Multi-Document Summarization from First Principles

William M. Darling
School of Computer Science
University of Guelph
Guelph, Canada
wdarling@uoguelph.ca

***Abstract*—We present *SumBasic+*, a powerful multi-document summarization system built from first principles. *SumBasic+* is designed as a baseline system to gauge the level of summarization results we could obtain using simple statistical techniques. Our extractive summarization system is based on word frequency statistics similar to the *SumBasic* method. Nevertheless, we were able to considerably improve its summarization performance by tuning the amount and type of redundancy removal performed, adding a simple query-focused summarization component, and by employing a number of pre- and post-processing compression techniques. The resulting system, *SumBasic+*, is a strong baseline system that is ideal for comparing with new summarization approaches, as it principally uses existing techniques and performs surprisingly well. Of 43 competing systems in the TAC 2010 summarization track, our system achieved fourth and third place in R-2 and R-SU4 ROUGE scores respectively, and second overall in the manual average pyramid evaluation for the initial summaries.**

***Keywords*-Text processing; Artificial intelligence**

I. Introduction

The summarization track of the Text Analysis Conference (TAC) affords participants a chance to compare their summarization systems to other research institutes' state-of-the-art work. While NLP scholars who engage in summarization research are fortunate enough to have access to a robust automatic summary responsiveness metric in the ROUGE toolkit ([1]), the qualitative aspect of a summary can best be determined by a human judge – something that the TAC provides. Along with the agglomeration and licensing of datasets performed by NIST, and the supplying of model summaries, the TAC is an excellent forum to test and report on new breakthroughs in summarization research. It is also, however, an ideal venue to determine what can be achieved with existing and well-known summarization techniques.

Our goal in participating in this year's summarization track was to develop a strong baseline summarization system that will be used in future years to determine our progress on more advanced summarization techniques. We took an existing system, *SumBasic* (described in [2]), based on Luhn's 50-year-old word frequency-based summarization ideas highlighted in [3], and tuned and expanded it. We performed numerous experiments using previous years' datasets to tune the degree of smoothing and redundancy removal. We also implemented simple techniques that improved our average ROUGE scores (and some that did not) including sentence compression, n-gram language models, named entity recognition (NER) methods, and others. Each of the paths that we explored will be described along with data on how our ROUGE results were affected.

The rest of the paper is organized as follows. The next section will discuss related work, focusing on the word frequency-based summarization methods in *SumBasic*. Our system, *SumBasic+*, will be described in section 3, including the core details on building the probability distribution, the composition function and sentence selection. Section 4 will discuss further extrinsic details of *SumBasic+* that improve its performance including sentence boundary detection, query-based term weighting, redundancy removal techniques, sentence compression, and sentence ordering. This section also includes a description of our experimental design in testing each of these additions. Our results in the TAC 2010 summarization track are presented in section 5, which is followed by a discussion of future work in section 6. Section 7 concludes.

II. Related Work

Our summarization system borrows heavily from the work of Nenkova, et al. in [2]. In the work described therein, Nenkova, et al. studied human summaries and empirically determined that words that appear with high frequency in a document set appear with very high probability in the related summary. It was shown that, in general, the more frequently a word appears in the original text, the more likely it is that it will appear in a human generated summary. Following these findings, Nenkova, et al. defined a family of summarization systems, SUM_{CF}, where each member of the family uses the same method to determine the frequencies of words, but uses a different composition function to determine the best sentences to select for extractive summarization.

For each summarizer $s \in SUM_{CF}$, the unigram probability distribution for all words appearing in the input (ignoring stopwords) is computed such that for each word w, $p(w) = \frac{n_w}{N}$ where n_w is the number of times the word w appears in the input and N is the total number of word tokens. Each sentence is then given an "importance" score based on a function CF with the unigram probability distribution for the words in that sentence, $\mathbf{w} = w_1 w_2 ... w_m, w_i \in S_n$, as an input:

$$Score(S_n) = CF(p(\mathbf{w})) \tag{1}$$

The algorithm then iteratively picks the highest scoring sentences to include in the summary until the desired length l is reached.

Nenkova, et al. considered three candidate CF functions: CF_Π multiplies the probabilities of the sentence's words together; CF_{Avg} adds the probabilities together and divides by the number of word tokens in the sentence; and CF_Σ simply adds the sentence's word probabilities together to compute the importance score. It is immediately apparent that CF_Π will favor short sentences, CF_Σ will favor longer sentences, and CF_{Avg} should be somewhere in between. Finally, Nenkova, et al. also consider an additional step in the algorithm to reduce redundancy. After a sentence has been selected for inclusion, the probabilities for the words in that sentence are reduced to 1.0×10^{-4} (a number close to, but not equal to, 0) to discourage sentences with similar information from being chosen again. In [4], Haghighi and Vanderwende modify this update such that the words in a selected sentence are updated as $p'(w) = p(w)^2$ allowing the probabilities of words to follow a log reduction in unigram probability as they are chosen. The resulting simple frequency-based summarizers perform extremely well, with SUM_Σ and SUM_{Avg} achieving better than average results at the DUC 2004 summarization competition.

III. SumBasic+

To build *SumBasic+*, we began with the *SumBasic* system described in [2] and followed an iterative approach of adding and removing features and tuning parameters with the goal of maximizing ROUGE metric results on the TAC 2009 dataset. Mathematically, we aimed to maximize the following equation.

$$SumBasic{+} = \arg\max_{\theta \in \Theta} ROUGE(\theta) \tag{2}$$

where Θ contains all possible permutations of statistical summarization features and tuning parameters that were reasonably available to us. Using this method, we were able to increase *SumBasic* ROUGE scores on the TAC 2009 dataset from 0.092 and 0.134 for R-2 and R-SU4 respectively, up to 0.116 and 0.158 with the final version of *SumBasic+*, for increases of 26% and 18% beyond the original system on the initial summary creation task.[1] For the update summary task, we were able to improve the system even further. ROUGE scores increased from 0.073 and 0.123 to 0.099 and 0.147, for R-2 and R-SU4 respectively. These improvements represent increases of 35% and 19%. While these increases appear to be more significant than for the initial summary task, we should note that *SumBasic* has no knowledge of the update summary task and is therefore at a disadvantage in that comparison.

Each of the possible techniques and methods of implementation that comprise the possible summarization models in Θ are described in the following subsections. These include building the document set probability distribution, determining the best sentence score composition function, selecting sentences for extraction, query-based term weighting, sentence compression, redundancy removal, and sentence ordering.

A. Probability Distribution

At the heart of the *SumBasic+* summarization system is the probability distribution describing the word frequency statistics in the document set to be summarized. Nenkova, et al. follow a simple unigram probability distribution in [2] where $p(w) = \frac{n_w}{N}$. Our first experiment in attempting to improve this model was to take bigram frequency statistics into account. We updated $p(\mathbf{w})$ as follows: for each $w_i \in \mathbf{w}$,

$$p(w_i|w_{i-1}) = \lambda p_U(w_i) + (1-\lambda)p_B(w_i|w_{i-1}) \tag{3}$$

where $p_U(w_i)$ is the document set unigram probability for w_i, $p_B(w_i|w_{i-1})$ is the document set bigram probability for w_i given w_{i-1}, and $\lambda \in [0,1]$ is a parameter that controls how much of each distribution contributes to the overall probability $p(w_i|w_{i-1})$. We let λ vary from 0 to 1 in 0.05 increments and recorded the relative effect on ROUGE R-2 and R-SU4 scores. Unfortunately, as $\lambda \rightarrow 0$, ROUGE results monotonically decreased. In other words, the more effect the bigram statistics had on the distribution, the worse our summarizer performed. Due to these experiments, only unigram probability information was used in *SumBasic+*.

The reason for the poor ROUGE results when using bigram statistics is likely a data sparsity issue. While for most document sets there were typically some bigrams with very high probability (one example is "Columbine High" in the document set describing the shooting at that school), besides these few common bigrams, the rest of the bigrams are essentially noise in that their existence is coincidental and only occurs very rarely. One potential avenue worth exploring is allowing λ to start off low and after a few sentences have been chosen, let it increase towards 1.

The next aspect of modeling the document set's probability distribution that we explored was the dynamic updates of word probabilities as sentences are chosen to be included in the summary. In [2], when a sentence S is selected for inclusion in the output, the probability $p(w)$ is updated as $p'(w) = 1.0 \times 10^{-4}$ for each $w \in S$. In [4], Haghighi and Vanderwende update the distribution as $p'(w) = p(w)^2$. We found that both approaches perform too much redundancy removal when the goal is to maximize ROUGE scores. A number of linear functions were considered but by far the most successful was reducing the word probabilities by a simple constant. In *SumBasic+*, when a word w appears in a sentence that is to be included in the summary, we set $p'(w) = \frac{p(w)}{2}$. This update allows common words to contribute to the scoring function for a longer period of time before they are forced towards irrelevance by the redundancy removal step. This change, which represents a very small but central modification to *SumBasic*, resulted in an 14% increase in the R-2 ROUGE

[1] Note that these comparisons are with our implementation of *SumBasic*.

metric, and a 12% increase in the R-SU4 ROUGE metric on the TAC 2009 dataset initial summaries.

B. Composition Function

In [2], SUM_{Avg} results in the most powerful summarization system amongst the three considered composition functions. Nevertheless, for completeness we tested each of the composition functions on the TAC 2009 data. Like the results described in [2], SUM_{Π} performed the worst as words with low probability draw the scoring function down too much and the system favors short sentences much too heavily.[2] However, in our experiments it was SUM_{Σ}, not SUM_{Avg}, that performed best. While using the CF_{Σ} composition function does give preference to longer sentences, these long sentences are often strong candidates for summary inclusion due to their high semantic content.

C. Sentence Selection

While the composition function described above is the means for which *SumBasic+* chooses the best sentence for inclusion in a summary, there are a couple of other more nuanced issues that can play a part in sentence selection. One approach that is mentioned in [2] is requiring the selected sentence to contain the current most probable word in the unigram probability distribution. This is of course a shifting requirement as the most probable word will typically change after each sentence is selected due to the redundancy removal probability update step. We tried this technique with *SumBasic+* but found that in some cases there were no sentences remaining that contained the given word. Therefore, we relaxed this requirement such that if there was a remaining sentence containing the word with the current highest unigram probability, we choose that sentence. Otherwise, we choose the sentence with the highest score.

Another issue to consider in sentence selection is linguistic quality and coherence when choosing a final sentence. Because we have the limitation that for any summary S, $len(S) \leq 100$, when choosing the final sentence we could opt for the highest scoring sentence that fits within the word limit. Another option is simply adding the highest scoring sentence and then truncating the summary down to fit the word limit. Generally, the former option will do better on readability and coherence while the latter, depending on how many words are left after truncation, will do better on semantic responsiveness and therefore ROUGE score as every extra word that is squeezed in – as long as it appears in a model summary – should help the ROUGE score. In our experiments, this theory proved to hold true and following the add-and-truncate method led to the best ROUGE results (and was therefore used in *SumBasic+*. Note that if one were to follow this logic to its conclusion, one way to potentially increase ROUGE results even further would be to truncate the final sentence by iteratively removing the words with the lowest probability in $p(\mathbf{w})$ until $len(S) = 100$. This would lead to even poorer readability, but could fit more content words in to help maximize ROUGE scores.

IV. SumBasic+ Further Details

This section describes a number of summary performance improvements that were included in *SumBasic+* but that are not central to its functioning. Many of these approaches could be applied to any summarization approach to help build a more concise and accurate output summary.

A. Sentence Boundary Detection

One aspect of summarization that many papers take for granted (or at least fail to discuss) is the segmentation of a document into a list of sentences. As noted in [5], sentence boundary detection (SBT) is used widely in NLP, but is often done so with outdated tools. Because of the ambiguity of punctuation – principally the *period* – it is not always clear where one sentence ends and another begins. For simplicity, and because we use the python *NLTK* package for other features in our implementation, we use an unsupervised algorithm included in that library described in [6]. This approach has an error rate of 1.65% on the *Wall Street Journal* dataset, which is enough to give us incorrect sentence boundary information in more than 1 out of every 10 ten-sentence summaries. Another approach we tried, and which we submitted as our primary run in the TAC 2010 summarization track (our secondary run relied solely on the *NLTK* sentence segmenter), was to use the paragraph segmenting included in most XML-based newswire articles. This, surprisingly, resulted in an increase of average ROUGE scores on the TAC 2009 dataset. In general, the $< p >$ tags in the newswire data separate sentences as these news articles are typically written in a style such that each paragraph comprises a single sentence. We put in some logic that allowed *SumBasic+* to fall back on the *NLTK* segmentation model if the paragraph information was missing or if the sentences it provided were longer than some arbitrary maximum length.

B. Query-Based Term Weighting

While the TAC 2010 summarization task did not strictly require a query-focused method, each document set contained a short "title" describing the contents of the enclosed documents.[3] The document set title (examples include "Eating Disorders" and "Rain Forest Destruction") can be thought of as the search term that was used to collect the relevant articles contained in the document set. We implemented a simple "query"-based term weighting system where more probability mass would be put on a word in the unigram probability distribution if that word was contained in the document set title. We call this parameter γ_Q and adjust the probabilities as follows. If w is contained in the document set title,

$$p(w) = \frac{n_w + \gamma_Q}{N + \zeta} \tag{4}$$

[2] Note that we implemented SUM_{Π} using log probabilities to avoid problems with underflow.

[3] The shortest title was one word long while the longest title contained four words. The average title word length was approximately 2.4.

TABLE I
PRE-PROCESSING SENTENCE COMPRESSION

Original Sentence	Compressed Sentence
The mortars came from across the border Tuesday evening and landed in India's Durga Post area in the Poonch sector, an Indian police spokesman said.	The mortars came from across the border Tuesday evening and landed in India's Durga Post area in the Poonch sector.
At the heart of the rebuilding is the creation of a lasting memorial which will honor the memories of those we lost and help tell their story to the world, said New York Governor George Pataki.	At the heart of the rebuilding is the creation of a lasting memorial which will honor the memories of those we lost and help tell their story to the world.

where n_w and N are as before, and ζ is a normalization constant that ensures that all probabilities sum to 1. To determine the ideal value for γ_Q we followed equation (2) and tried a number of values between 0 and 5. We empirically determined that on the TAC 2009 data, the best initial summary ROUGE results are obtained with a value of $\gamma_Q \approx 1.5$ and $\gamma_Q \approx 1.0$ for the update summaries. Using our query-based term weighting improved ROUGE scores for both initial and update summaries. More specifically, employing this version of query-based term weighting in *SumBasic+* resulted in a 5% increase in both R-2 and R-SU4 scores for the initial summaries, and a 20% increase in R-2 scores and a 10% increase in R-SU4 scores for the update summaries over our base implementation of the *SumBasic* system.

C. Sentence Compression

Because there is a limit of 100 words per summary, one of our goals in summarization is to be as concise as possible. We are limited by the fact that our method works by extracting existing sentences, but we can introduce conciseness by attempting to remove redundant or unneeded information from a sentence. *SumBasic+* includes both pre- and post-processing sentence compression steps. Each will be examined in turn.

1) Pre-Processing: To maximize the ROUGE scores obtainable by *SumBasic+*, an analysis of commonly occurring patterns in both the input documents and output summaries was performed. In the newswire domain, which the TAC summarization data consists of, it is extremely common for a sentence to end with a phrase of the form "..., an official said." This type of phrase is quite common and adds important credibility information to a new story, but offers little in terms of semantic content for a summary. Further, because the effect on word frequency statistics for this kind of phrase should have little effect on the sentences we choose for inclusion in a summary, we chose to compress all sentences containing this form of phrase by finishing the sentence before that clause begins. For example, this pre-processing step performs the conversions shown in Table I. While it is noted that this could have the potential of removing important context information, in a statistical summarization system where the output is limited to 100 words, this pre-processing step improves our average ROUGE scores and allows more content to be fit in the constrained space of the output. Other pre-processing compression steps we took included removing the words "but" and "however" when they appeared at the beginning of a sentence, and removing the word "also" in all contexts. This last step generally allows us to fit more content in less space and has a very low effect on readability.[4]

In addition to the "content"-based pre-processing steps described above, there are a number of other pre-processing steps we took that led to improved ROUGE results on our test dataset. While both of these methods in some cases may lose some important information from the input, they both improved average ROUGE scores and were therefore included in the *SumBasic+* system. First, if a sentence contains any information enclosed with brackets, that information was removed. This idea is based on the theory that parentheses typically contain superfluous information that may be of interest to the reader but that is not necessary for understanding the content of a story. Another, arguably more drastic, compression technique that was also investigated and ultimately employed was removing any sentence that began with a quotation mark. Our rational for taking this step was that quotes in news articles often add opinion as opposed to objective information to the story. A news article will rarely convey important semantic information through a direct quotation. Taking this step, we were successful in again improving our ROUGE R-2 and R-SU4 results.

2) Post-Processing: In addition to performing sentence compression before our probability distribution is computed, there is also room for post-processing compression which will in many cases allow more content words to be fit into the 100-word summary. These compression steps are done following the probability distribution calculations because the words removed or modified here may contribute positively to our sentence selection scoring function. In our post-processing sentence compression experiments, we considered a number of abbreviations and word transformations that would shorten the length of a sentence while refraining from removing any content or diminishing readability. Some of these approaches included transforming dates, such as names of days of the week and months, to short forms. Examples include "February 1, 2010" becoming "02/01/2010", and transforming numbers spelled out into their Arabic numeral form ("one hundred thousand" becomes $100,000$, etc.). Most of these transformations will have only a negligeble effect on ROUGE scores as at most they will allow space for an extra word or two in the output summary. It is likely that the other modifications could lead to more concise and therefore more content-heavy summaries, but the ROUGE scores decreased due to the lack of direct unigram and bigram matches between the model and peer summaries. For example, while the sentence "The TSX lost 12 % or 100 points of its value last Tuesday." conveys the same amount of semantic information as "The TSX lost 12 percent or one hundred points of its value last Tuesday, December

[4] As an example, consider the following sentence: "The peace process has *also* led to a resumption of train service". Removing the word "also" has no effect on the semantic content in the sentence but reduces its length by 5 characters (including the additional required whitespace).

14" in a more concise manner, the ROUGE score will be lower if the compressed sentence is used because the unigram and bigram statistics used in computing ROUGE scores will not match for the modified words. This is perhaps an area of future research in developing more advanced automatic summarization metrics.

D. Redundancy Removal

While redundancy removal in the core *SumBasic+* system was discussed in a previous section, here we discuss our experiments with implementing redundancy removal for the update summaries. The idea behind the update summaries is that a user has read an initial summary and would like the update summary to only contain information that is novel, or different from the initial summary. To perform this task well, it is likely that advanced techniques will be required. Nevertheless, we continued to experiment with simple and existing statistical techniques. The update summary component of *SumBasic+* receives both the document set to be updated, and the initial summary, as inputs. It then uses the cosine similarity metric to ensure that none of the selected sentences are too alike the sentences that were included in the initial summary.

The first update summary redundancy removal technique that we investigated was using the left over unigram probability distribution $p(\mathbf{w})$ after summarizing the initial document set and applying the word probability updates. Unfortunately, in nearly every case this resulted in too much redundancy removal and the selected sentences were random at best. Instead, when selecting a sentence we compared its vector space representation to that of all sentences selected thus far and the sentences in the initial summary and ensured that it was lower than a given threshold value τ. In our experiments, the best ROUGE results were achieved when τ was set to be very high; $\tau \approx 0.994$ was ideal. This resulted in removing very few (but some) sentences and helped to register a modest increase in update summary ROUGE scores.

The fact that ignoring redundancy removal for all but the most similar sentences resulted in the best ROUGE results with *SumBasic+* shows us that though these simple statistical techniques can result in powerful summarization systems, they are not without their limits. Both initial and update summaries should include sentences with the most common words in the document set because that is what the articles are describing. The update summaries should differ not in the type of content words that appear, but in how those content words are used. A powerful update summarization system will need to learn more about what certain words are conveying with the help of other surrounding words, as opposed to simply looking for sentences with less frequent content words.

E. Sentence Ordering

In multi-document summarization, a problem that does not exist (or at least is not as severe) in single-document summarization is sentence ordering. In single-document summarization, the extracted sentences are pasted together in the output in the same order as they appeared in the input. In multi-document summarization, where extracted sentences come from several documents of different lengths, there is no coherent concept of sentence order. While there are multiple potential solutions to this issue, including powerful supervised techniques such as those found in [7], we followed a statistical heuristic that seemed to offer acceptable results. Each extracted sentence is given an index number $i \in [0, 1]$ where $i = \frac{l}{N_S - 1}$, $l \in \{0, ..., N_S - 1\}$ is the location of the sentence in its document, and N_S is the number of sentences in the document. Selected sentences are then placed in the output summary in order of lowest to highest index number, and ties are broken arbitrarily.

TABLE II
ROUGE AND BE RESULTS

	R-2	R-SU4	BE
Initial	0.09196 (4th)	0.12829 (3rd)	0.05349 (6th)
Update	0.06663 (15th)	0.10953 (7th)	0.03564 (19th)

V. RESULTS

In this section we present our ROUGE, Basic Elements (BE), and manual results in the TAC 2010 summarization track competition. Note that while we submitted two runs for *SumBasic+* – each using a different sentence boundary detection algorithm – we only report results for our principal submission (though generally they differed only slightly). *SumBasic+* was built following an iterative approach where if a method or implementation decision increased the average ROUGE score on the TAC 2009 dataset, it would be included.[5] For that reason, we expected to do well in the automatic evaluation approaches, and perhaps not as well in the manual evaluation. More specifically, we would also expect to do better in the ROUGE evaluation than with the BE evaluation as we took no steps to maximize BE scores during the creation of our system. The actual results are fairly faithful to that viewpoint but we were also pleasantly surprised at how well *SumBasic+* performed on the manual evaluation as well. We also expected to do much better with the initial summaries than with the update summaries as the simple methods that we implemented were not designed to perform the specific nuanced level of redundancy removal that was required of that task.

Our automatic ROUGE and BE results are shown in Table II. Again, our expectations were roughly in line with our results as we did very well for initial summaries, recording the fourth, third, and sixth best initial summary results for R-2, R-SU4, and BE metrics, respectively. Our update summary results, also as expected, came in lower than our initial summary results but were still well in the top half of all submissions achieving the fifteenth, seventh, and nineteenth best scores respectively. Our BE scores are in line with our ROUGE scores but in both cases a little bit worse which

[5] We did, however, include some features that were included principally for creating a better summary linguistically and that would have no effect on ROUGE score such as the sentence ordering component described in section III.H.

TABLE III
MANUAL RESULTS

	Responsiveness	Linguistic Quality	Avg Pyramid Score
Initial	3.065 (4th)	3.22 (10th)	**0.41 (2nd)**
Update	2.391 (11th)	3.28 (4th)	0.22 (23rd)

is again not surprising as we tuned our system to excel at ROUGE performance. In all cases but one *SumBasic*+ also outperformed the MEAD automatic summarizer which was used as a baseline for the summarization track.[6]

For the manual results, our expectations were more subdued than with the ROUGE results as our approach to building the system was aligned with maximizing an automatic metric. However, our results give credence to ROUGE's acceptability as an automatic summarization metric as *SumBasic*+ performed very well in the manual evaluations as well. As shown in Table III, we earned the fourth best responsiveness score and the tenth best linguistic quality score for the initial summaries, and the eleventh best responsiveness score and fourth best linguistic quality score for the update summaries. Our best result, however, is the average pyramid score where we obtained an 0.41, good enough for a tie with the second best system at TAC 2010. These results enforce the power of simple statistical techniques and the integrity of ROUGE as an accurate automatic metric for evaluating document summaries.

VI. Future Work

As TAC 2010 marked our first entry in the summarization track competition, we worked on building a strong baseline system that achieved good results using simple statistical techniques. Now that we have a strong foundation in the many extrinsic requirements of a strong summarization system, we plan to move into building more powerful probability distributions describing the input document set. Following [4] and [8], we will use probabilistic generative modeling techniques to build $p(\mathbf{w})$ in a way that puts more probability mass on the important content words that should appear in a document set summary.

VII. Conclusion

This paper has described *SumBasic*+, a powerful baseline statistical summarization system built from first principles and iteratively developed to achieve high ROUGE scores on newswire data. Despite its simplicity and its reliance on well-known and easy to implement methods, *SumBasic*+ received R-2 and R-SU4 ROUGE scores for the initial summaries that were statistically no different from the best performing systems at the TAC 2010 summarization track. Perhaps more impressively, it also achieved the second best manual average pyramid score for responsiveness and linguistic quality for initial summaries. While *SumBasic*+ did not score as highly in the update summary competition, its ROUGE, BE, and manual scores were still all in the top half of all submitted systems. These results show two important things. First, simple statistical techniques can do a very good job at automatic summarization. And second, that doing automatic summarization well – that is, statistically significantly better than an established baseline – is difficult. While all of the methods and tuning that we described are based on simple techniques, together they create a very strong baseline that is perfect for future summarization methods to be compared to, as straightforward techniques for each part of an ideal summarization system are all implemented and tested.

Acknowledgment

The author would like to thank his Ph.D. advisor Fei Song for his ideas and encouragement, and Lucy Vanderwende for taking the time to respond to several summarization related queries.

References

[1] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, S. S. Marie-Francine Moens, Ed. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 74–81.

[2] A. Nenkova, L. Vanderwende, and K. McKeown, "A compositional context sensitive multi-document summarizer: exploring the factors that influence summarization," in *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. New York, NY, USA: ACM, 2006, pp. 573–580.

[3] H. P. Luhn, "The automatic creation of literature abstracts," *IBM J. Res. Dev.*, vol. 2, no. 2, pp. 159–165, 1958.

[4] A. Haghighi and L. Vanderwende, "Exploring content models for multi-document summarization," in *NAACL '09: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 2009, pp. 362–370.

[5] D. Gillick, "Sentence boundary detection and the problem with the u.s." in *NAACL '09: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*. Morristown, NJ, USA: Association for Computational Linguistics, 2009, pp. 241–244.

[6] T. Kiss and J. Strunk, "Unsupervised multilingual sentence boundary detection," *Comput. Linguist.*, vol. 32, no. 4, pp. 485–525, 2006.

[7] R. Barzilay and L. Lee, "Catching the drift: Probabilistic content models, with applications to generation and summarization," in *HLT-NAACL 2004: Proceedings of the Main Conference*, 2004, pp. 113–120, best paper award.

[8] H. Daumé, III and D. Marcu, "Bayesian query-focused summarization," in *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 2006, pp. 305–312.

[6] For more information on MEAD, see http://www.summarization.com/mead/.