Toward a Gold Standard for Extractive Text Summarization

Alistair Kennedy[1] and Stan Szpakowicz[1,2]

[1] School of Information Technology and Engineering
University of Ottawa, Ottawa, Ontario, Canada
{akennedy,szpak}@site.uottawa.ca
[2] Institute of Computer Science
Polish Academy of Sciences, Warsaw, Poland

Abstract. Extractive text summarization is the process of selecting relevant sentences from a collection of documents, perhaps only a single document, and arranging such sentences in a purposeful way to form a summary of this collection. The question arises just how good extractive summarization can ever be. Without generating language to express the gist of a text – its abstract – can we expect to make summaries which are both readable and informative? In search for an answer, we employed a corpus partially labelled with Summary Content Units: snippets which convey the main ideas in the document collection. Starting from this corpus, we created SCU-optimal summaries for extractive summarization. We support the claim of optimality with a series of experiments.

1 Introduction

One of the hardest tasks in Natural Language Processing is text summarization: given a document or a collection of related documents, generate a short – often very short – text which presents only the main points of those documents. There can be a generic summary, when there are no restrictions other than the required compression into the most salient points, or a query-driven summary, when we seek answers to one or more questions, or focus on the broad topic of the query. Language generation is quite a difficult task, for which no easily applicable tools exist in the public domain; in any event, generation would require the creation of a detailed formal model of the summary, itself a formidable task. That is why summarization systems usually rely on *extracting* a set of relevant sentences and then arranging them into a summary. The inherent imperfection of such summaries invites some obvious questions: just how good can an extractive summary ever be? Does the reliance on stitching sentences together rather than generating new text mean that we cannot hope to achieve the quality of summaries generated by hand? We will argue that many of the criteria which underlie summary evaluation can be re-examined by way of building and evaluating upper-bound extractive summaries.

Previous work on finding baseline systems for extractive summarization includes Optimal Position Policy [1] and Sub-Optimal Position Policy [2]. Those

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 63–74, 2010.

are *lower* bounds on how good an automatic text summarization system should be. We present an *upper* bound on how good an extractive summary we can really hope to produce. We evaluate sample summaries using a variety of standard evaluation techniques, including manual evaluation and semi-automatic methods, specifically ROUGE [3]. Unlike algorithms for building baseline summaries, which require no summarizer output for comparison, our gold-standard summaries are built using a corpus labelled with Summary Content Units (SCUs) – Section 1.1 defines them. The SCU-labelled corpus has been compiled every year since 2005, and first noted in [4]. We attempt to maximize the number of SCUs according to one of two criteria. We believe that generating SCU-optimal summaries is an important step toward generating a gold standard for extractive text summarization. There has been earlier work on manually constructing extractive upper bounds [5]. In TAC 2009, a run of that system was evaluated and scored at or near the top in terms of responsiveness, readability and SCU scores. That was a rating with respect to the other peer systems; manually built reference summaries still did better.

The Text Analysis Conference (TAC; formerly Document Understanding Conference, or DUC), organized annually by the National Institute of Standards and Technology (NIST), includes tasks in text summarization. In 2005-2007, the challenge was to generate 250-word summaries of news article collections of 20-50 articles. Summaries were to be built around a query – a few questions on the main topic of the collection and perhaps postulates for how to answer the questions. In 2008-2009 (after a 2007 pilot), the focus has shifted to creating *update summaries*. The document set is split into a few subsets. From each subset, a 100-word summary is generated. The subsets are ordered chronologically, and the goal is to exclude from a summary any information which can be found in a previous document set. For example, given subsets A_1, A_2 and A_3, a summary for A_1, $sum(A_1)$, will be generated normally, while $sum(A_2)$ must not contain any information found in document set A_1. Likewise $sum(A_3)$ should not contain information from document sets A_1 and A_2.

1.1 Manual Evaluation at TAC

Manual summary evaluation[1] at DUC/TAC, financed by NIST, is an expensive but highly useful part of the exercise. It includes *pyramid evaluation*, outlined in [6], which begins with creating several reference summaries and determining what information they contain is most relevant. A relevant element is called a Summary Content Unit (SCU), carried in text by a varying-size fragment, between a few words and a complete sentence. All SCUs, marked in the reference summaries, make up a so-called pyramid, with few frequent SCUs at the top and many rare ones at the bottom. In the actual pyramid evaluation, annotators use a custom-made tool to identify SCU occurrences in peer summaries. More SCUs mean more relevance for a peer summary; there may be redundancy if a SCU appears more than once. If a peer summary contains relevant information absent from reference summaries, the tool allows the creation of a new SCU. Two

[1] See ⟨`www.nist.gov/tac/2009/Summarization/update.summ.09.guidelines.html`⟩

kinds of scores measure the quality of the summary after pyramid evaluation: the pyramid score (precision) and the modified pyramid score (recall) [6]. Only modified pyramid scores are reported in TAC.

Pyramid evaluation supplements the manual summary evaluation for readability and overall responsiveness. Readability evaluates a mixture of grammaticality, non-redundancy, referential clarity, focus, and structure/coherence. For the latter, trained evaluators read peer summaries and score them 1 to 10.

1.2 Semi-automatic Summary Evaluation

Semi-automatic evaluation, notably ROUGE [3], requires hand-made reference summaries to measure lexical overlap between these references and a summary generated by a participating summarization system – a *peer summary* in the TAC terminology. Variations on this method include using synonyms and finding matching sub-sequences. The two used in TAC/DUC are ROUGE-2 and ROUGE-SU4. ROUGE-2 seeks to match bigrams between the peer and reference summaries. ROUGE-SU4 allows for unigrams and for *skip bigrams*, non-adjacent word pairs up to 4 spaces apart. In [7], an upper bound on ROUGE scores was determined by selecting sentences which contained bigrams most frequent in the reference summaries. The results for the 2008 data are presented in Section 5.

Another method used in recent years is Basic Elements (BE) [8]. It works similarly to ROUGE, but requires syntactic parsing to match syntactic structures in reference summaries and peer summaries. Both the BE and ROUGE measures have been found highly correlated with the responsiveness of a summary, 0.975 and 0.972 Pearson correlation respectively on the 2005 DUC data.

2 The SCU-Labelled Corpus

One of the primary advantages of pyramid evaluation is that it provides us with fully annotated peer summaries. Assuming, then, that TAC peers usually build extractive summaries, it becomes feasible to map the sentences from these summaries back to the original corpus [4]. Many sentences in the corpus can be labelled with the list of SCUs they contain, as well as the score for each of these SCUs and their identifiers. [9] reported a mapping back to the original corpus of 83% and 96% of the sentences from the peer summaries in 2005 and 2006 respectively. A dataset has been generated for the DUC/TAC main task data in years 2005-2009, and the update task in 2007. This corpus indicates what useful information is included in a sentence and can be used to give sentences scores. We generate the SCU-optimal summaries by assembling summaries from the highest-scored sentences.

Figure 1 illustrates the format of the data. The example comes from the 2008 data set D0801; the goal was to build a summary around the query "Airbus A380 – Describe developments in the production and launch of the Airbus A380". The first sentence is tagged with the <annotation> tag indicating that it was used in at least one summary. This sentence appeared in exactly one summary, with ID 0. There are two SCUs. One, with ID 11, is "Airbus A380 flew its maiden

<line>*As opposed to the international media hype that surrounded last week's flight, with hundreds of journalists on site to capture the historic moment, Airbus chose to conduct Wednesday's test more discreetly.* <annotation scu-count=”2” sum-count=”1” sums=”0”><scu uid=”11” label=”Airbus A380 flew its maiden test flight” weight=”4”/><scu uid=”12” label=”taking its maiden flight April 27” weight=”3”/></annotation> </line>

<line>*After its glitzy debut, the new Airbus super-jumbo jet A380 now must prove soon it can fly, and eventually turn a profit.*<annotation scu-count=”0” sum-count=”3” sums=”14,44,57”/> </line>

<line> *"The takeoff went perfectly," Alain Garcia, an Airbus engineering executive, told the LCI television station in Paris.*</line>

Fig. 1. Positive, negative and unlabelled sentence examples for the query "Airbus A380 – Describe developments in the production and launch of the Airbus A380"

test flight" with a weight of 4. The other, with ID 12, is "taking its maiden flight April 27" with a weight of 2. This is an example of a positive sentence with a weight of 6. The second sentence in Figure 1 is annotated but has a SCU count of 0. This means that the sentence was used – in three summaries numbered 14, 44 and 57 – but no SCU is contained in the sentences. Such sentences are negative examples. The third example in Figure 1 was not used in any summary, so it has no annotations. We call it an *unlabelled* sentence. The complete SCU-labelled corpus contains 19247 labelled sentences from a total set of 91658; Table 1 gives the number of positive, negative and unlabelled sentences.

The labelled part of the corpus contains about 40% positive and 60% negative examples. We cannot assume the same distribution in the unlabelled data, so we cannot really be sure how many positive and negative sentences are in the corpus as a whole. One way of estimating this is to graph the likelihood of a sentence containing a SCU against the number of summaries where this SCU appears. We present this graph in Figure 2; it shows the accuracy and the proportion of the sentences from the data set which appeared in a given number of summaries. When a sentence appears in a large number of summaries, it is more likely to

Table 1. Counts of the positive, negative and unlabelled SCU data

Year	Pos	Neg	Unlabelled	% Labelled
2005	1187	1490	16176	14.2%
2006	988	1368	11642	16.8%
2007	937	975	10670	15.2%
2007-A	201	233	1580	21.5%
2007-B	178	285	955	32.7%
2007-C	164	289	912	33.2%
2008-A	1223	1140	8639	21.5%
2008-B	969	1519	7753	24.3%
2009-A	992	2075	7511	30.0%
2009-B	794	2241	6572	31.6%
Total	7633	11615	72410	21.0%

contain a SCU than when it appears in just one or two summaries. The data for sentences which appear in a large number of peer summaries (5 or more) in Figure 2 are quite erratic. This is largely because there are so few such cases; when we consider sentences which appeared in fewer than 5 summaries, we begin to see a trend. If we perform linear regression on these four points, we will expect an accuracy of about 0.22 on sentences which appeared in zero summaries – the unlabelled data. This would mean that, from the 72410 unlabelled examples, about 15930 would have been positively labelled had they appeared in a summary evaluated using the pyramid method. This suggests that our data set currently identifies about one third of all the positive sentences. That is why our SCU-optimal summaries may not actually contain the highest SCU scores possible, but we will find that they do have very high SCU counts.

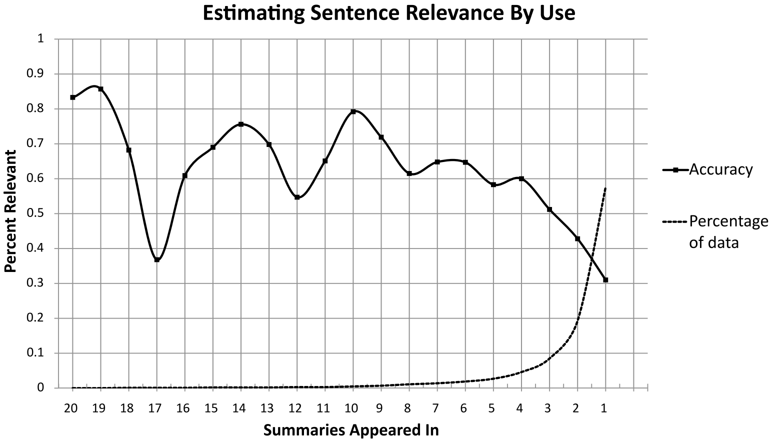

Fig. 2. Frequency of the sentences used by the system summaries

2.1 Previous Users of the SCU-Labelled Corpus

Parts of the SCU-labelled corpus have been used in other research. In [10], the 2005 data are the means for evaluating two sentence-ranking summarization algorithms. In [11], Support Vector Machine is trained on positive and negative sentences from the 2006 DUC data and tested on the 2005 data. The features include sentence position, lexical overlap with the query and others based on text cohesion.

In [2], the SCU-based corpus is used to find a baseline algorithm for update summarization called Sub-Optimal Position Policy (SPP). This is an extension of Optimal Position Policy (OPP) [1] where sentences are selected based on their location in a document. The SCU corpus from 2005-2006 was used for learning SPP, while the 2007 and 2008 data was used for testing.

In [12], the SCU-labelled corpus from 2005 - 2007 is used to identify whether summaries generated automatically tend to be query-focused or query-biased.

A query-focused summary is one built to answer a query, while a query-biased summary is one that selects sentences with as much overlap with the query as possible. It turns out that words found in the query are much more likely to be repeated in machine-generated summaries than in human-made summaries. This is not altogether surprising, because many summarizers determine relevant sentences by measuring lexical overlap with the query.

3 Building a SCU-Optimal Summary

Using the SCU-labelled corpus, we create theoretical upper bounds on how good an extractive summary can possibly be. This is done by selecting sentences in a way to maximize one of two measures:

- combined weight of all SCUs in a summary – maximum SCU weight (MSW);
- combined weight of unique SCUs in a summary – maximum unique SCU weight (MUSW).

Summaries of no more than 100 words are generated, maximizing one of these two criteria. For evaluation we run experiments on the 2008-A and 2009-A TAC data sets, 48 and 44 document sets respectively. Since our aim is to only discover an upper bound for extractive text summarization, not specific to update summaries, we only used the A data sets, not the B sets. We also did not select data from earlier years, because summaries of size 250 became extremely difficult to build for the maximum unique SCU count – see Section 3.2. It should be noted that our SCU-optimal methods will necessarily have higher pyramid scores than any other extractive method tested at TAC. This happens because every sentence they used was labelled in our SCU-labelled corpus and we generate SCU-optimal summaries from this corpus.

3.1 Maximum SCU Weight

Building a summary that maximizes the total SCU score is not difficult. Each sentence has a length and a score where we wish to select the sentences whose combined length is $\leq$ 100 words in a way that maximizes the weight of these sentences. This is in an instance of the well known 0-1 knapsack problem, which can be solved using a dynamic programming algorithm.

We created maximum-weight summaries for all the document sets in the 2008-A and 2009-A data sets. Next we summed all the SCU scores together for each summary. Table 2 shows the results for total SCU weight, total SCU count, unique SCU weight, unique SCU count number of redundant SCUs and total number of sentences. This is the *maximum SCU weight* method (MSW).

3.2 Maximum Unique SCU Weight

For this upper bound we want to maximize the weight of the unique SCUs. Unlike creating summaries which maximize the total SCU weight, we cannot apply

Table 2. Counts for SCU-optimal summaries – MSW & MUSW

	MSW		MUSW	
	2008-A	2009-A	2008-A	2009-A
Total Weight	1430	1932	1178	1464
Total SCUs	518	717	476	593
Unique Weight	932	1104	1132	1298
Unique SCUs	361	454	476	538
Redundant SCUs	157	263	20	55
# of Sentences	213	178	212	167

simple dynamic programming: the score of a sentence depends on every other sentence in the summary. Instead, we built a brute-force algorithm which recursively branches whenever it decides whether to add a sentence to a summary. This algorithm's run time will grow exponentially with the size of the summary generated, but it can still build summaries of up to 100 words in a timely fashion, taking under a minute each on a computer with 2.4 GHz Intel Core 2 Duo processor. (Generating 250-word summaries from the 2005-2007 DUC data becomes prohibitively slow: no summaries built after an hour.) We refer to this method as *maximum unique SCU weight* (MUSW). The results appear in Table 2. This method is the better of the two upper bounds, because our goal should be to maximize unique information. MSW is presented mostly for comparison's sake.

3.3 Sample Summaries

Figure 3 shows sample summaries for the MSW and MUSW methods. The order of the sentences could be changed in a SCU-optimal summary, but this will not change the total SCU score or unique SCU score.

4 Pyramid Evaluation

Naturally, the MWS system will have a higher total SCU weight and the MUSW system will have a higher unique SCU weight as seen in Table 2. We note, however, that the MUSW method still gives redundant SCUs: 4% from 2008-A and 9% from 2009-A. This happens because some SCUs appear in so many sentences that in order to maximize unique SCU weight the summary must repeat some information. It is worth remembering that to maximize unique SCUs does not necessarily mean to eliminate all redundancy. Comparably the MSW summaries have a very high amount of redundancy. On average there are 0.74 (2008-A) and 1.48 (2009-A) redundant SCUs for each sentence. About 30%-37% of SCUs in the MWS summary are redundant.

The modified pyramid score is used as a measurement of recall for how many SCUs were retrieved by a summary. The recall of one of these summaries is the observed SCU weight of the summary, normalized by the average number of SCUs found in the four reference summaries [13].

Maximum SCU weight – MSW
As opposed to the international media hype that surrounded last week's flight, with hundreds of journalists on site to capture the historic moment, Airbus chose to conduct Wednesday's test more discreetly. The A380 will take over from the Boeing 747 as the biggest jet in the skies. So far, Airbus has 154 firm orders for the A380, 27 of them for the freighter version. March 2005: Scheduled first test flight of the plane. June 1994: Airbus begins engineering development of the plane, then known as the A3XX. Assembly of the plane itself is to take place in Toulouse, France.

Maximum unique SCU weight – MUSW
Most A380 traffic will go into just 25 of those airports, Dupont said. March 2005: Scheduled first test flight of the plane. January 23, 2002: Production starts of Airbus A380 components. The A380 will take over from the Boeing 747 as the biggest jet in the skies. Federal Express has ordered 10 of the planes. Assembly of the plane itself is to take place in Toulouse, France. The program, launched in December 2000, banks on a strategy of transporting huge numbers of passengers. International airport standards call for no plane to exceed 80 meters in length and width.

Fig. 3. Summaries generated for the query "Airbus A380 – Describe developments in the production and launch of the Airbus A380" for the 2008 document set D0801-A

$$Modified\_Pyramid\_Score = \frac{\sum_{i=1}^{n} i \times O_i}{\frac{1}{n}\left(\sum_{i=1}^{n} i \times |T_i|\right)}$$

In this formula, O_i is the number of observed SCUs of weight i, T_i represents the set of SCUs of weight i and $|T_i|$ is that set's cardinality. The number of reference summaries, 4 in these data sets, determines n, the maximum weight of a SCU. Table 3 shows the average values of the modified pyramid scores for the MSW and MUSW summaries as well as reference summaries and a random baseline. We have re-implemented the modified pyramid score, so these numbers are not directly comparable to those published by TAC. Jackknifing was used to try and ensure the fairest comparison possible. Note that for this kind of evaluation scores > 1.0 are possible. In fact, since the score is normalized by the combined SCU count from the reference summaries, not the combined SCU scores, the reference summaries will regularly have scores > 1.0. As a lower bound, we also present the results from a system which makes a random selection from the set of positive and negative sentences in the SCU corpus. This baseline is meant to replicate an average extractive summary submitted to TAC.

As can be seen, the scores for SCU-optimal summaries are very high, and scores for MUSW are slightly above those of the reference summaries. Given that we use an estimated one third of the positive sentences from the data set to generate these SCU-optimal summaries, automatic summaries have the potential to contain as much information as human summaries. The random baseline's low score shows how much more room for improvement there is in extractive summarization.

Table 3. Modified pyramid scores for MSW, MUSW, reference summaries and random baseline

	MSW	MUSW	Reference	Random
2008-A	1.06	1.31	1.30	0.39
2009-A	1.23	1.45	1.30	0.29

5 ROUGE

TAC evaluates its systems using two variations on the ROUGE metrics (Section 1.2). ROUGE-2 and ROUGE-SU4 are reported in [14]. Table 4 shows these measures for the MSW and MUSW summaries on the 2008-A and 2009-A data sets. Also in this table are the ranges of scores for the reference and peer summaries for the respective measurements. The recall for these measures for 2008 appears in [14], and the 2009 recall is for now available to the participants. We exclude the scores of some baseline systems also evaluated in TAC 2008 and 2009. For the 2008 data, we also give the upper bounds established in [7]; they are labelled *Max* in the table.

Table 4. ROUGE recall for SCU-optimal and reference/peer summaries

year	measure	MSW	MUSW	Reference Summaries	Peer Summaries	Max
2008-A	ROUGE-2	0.118	0.116	0.108 .. 0.131	0.039 .. 0.111	0.199
	ROUGE-SU4	0.150	0.151	0.140 .. 0.170	0.074 .. 0.143	0.219
2009-A	ROUGE-2	0.105	0.097	0.111 .. 0.149	0.028 .. 0.122	
	ROUGE-SU4	0.140	0.133	0.148 .. 0.184	0.059 .. 0.151	

Both MSW and MUSW fall within the range of the reference summary scores for the 2008 TAC data on ROUGE. For the 2009 data, the ROUGE scores were towards the higher end of the peer summary scores, but they were not quite as good as the reference summaries. ROUGE is really a heuristic method for estimating responsiveness in a summary; it does not directly evaluate content or readability. It certainly does not address redundancy in summaries, considering that the MSW summaries outperformed the MUSW most of the time. That said, these scores show that the SCU-optimal summaries can come quite close to reaching the quality of reference summaries, which would serve to confirm our findings in Section 4.

6 Manual Evaluation

Next we look at the readability and responsiveness of summaries. Unfortunately, the SCU-labelled corpus does not give us any method of determining how readable the summaries we generate are. Readability evaluation takes into account a mixture of grammaticality, non-redundancy, referential clarity, focus, and structure/coherence. Extractive text summarization has both strengths and weaknesses when it comes to readability. Sentences all come from original documents, so they are almost always grammatically correct. The MUSW summaries should do well for non-redundancy, because they contain little repetition, but the MSW summaries have no explicit redundancy checking. Referential clarity can be a problem for any extractive summarization system, if there is no attempt at co-reference resolution. Focus measures how relevant each sentences is to the rest of the summary, while structure and coherence measure whether the sentences are just a heap of information or whether they flow together well; we do not expect the SCU-optimal summaries to perform well on either of these measures, because the flow can even be broken between every two sentences.

Four volunteer annotators helped test the readability and responsiveness of the summaries generated. Each annotator rated 5 kinds of summaries for readability (and its 5 sub-criteria) and responsiveness on a scale 1..10. Two of the summaries were reference summaries generated for TAC, one of the summaries was MSW, one was MUSW and one was a random baseline summary (generated by randomly selecting labelled sentences from the SCU corpus). The probability of selecting a sentence was proportional to the number of peer summaries in which it appeared. This evaluation was done on summaries for 8 different randomly selected document sets (4 each from 2008 and 2009). Table 5 shows the average responsiveness and readability scores for each of these 5 kinds of summaries. We used code from [15] to calculate Krippendorff's α [16] with the interval distance metric to measure inter-annotator agreement. For responsiveness and overall readability we had $\alpha = 0.420$ and $\alpha = 0.459$ respectively.

Regrettably, our sample set is too small to prove conclusively one system's superiority over another, but there are a number of interesting observations which arise from this experiment. Table 5 shows that the human summaries scored better than others on all the measures, with scores between 7.6 and 9.0. The MSUW SCU-optimal summaries and the random baseline had similar performance when comparing overall readability but there was a noticeable difference in responsiveness. The most interesting results come when we look at the sub-criteria of readability. There was little difference between the grammaticality scores. For non-redundancy, the MUSW was not too far below the human summaries, and even the random baseline did not contain much redundancy. In terms of referential clarity, the SCU-optimal summaries and the random baseline differed quite a bit, and generally had scores much lower than the human summaries. The score for this measure may change noticeably by having just one or two additional unclear references. Focus and structure/coherence are measures on which the extractive summaries all performed poorly.

Here is what we can learn from all this: when it comes to readability, non-redundancy is the only sub-measure over which an extractive summarization system can really have influence. A co-reference resolution system might help

Table 5. Average responsiveness and readability scores for each system

Measure	Reference Set 1	Reference Set 2	MSW	MUSW	Random
Responsiveness	7.63	7.88	5.16	6.03	5.38
Readability	8.28	8.22	5.69	6.51	6.46
Grammaticality	*8.75*	*8.53*	*7.81*	*8.31*	*8.34*
Non-Redundancy	*8.65*	*9.00*	*6.69*	*7.91*	*7.53*
Referential Clarity	*8.84*	*8.50*	*6.75*	*6.44*	*7.22*
Focus	*8.09*	*8.28*	*5.34*	*6.00*	*5.75*
Structure/Coherence	*7.91*	*7.81*	*4.38*	*5.31*	*5.06*

improve referential clarity, but in terms of focus and structure/coherence it is difficult to see how these scores can be improved when we restrict ourselves to extractive summarization.

7 Conclusions

We have shown that it is possible to generate summaries which contain content comparable to human summaries, from the perspective of both pyramid and ROUGE evaluation. Despite the high scores on these two measure we found that the responsiveness of the SCU-optimal summaries was not as high as of the reference summaries. When evaluating readability of the summaries, we showed that grammatically of the SCU-optimal summaries is very close to human summaries, and there is the potential to nearly match human summaries in terms of non-redundancy. Other measures based on the coherence of the summaries, however, showed a wide gap between human-written summaries and the SCU-optimal summaries. In future work, we would like to compare our SCU-optimal summaries to other state-of-the-art extractive summaries in terms of readability, responsiveness and modified SCU scores.

Our final conclusion is that it is possible to generate extractive summaries which perform very well on automated measures such as ROUGE, or measures which follow a strict process, as pyramid evaluation does. Ultimately these summaries will not score as well when it comes to manual evaluation, because the readability tends to be low. It may be possible to improve on these SCU-optimal summaries with the addition of co-reference resolution, or perhaps some method of ordering sentences to make them more readable, but from the point of view of content these summaries are as good as can be generated extractively.

Acknowledgments

Our research is supported by the Natural Sciences and Engineering Research Council of Canada and the University of Ottawa. We thank Anna Kazantseva, Richard Keelan and Fazel Keshtkar for help with annotation, and Terry Copeck for maintaining the SCU-labelled corpus. We would also like to thank an anonymous reviewer for the insightful comments.

References

1. Lin, C.Y., Hovy, E.: Identifying topics by position. In: Proc. 5th Conference on Applied Natural Language Processing, Morristown, NJ, USA, pp. 283–290. ACL (1997)
2. Katragadda, R., Pingali, P., Varma, V.: Sentence position revisited: a robust light-weight update summarization 'baseline' algorithm. In: Proc. Third International Workshop on Cross Lingual Information Access, Morristown, NJ, USA, pp. 46–52. ACL (2009)
3. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Proc. ACL workshop on Text Summarization Branches Out, p. 10 (2004)

4. Copeck, T., Szpakowicz, S.: Leveraging pyramids. In: HLT/EMNLP - Document Understanding Workshop, DUC (2005)
5. Genest, P.É., Lapalme, G., Yousfi-Monod, M.: Hextac: the creation of a manual extractive run. In: TAC 2009 Notebook, Gaithersburg, Maryland, USA (November 2009)
6. Nenkova, A., Passonneau, R.J.: Evaluating content selection in summarization: The pyramid method. In: HLT-NAACL, pp. 145–152 (2004)
7. Gillick, D., Favre, B., Hakkani-Tur, D.: The ICSI Summarization System at TAC 2008. In: Proc. of the Text Analysis Conference workshop, Gaithersburg, MD, USA (2008)
8. Hovy, E., Lin, C.Y., Zhou, L., Fukumoto, J.: Automated Summarization Evaluation with Basic Elements. In: Proc. 5th International Conference on Language Resources and Evaluation (LREC), pp. 899–902 (2006)
9. Copeck, T., Inkpen, D., Kazantseva, A., Kennedy, A., Kipp, D., Nastase, V., Szpakowicz, S.: Leveraging duc. In: HLT-NAACL 2006 - Document Understanding Workshop, DUC (2006)
10. Nastase, V., Szpakowicz, S.: A study of two graph algorithms in topic-driven summarization. In: TextGraphs 2006: Proc. TextGraphs: the First Workshop on Graph Based Methods for Natural Language Processing, Morristown, NJ, USA, pp. 29–32. Association for Computational Linguistics (2006)
11. Fuentes, M., Alfonseca, E., Rodríguez, H.: Support vector machines for query-focused summarization trained and evaluated on pyramid data. In: Proc. 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, Morristown, NJ, USA, pp. 57–60. ACL (2007)
12. Katragadda, R., Varma, V.: Query-focused summaries or query-biased summaries? In: Proc. ACL-IJCNLP 2009 Conference Short Papers, Suntec, Singapore, August 2009, pp. 105–108. Association for Computational Linguistics (2009)
13. Passonneau, R.J.: Formal and functional assessment of the pyramid method for summary content evaluation. Natural Language Engineering, 1–25 (2009)
14. Dang, H.T., Owczarzak, K.: Overview of the tac 2008 update summarization task. In: Proc. Text Analysis Conference, pp. 1–16 (2008)
15. Artstein, R., Poesio, M.: Inter-coder agreement for computational linguistics. Comput. Linguist. 34(4), 555–596 (2008)
16. Krippendorff, K.: Content Analysis: An Introduction to Its Methodology. Sage, Thousand Oaks (2004)