\texttt{combine} = \texttt{mean}$. In the complete-link method, the overall similarity is the minimum of the pairwise similarity, and $\texttt{combine} = \texttt{min}$.

Many alternatives for the pairwise similarity metric `span_sim` between contributors are possible. We experimented with simple cosine similarity, cosine similarity with TF*IDF weighting, unigram overlap, bigram overlap, and word-wise edit distance.

Currently, we assign each contributor to its "best fit" SCU. It may be that retaining an n-best list would allow the next step (**Select**) to choose a disjoint set of contributors.

4.3 Selecting a covering, disjoint set of possible contributors

Once all candidate contributors have been matched to their most-similar SCUs, the similarity scores can be used to find an optimal subset of the candidate SCUs. As in the manual pyramid method, we have chosen to require a covering, disjoint set of contributors, i.e. each word of a peer summary should belong to one

of the final contributors, and no word can belong to more than one contributor. There are $O(2^n)$ possible such sets for sentences of n words; to avoid exponential runtime, we use a two-dimensional dynamic programing algorithm, which selects the best contributor set for each span of words between the ith and jth words of a sentence, eventually producing a preferred covering for the entire sentence. The scoring method chooses the contributor set that produces the highest total overall similarity score between the chosen contributors and their SCUs. The score for the best covering for a span (i, j) in a sentence is the maximum of the sums of the scores of the subsequences (i, k) and $(k + 1, j)$ for $k = i, \ldots, j - 1$, and of the direct score for the span (i, j) itself.

Consider a brief example with a sentence beginning *In 1998 two Libyans* .... Initially the span (1, 1) is considered, and hence the optimal contributor set is simply the word *In*. The overall score for this span is simply the similarity score between *In* and its best-match SCU. Next, the spans (1,2) and (2,2) are considered. The optimal contributor set for the span (2,2) is simply the word *1998*. The dynamic programming comes into play in the next span, (1,2). The optimal set of contributors for the span (1,2) can be either the contributor *In 1998* (i.e., the span (1,2)), or the union of each of the optimal sets for the spans (1,1) and (2,2), i.e. *In* and *1998*. Suppose that the single-contributor set *In 1998* produces a better score. We record this fact and need not examine the span (1,2) again, even as this span participates in larger spans. Then we consider the spans (1,3), (2,3), and (3,3). The process continues in typical dynamic programming fashion until an optimal set of contributors for the span (1,n) is chosen.

4.4 Score

Finally, the selected set of contributors are scored as in the manual pyramid method. The sum of the weights of all SCUs in the peer summary (assigned in the preceding step) is normalized by the maximum sum possible for an "ideal" summary which contains as many high-weight SCUs as possible in a summary of the same size (see section 3). This gives a normalized score between 0 and 1.

5 Evaluation

5.1 Comparing Human Pyramid Score to Automated Pyramid Score

The goal of this evaluation is to determine the correlation between human Pyramid scores and our automatically obtained Pyramid scores. It is not the object of this paper to show that the human Pyramid scores correlate with other measures of summary quality; see (Nenkova & Passonneau 04) for details. Because of methodological issues in averaging correlations, we use for our correlation study not the scores for individual summaries, but instead for human summarizers. This evaluation mimics the standard case where we wish to evaluate (or rank) several summarization systems which have produced summaries for the same document sets.

For our evaluation, we used the three sets of data from (Nenkova & Passonneau 04). The three document sets are from the DUC'03 test set. For each document set, we have 10 summaries, each manually annotated for content units. We chose to evaluate six human summarizers from whom we had summaries for each of the three sets (the other summarizers did not summarize all three sets). These summarizers are Columbia University graduate students in the School of Journalism, who were compensated for their work, and who followed the guidelines for summary creation used in DUC.

We evaluated each summary by one of the six Columbia summarizers against a pyramid consisting of the remaining nine summaries for that document set. This gives us 18 manual and 18 automated scores. To obtain an overall summarizer performance score, we calculated the mean human Pyramid score and mean automated Pyramid score for each summarizer across the three sets, giving us six scores for each scoring method (human or automated). Then we computed the correlation between the automatic scores and the original Pyramid scores. Both Pearson's correlation (a measure of the linear association between the two types of score), and Spearman's rank correlation (a correlation based only on the rank of the scores, not their value) were computed. The Pearson correlation is a useful measure of whether the automatic scores could be used as drop-in replacements for human scores. Since the usual ultimate goal of summary evaluation is to compare summarization systems, and hence relative rank rather than raw score is more important, the Spearman rank correlation is arguably a better measure of whether the automated evaluation system can produce similar judgments as human scorers.

Figure 1 shows the main results. The upper table is the Pearson correlation, the lower table the Spearman rank correlation. The rows are labeled with the `span_sim` metric used to compute the sim-

	Min	Mean	Max
Unigram Overlap	**0.942***	0.866*	0.026
Simple Cosine	0.890*	0.751*	−0.052
Edit Distance	0.941*	0.551	−0.1478
Bigram Overlap	−0.119	−0.085	0.529
Cosine-TF*IDF	0.268	0.717	−0.074

	Min	Mean	Max
Unigram Overlap	**0.886***	0.714	−0.029
Simple Cosine	0.886*	0.257	−0.200
Edit Distance	0.886*	0.371	−0.143
Bigram Overlap	−0.200	−0.086	0.428
Cosine-TF*IDF	0.200	0.771	0.086

Figure 1: Pearson (above) and Spearman (below) correlation between automatically scored summary and fully manual scores, for different `scan_sim` functions (rows) and `combine` functions (columns). Starred cells (*) have a p-value ≤ 0.05, single-tailed.

Stop words list	Yes	No
Words unchanged	0.843*	0.726*
Lowercased	0.903*	0.594
Lemmatized	**0.942***	0.819*

Stop words list	Yes	No
Words unchanged	**0.943***	0.714
Lowercased	0.829*	0.371
Lemmatized	0.886*	0.600

Figure 2: Pearson (above) and Spearman (below) correlation for different ways of preparing data. All results in Figure 1 are for Lemmatized, Using Stop Words List. All Results here are for Min, Unigram Overlap. Starred cells (*) have a p-value ≤ 0.05, single-tailed.

ilarity between a candidate span is to a contributor. The columns are labeled with the different `combine` functions, which, as discussed above, correspond to choosing a method in clustering. All figures assume the use of a stop list and a lemmatizer; we return to these parameters below. We have boldfaced the best results, which for both types of correlation is a unigram overlap `span_sim` metric, with the `combine` function being the minimum.

We make the following interpretative observations about the results in Figure 1. We find that for different `combine` methods, different `span_sim` metrics are better. The unigram overlap metric counts the number of shared words between two spans, but abstracts completely from word order. By using the minimum `combine` function (i.e., the single-link clustering method), we require that *all* contributors for a particular SCU in the pyramid show some word overlap with the candidate. Thus, we want a similarity metric which imposes as few constraints as possible, which is the unigram metric. (In fact, we fail to identify the correct SCU if there is a contributor which is a radical paraphrase, to the point of having no overlapping words at all.) On the other hand, for the maximum `combine` function, we require only one contributor to match, so we expect this match to be more constrained. Indeed, for the maximum `combine` function, the best overlap metric is the cosine-TF*IDF metric. In contrast, for minimum and mean, the cosine-TF*IDF is the worst performing.

The lower table in Figure 1 shows the Spearman rank correlation. We see that the results are similar to the Pearson correlation, but with some exceptions, especially for the maximum and mean `combine` functions.

5.2 Preprocessing the Data

Further, we examine how the different ways to prepare the data impacts results. We consider two questions:

- Should we use a list of stop words, which we exclude from both SCU contributors and candidate sentences before we apply the similarity metrics?
- Should we normalize words by either lemmatizing them, or lowercasing them, or should we leave them unchanged?

To investigate these issues, we used the best performing combination `span_sim` metric and `combine` function, namely unigram overlap and minimum. We then varied the two new parameters. The results are shown in Figure 2. As expected, the use of a stop word list helps, since it eliminates noise caused by matches on function words and other content-free or common words. At the same time, we find that we get a slight improvement by lemmatizing words, but only for the Pearson correlation. For the Spearman (rank) correlation, keeping the words unchanged results in a higher correlation, a difference for which we have no explanation at present. Overall, our best results are 0.942 for the Pearson and 0.943 for the Spearman correlations (both significant with $p < 0.05$.

5.3 Comparison with ROUGE

We compare our results with those achieved by the ROUGE system. We report recall and precision scores for ROUGE-1 (the most used metric until 2005), ROUGE-2 and ROUGE-SU4 (which are used for the

	Recall	Precision
ROUGE-1	**0.805**	0.242*
ROUGE-2	0.552*	0.212*
ROUGE-SU4	0.572*	0.176*
Automatic Pyramid	0.942	

	Recall	Precision
ROUGE-1	0.600*	0.543*
ROUGE-2	0.543*	0.371*
ROUGE-SU4	0.314*	0.118*
Automatic Pyramid	0.943	

Figure 3: Summary of results: Pearson (above) and Spearman (below) correlations between manual pyramid scores and six different versions of ROUGE. Starred cells (*) are significantly different from corresponding correlations between the manual and automated Pyramid methods at a p-value ≤ 0.05, single-tailed.

DUC'05 evaluation). ROUGE was originally developed as a recall metric — in fact, its name is an acronym for Recall-Oriented Understudy for Gisting Evaluation. The precision version of ROUGE was added in 2005. The Pyramid evaluation has characteristics of both a precision measure (as the score is a function of the size of the summary) and of a recall measure (as the score is also a function of the weights of the optimal SCUs). The settings we used for all ROUGE experiments were exactly the ones used in DUC.[3]

Figure 3 compares our performance to ROUGE. We use three ROUGE variants: unigram overlap (ROUGE-1), bigram overlap (ROUGE-2), and skip bigram and unigram combination(ROUGE-SU4), where a skip bigram is any pair of words in their sentence order, with up to four intervening words in between. We report both recall and precision scores for the ROUGEs. We see that the automatic Pyramid evaluation has higher Pearson and Spearman correlation than all three ROUGE scores. The difference in correlation between the automatic Pyramid and the ROUGE scores is statistically significant ($p \leq 0.05$) for all cases except the Pearson correlation between the automatic Pyramid (0.942) and ROUGE-1 recall score (0.805), which is not statistically significant ($p = 0.129$). We expect that more data will allow us to establish statistical significance for the remaining comparison as well.[4]

[3]ROUGE-1.5.5.pl -n 2 -x -m -2 4 -u -c 95 -r 1000 -f A -p 0.5 -t 0 -d

[4]We also performed experiments with ROUGE with stopwords removed, which did not lead to a consistent improvement in correlations.

Note that for ROUGE, as for our automatic evaluation, unigrams performs best, followed by the skip bigrams/unigrams combination, followed by the bigrams. The differences among the ROUGE scores are considerable. Experiments on the correlation between ROUGE and the DUC manual evaluation showed that for both DUC'02 and DUC'03 hundred words summaries, the best correlation was achieved for bigram matches, with stopwords removed (Lin 04). We have no immediate explanation for our different result (favoring unigrams), other than to point out that the human evaluations (to which correlation is being measured) differ.

6 Discussion and Future Work

We consider the work reported in this paper to be a foundation for future work. In this section, we discuss some possible extensions of this approach.

6.1 Tree-Based Approaches

We initially explored a more linguistically motivated order of operations, in which the peer summary was first broken into text fragments corresponding to subtrees in a dependency parse of the sentence, using a machine learning approach with human-annotated summaries as training data. The use of dependency tree representations was motivated by the observation that the overwhelming majority of SCU contributors chosen by humans are in a single subtree of a dependency tree, in particular, including constituents that are discontinuous in surface structure. For example, in *The report, later published by the Times, cost the government half a million*, the *later published by the Times* may be a separate contributor, making *The report . . . cost the government half a million* discontinuous, but only in the linear order, not in the tree. In addition, we hoped to develop a feature set that would take advantage of dependency relations to express more of the semantics of a contributor than is given by the actual word sequence; e.g., that a temporal locative PP like *on November 9* gives the date of the event described in the governing phrase.

The approach uses a set of features extracted from a dependency tree of each sentence to machine-learn the binary classifier of whether to "clip" each subtree into a separate contributor. However, this method does not yield contributors that are very similar to those chosen by human annotators The likely reason for the poor performance is that this purely local and syntactic selection of contributors does not capture the key decision in SCU contributor selection, which is

whether a possible contributor expresses roughly the same meaning as other contributors from reference summaries. Therefore we rejected the purely syntactic method of contributor selection in favor of the above set of steps, which performs an optimization over the whole summary.

A natural consideration is extending the dynamic programming approach proposed here to trees. We would enumerate all subtrees of a dependency parse as possible contributors, and compare them to trees derived from the contributors in the pyramid. Unfortunately, this approach would also produce exponentially many candidate contributors. A solution may be to use dynamic programming in the matching itself (and not just in the selection of a covering, as we do now), so that when we match a larger tree, we base the results on the matches of its constituent trees.

6.2 Improving the Matching

For the span distance function `span_sim`, we can consider variants such as word-wise edit distance weighted by TF*IDF scores, centroid measures, and so on. Even more sophisticated possibilities include a tree edit distance of a dependency parse of the contributors, or incorporating syntactic features in other ways, for example favoring contributors that are bounded on either side by a mother and child in the dependency tree. (In this proposal, the contributors are still defined as word sequences but are then parsed, unlike the tree-based approaches proposed in Section 6.1, where contributors are defined in terms of tree structure.)

Another possible strategy is to measure similarity of the target contributor to a derived template contributor in the pyramid that incorporates elements of each member contributor. Or, borrowing from computational biology, one can do a multiple sequence alignment of the peer candidate contributor to the entire set of member contributors.

For the score combination function `combine`, we found that the single-link method produces SCU assignments with highest accuracy compared to human judgments; but this choice can be revisited as we choose different similarity metrics (`span_sim`) in that there is likely to be a trade-off between the features and weightings associated with a specific metric, and the way pairwise similarity scores of a candidate with each SCU contributor are combined.

6.3 Score Stability

The manual pyramid method has been found to elicit stable rankings of individual summaries when five or more reference summaries are used (Nenkova & Passonneau 04). It would be interesting to discover whether the automatic Pyramid scoring method shows similar behavior, and to investigate system rankings from the automatic Pyramid method across more document sets, to explore whether stable single-summary scores yield stable system ranking across many document sets, and to determine whether even unstable single-summary scores could yield stable rankings over a sufficient number of document sets.

7 Conclusion

We have presented a method for automation of summary evaluation that incorporates the insights of the manual Pyramid method. We believe the method, in addition to correlating better with human Pyramid scores on our test set, offers some advantages over the automated ROUGE methods, as it is a more general framework that takes human insight into meaning into account, and that can incorporate different ways of measuring similarity, not simply n-grams.

References

(Banko & Vanderwende 04) Michele Banko and Lucy Vanderwende. Using n-grams to understand the nature of summaries. In *Proceedings of HLT/NAACL'04*, 2004.

(Halteren & Teufel 03) Hans Halteren and Simone Teufel. Examining the consensus between human summaries: initial experiments with factoid analysis. In *HLT-NAACL DUC Workshop*, 2003.

(Lin & Hovy 02) Chin-Yew Lin and Eduard Hovy. Manual and automatic evaluation of summaries. In *Proceedings of the Workshop on Automatic Summarization, post conference workshop of ACL 2002*, 2002.

(Lin & Hovy 03) Chin-Yew Lin and Eduard Hovy. Automatic evaluation of summaries using n-gram co-occurance statistics. In *Proceedings of HLT-NAACL 2003*, 2003.

(Lin 04) Chin-Yew Lin. Rouge: a package for automatic evaluation of summaries. In *Proceedings of the Workshop in Text Summarization, ACL'04*, 2004.

(Nenkova & Passonneau 04) Ani Nenkova and Rebecca Passonneau. Evaluating content selection in summarization: The pyramid method. In *Proceedings of HLT/NAACL 2004*, 2004.

(Pastra & Saggion 03) Katerina Pastra and Horacio Saggion. Colouring summaries bleu. In *EACL 2003*, 2003.

(Saggion *et al.* 02) H. Saggion, D. Radev, S. Teufel, and W. Lam. Meta-evaluation of summaries in a cross-lingual environment using content-based metrics. In *International Conference on Computational Linguistics (COLING'02)*, 2002.