Algorithm Parallelism for Improved Extractive Summarization

Arturo N. Villanueva, Jr.
Department of Systems Engineering
Colorado State University
Fort Collins, CO, USA
art.villanueva@colostate.edu

Steven J. Simske
Department of Systems Engineering
Colorado State University
Fort Collins, CO, USA
steve.simske@colostate.edu

ABSTRACT

While much work on abstractive summarization has been conducted in recent years, including state-of-the-art summarizations from GPT-4, extractive summarization's lossless nature continues to provide advantages, preserving the style and often key phrases of the original text as meant by the author. Libraries for extractive summarization abound, with a wide range of efficacy. Some do not perform much better or perform even worse than random sampling of sentences extracted from the original text. This study breathes new life to using classical algorithms by proposing parallelism through an implementation of a second order meta-algorithm in the form of the Tessellation and Recombination with Expert Decisioner (T&R) pattern, taking advantage of the abundance of already-existing algorithms and dissociating their individual performance from the implementer's biases. Resulting summaries obtained using T&R are better than any of the component algorithms.

CCS CONCEPTS

• [Computing methodologies]: Artificial intelligence – Natural language processing • [Computing methodologies]: Machine learning – Machine learning algorithms • [Computing methodologies]: Parallel computing methodologies – Parallel algorithms

KEYWORDS

Natural Language Processing, Extractive Summarization, Meta-algorithmics, Machine Learning, Document Summarization, Tessellation and Recombination

ACM Reference format:

Arturo N. Villanueva, Jr, and Steven J. Simske. 2023. Algorithm Parallelism for Improved Extractive Summarization. In *ACM Symposium on Document Engineering 2023 (DocEng '23), August 22–25, 2023. Limerick, Ireland, 4 pages.* https://doi.org/10.1145/3573128.3609350

DocEng '23, August 22–25, 2023, Limerick, Ireland

ACM ISBN 979-8-4007-0027-9/23/08.
https://doi.org/10.1145/3573128.3609350

1 Introduction

Text summarization can be either extractive or abstractive. While extractive summarization picks verbatim, representative sentences or phrases in the text in context, abstractive summarization attempts to generate novel sentences that do not necessarily exist in the text. Extractive summarization results in a subset of existing sentences or phrases, while abstractive summarization does not guarantee such a set since the generated sentences are not directly extracted from the sample. It is not unusual that words that do not exist in the text appear in the summary. For better or for worse, extractive methods somewhat preserve the author's style while abstractive summarization does not necessarily do so. Simske and Vans [16] characterize extractive summarization as lossless and abstractive summarization as lossy in reference to the compression that either of the methods perform.

The objective of this research was to introduce parallelism through the use of a second-order meta-algorithm referred to as *Tessellation and Recombination with Expert Decisioner* [15] to see if existing methods could be combined to produce better results compared to their individual outcomes. While [6] argues that extractive summarization has mostly given way to abstractive summarization, the authors believe that meta-algorithmic techniques could be used to continue making advances in the field and furthermore, be bases for use with new algorithms for both extractive and abstractive summarization, and even a hybrid approach wherein both techniques are combined to get even better results. Additional benefits for extractive summarization include the retainment of key words and phrases suitable for indexing and preservation of query behavior [16,18].

2 Research Goals

The overarching goal of this research was to investigate how meta-algorithmic techniques could be used to improve existing as well as future component algorithms. In doing so, metrics for measuring the appropriateness of the generated summaries were also compared.

3 Process / Tasks

We executed the following process tasks:

3.1 Data Set Collection

The data used for this study was the Cable News Network (CNN) set used in [18] and first introduced in [4], a set of 3,000 English language articles from early 2015 spanning business, health, justice, living, opinion, politics, showbiz, sports, technology, travel, US, and world news. Articles contained anywhere from 10 to 197 sentences with a mean of 38.4. The "Gold Standard" summaries ranged from 4 to 13 sentences with a mean of 7.2.

The data set had been prepared by [4] in XML format which has the advantage of having many parsers available, including ElementTree [11], BeautifulSoup [12], and the standard XML Document Object Model (DOM) API [13]. We opted to use BeautifulSoup because of its maturity, having been available since 2004 [14].

3.2 Preparation and Initial Processing

The process of preparing and processing the articles is illustrated in Figure 1, taking advantage of the results of seven extractive summarizer algorithms prepackaged in NLTK's Sumy as well as an algorithm that simply picks random sentences from the article. The seven algorithms are described below:

Basic, or SumBasic [8], is the simplest of the algorithms we used, simply based on the premise that frequently occurring words are given more weight than those words that are not, and therefore drive the algorithm to pick sentences that have those heavily weighted words.

LexRank [3] is a graph algorithm that relies on the similarity of sentences with others. The idea is that a sentence that has a lot of similarity to the other sentences weigh more than sentences that do not have much similarity with other sentences.

The Luhn algorithm [17] is a heuristic extractive summarization algorithm with its roots in TF*IDF (Term Frequency-Inverse Document Frequency) and, while similar to SumBasic, discounts words that are too frequent (stop words).

LSA [5], short for Latent Semantic Analysis, is based on the idea that words that are more closely related to each other (such as *foot* and *shoe*) carry a higher weight than pairs that are remotely related (such as *foot* and *book*) and therefore semantically more relevant.

The TextRank [7] algorithm is similar to the PageRank algorithm use by Google and other search engines, with the difference that instead of web pages, TextRank ranks sentences based on similarities.

Edmundson [2] is an algorithm that weighs sentences using word position, word frequency, usage of cue words (e.g., superlatives), and document structure (titles, sub-titles, etc.)

KL, or the Kullback–Leibler algorithm [1], picks sentences based on entropy and can be very computationally heavy.

3.3 Tessellation and Recombination with Expert Decisioner

The Tessellation and Recombination with Expert Decisioner pattern (T&R), is a second-order meta-algorithmic pattern that first utilizes multiple generators (for this research, the algorithms identified above), tessellates the results, and recombines these tessellations into the desired results.

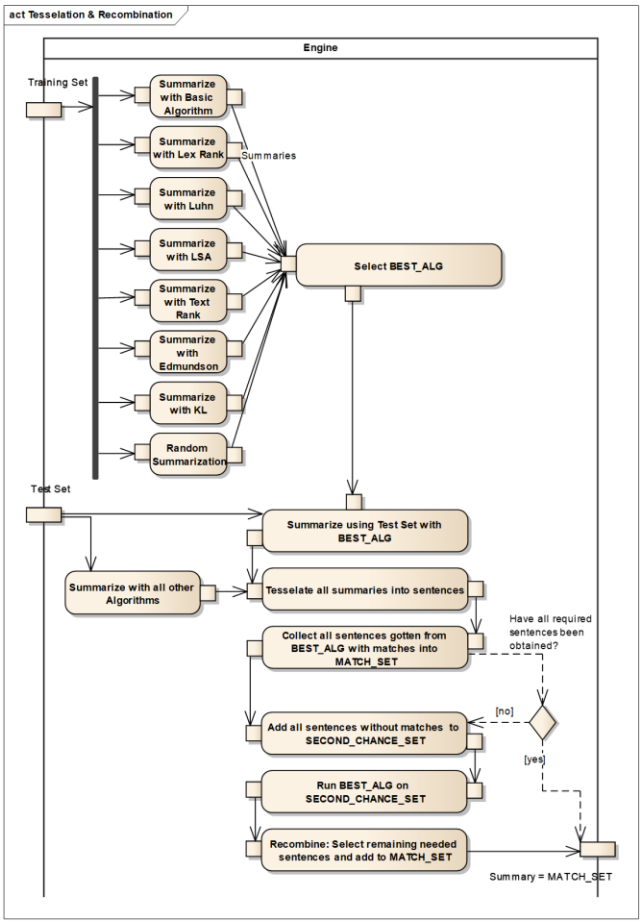

Figure 1. T&R meta-algorithm applied to the extractive summarization task

Prior to tessellation, a *best component algorithm* (BEST_ALG) is chosen using the training set. This BEST_ALG becomes the basis by which the other (non-best) algorithms are compared and later used for the second pass at summarization. The test set is then evaluated with each of the algorithms and the summaries are tessellated by breaking them down (except for the random summary) into their constituent sentences. Because we are extracting sentences from the articles, there is a good probability that the sentences extracted (rather than abstracted) using each of the algorithms have an overlap with each other. In fact, using BEST_ALG as the primary algorithm, we measured a mean overlap of 85%, not including random sentence selection. We took advantage of this and used those overlapping sentences as part of the final summary.

For summaries that have a 100% overlap, the pattern is complete, and we are left with the desired summary. For those that have an overlap less than 100%, we collect all the sentences that were left over (those that did not match with sentences

extracted by BEST_ALG) and run them through the BEST_ALG one last time to generate the remainder of the summary. That is, since after the first pass, the number of sentences in the generated summary is less than our required number, we take the remaining sentences (that were not chosen during the first pass) and once again run them through our preferred summarizer (chosen during the first pass) to complete the summary. Note that we have a guarantee of only requiring a maximum of two passes for our method.

4 Analysis and Results

4.1 Metric Selection

For comparing the efficacy of the algorithms, we opted to use the Jaccard index (also called the Jaccard similarity coefficient), calculated simply as $J(A,B) = \frac{|A \cap B|}{|A \cup B|}$ where A represents the words used in the Gold Standard summary and B represents the words used in the hypothesis summary. The Jaccard similarity coefficient has been shown to be effective and efficient for keyword similarity [9] and expanded to measure text similarity [10]. We have verified its appropriateness by comparing results to those reported for the BLEU and ROUGE metrics which, for our CNN set, tended to score random sentence selection favorably over most of the Sumy algorithms.

For example, while Luhn's ROUGE recall results were superior to all others, Luhn's precision scores were unremarkable, weighing down Luhn's F1 score towards the bottom, at roughly half of a random selection's F1 score. Furthermore, by using the Jaccard similarity coefficient, we were able to verify Luhn's advantage is consistent with results from functional analyses (in the form of querying) performed in [18].

4.2 Analysis

Results of our study are summarized in Figure 2, Figure 3, and Table 1, which reflect the mean values observed with ten randomized collections of training and test articles selected from the set of 3,000.

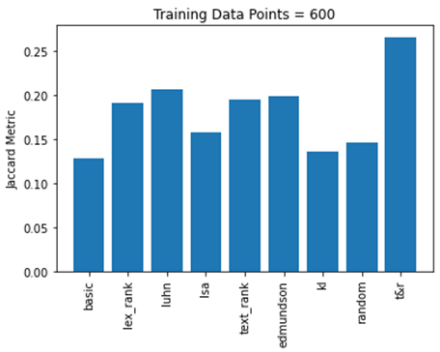

Figure 2. Training Results of using T&R

Using the Jaccard coefficient index, we show that the T&R approach produces improvements over every component algorithm, including a 4.9% improvement over Luhn and a 69% improvement over SumBasic.

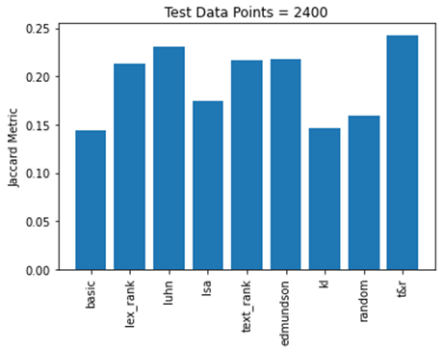

Figure 3. Test Results of using T&R

For comparison, BLEU and ROUGE-1 (unigram) F1 metrics were computed after performing T&R and obtained the results in Table 2.

Table 1. Jaccard Similarity Coefficients when using T&R

Algorithm / Meta-algorithm	Jaccard Similarity Coefficient	T&R Improvement
Basic	0.1436	0.6877
LexRank	0.2131	0.1376
Luhn	0.2310	0.0492
LSA	0.1741	0.3922
TextRank	0.2168	0.1179
Edmundson	0.2185	0.1093
KL	0.1463	0.6566
Random	0.1599	0.5160
T&R	0.2424	N/A

Table 2. Jaccard Results compared to BLEU and ROUGE-1

Algorithm /	Jaccard		BLEU		ROUGE-1	
Meta-algorithm	Index	Rank	Index	Rank	F1	Rank
Basic	0.1436	9	0.00238	1	0.1499	1
LexRank	0.2131	5	0.00157	5	0.1014	4
Luhn	0.2310	2	0.00126	8	0.0696	8
LSA	0.1741	6	0.00164	4	0.0930	5
TextRank	0.2168	4	0.00122	9	0.0675	9
Edmundson	0.2185	3	0.00182	2	0.0886	6
KL	0.1463	8	0.00143	6	0.1295	2
Random	0.1599	7	0.00174	3	0.1158	3
T&R	0.2424	1	0.00133	7	0.0731	7

A look into these summaries reveal that the BLEU and ROUGE-1 scores and rankings do not accurately reflect the relative summary representation of the complete text unlike those suggested by the Jaccard rankings. This is supported specifically by the relatively high score of the "Random" algorithm for both BLEU and ROUGE-1 (Rank = 3 for both). If a random summarization is evaluated as good or better than most

of the summarization techniques, there is either something amiss with the documents or with the evaluation approach. It appears that BLEU and ROUGE-1 both rank random summarizations too highly, undermining their overall credibility. This is quite different from the Jaccard metric.

The apparent accuracy of Jaccard over BLEU and ROUGE may, ironically, lie in its simplicity. Jaccard looks only at word sets (minus stop words) and should be considered for evaluating any summarization technique, perhaps as the main metric, as we have done, or as a supplement for evaluating the goodness of summaries. We surmise that since the sentences are extracted, there is a presumption that the sentences are well-formed and therefore focusing on word set intersections and unions only (as in the definition of the Jaccard index) is sufficient in providing accurate measures.

4.3 Limitations

Because of the way the SECOND_CHANCE_SET is obtained and tagged at the end of MATCH_SET, there is a probability that the summary will contain sentences that are not in the same order as the original text. This phenomenon is most likely to be a relevant consideration for summarizing timeline-critical corpora such as novels. News articles, such as in our study, are less affected by small perturbations in sentence order.

5 Conclusion

We have shown that given existing algorithms, regardless of their individual efficacy, we are able to obtain a 5% improvement in summarization results of news articles using the second-order meta-algorithmic pattern Tessellation and Recombination with Expert Decisioner.

We have also demonstrated careful consideration to metrics when evaluating summarization algorithms. Neither BLEU nor ROUGE-1 F1 results reflected the appropriateness of the summaries generated by our algorithms. Both BLEU and ROUGE ranked random selection higher than six of the eight algorithms. For the CNN corpus, the Jaccard similarity method is posed as a more germane means of assessment.

6 Future Work

As described above, the readability of summaries could potentially be improved by accurate ordering and placement of the SECOND_CHANCE_SET sentences as they relate to the rest of the generated summary. Since this would not improve the summary's Jaccard similarity coefficient as we have used it in this study, another metric would have to be considered. In addition, to further verify our findings in this research, we propose repetition of the functional tests described in [18] to study the effects of our proposed methods.

GPT-4 was not included in this study as it provides only abstractive summarization and not extractive without additional processing. However, the inclusion of modified ChatGPT results and other summarizers such as BERT may be the subject of an ensuing paper, with comparisons to T&R.

REFERENCES

[1] Berlin Chen, Hao-Chin Chang, and Kuan-Yu Chen. 2013. Sentence modeling for extractive speech summarization. In *2013 IEEE International Conference on Multimedia and Expo (ICME)*, IEEE, San Jose, CA, USA, 1–6. DOI:https://doi.org/10.1109/ICME.2013.6607518

[2] H. P. Edmundson. 1969. New Methods in Automatic Extracting. *J. ACM* 16, 2 (April 1969), 264–285. DOI:https://doi.org/10.1145/321510.321519

[3] Günes Erkan and Dragomir R. Radev. 2004. LexRank: graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.* 22, 1 (December 2004), 457–479.

[4] Rafael Dueire Lins, Hilario Oliveira, Luciano Cabral, Jamilson Batista, Bruno Tenorio, Rafael Ferreira, Rinaldo Lima, Gabriel de França Pereira e Silva, and Steven J. Simske. 2019. The CNN-Corpus: A Large Textual Corpus for Single-Document Extractive Summarization. In *Proceedings of the ACM Symposium on Document Engineering 2019* (DocEng '19), Association for Computing Machinery, New York, NY, USA, 1–10. DOI:https://doi.org/10.1145/3342558.3345388

[5] Shrabanti Mandal and Girish Kumar Singh. 2020. LSA Based Text Summarization. *(IJRTE)* 9, 2 (July 2020), 150–156. DOI:https://doi.org/10.35940/ijrte.B3288.079220

[6] Parth Mehta. 2016. From extractive to abstractive summarization: A journey. In *Proceedings of the ACL 2016 Student Research Workshop*, Association for Computational Linguistics, Berlin, Germany, 100–106. DOI:https://doi.org/10.18653/v1/P16-3015

[7] Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing Order into Texts. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, Barcelona, Spain.

[8] Ani Nenkova and Lucy Vanderwende. 2005. *The impact of frequency on summarization.* Microsoft Research, Redmond, Washington.

[9] Suphakit Niwattanakul, Jatsada Singthongchai, Ekkachai Naenudorn, and Supachanun Wanapu. 2013. Using of Jaccard Coefficient for Keywords Similarity. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, Hong Kong.

[10] Eva Y. Puspaningrum, Budi Nugroho, Ariyono Setiawan, and Nuraini Hariyanti. 2020. Detection of Text Similarity for Indication Plagiarism Using Winnowing Algorithm Based K-gram and Jaccard Coefficient. *J. Phys.: Conf. Ser.* 1569, 2 (July 2020), 022044. DOI:https://doi.org/10.1088/1742-6596/1569/2/022044

[11] Python Software Foundation. 2022. xml.etree.ElementTree — The ElementTree XML API. *Python documentation.* Retrieved May 1, 2022 from https://docs.python.org/3/library/xml.etree.elementtree.html

[12] Python Software Foundation. 2022. Beautiful Soup 4. *PyPI.* Retrieved April 1, 2022 from https://pypi.org/project/beautifulsoup4

[13] Python Software Foundation. 2022. xml.dom — The Document Object Model API. *Python documentation.* Retrieved April 1, 2022 from https://docs.python.org/3/library/xml.dom.html

[14] Leonard Richardson. Beautiful Soup. *Crummy.* Retrieved April 1, 2022 from https://www.crummy.com/software/BeautifulSoup/bs4/

[15] Steven J. Simske. 2013. *Meta-Algorithmics: Patterns for Robust, Low Cost, High Quality Systems.* John Wiley & Sons, New York.

[16] Steven Simske and Marie Vans. 2021. *Functional Applications of Text Analytics Systems.* River Publishers, Gistrup.

[17] Pradeepika Verma, Sukomal Pal, and Hari Om. 2019. A Comparative Analysis on Hindi and English Extractive Text Summarization. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 18, 3 (May 2019), 1–39. DOI:https://doi.org/10.1145/3308754

[18] Sam Wolyn and Steven J. Simske. 2022. Summarization assessment methodology for multiple corpora using queries and classification for functional evaluation. *Integr. Comput.-Aided Eng.* 29, 3 (January 2022), 227–239. DOI:https://doi.org/10.3233/ICA-220680