# EDU-Extractive Single-Document Summarization using Recursive Cardinality Potentials

Nitin Kishore Sai Samala
University of Massachusetts Amherst
nsamala@cs.umass.edu

## Abstract

*The initial investigation into automatic text summarization techniques began more than 50 years ago. Most of the earlier work, focused on general, news-related documents where the aim of the summarization tasks was to have relevant and up-to-date information for the users. Nowadays, there are more specific requirements and needs for document summarization. We tackle two difficult variants of this problem: single input document summarization and compressed sentence or EDU extractive summarization.We attempt a pure supervised learning approach to extract content based on ROUGE overlap score, using neural net features and a graphical models approach for inference. Since we can't solve for the MAP ROUGE configuration, we do learning with REINFORCE (the "score function estimator" of the gradient of the expected ROUGE reward).We compare our results against extractive and abstractive baselines on The New York Times Annotated Corpus(NYT50).*

## 1. Introduction

The Information age heralds exploding volumes of human and machine-generated data that grows exponentially each day. As a consequence of this information explosion, the amount of available textual information has risen at a commensurate rate, making the ability to store, process, analyze and interpret such unstructured data, a primary concern. Consumers of such information need help reducing the amount for a significant first encounter, even for normal tasks. This is prominently observed in news headlines, movie and product reviews, abridged books or song albums, and abstracts of scientific studies. Most readers, due to either lack of available time or other factors, would want to know the main content or gist of a document, without having to read the entire text. Summaries make the selection process easier and reduce reading time, when processing documents. Automatic summarization algorithms also improve the effectiveness of indexing and are less biased than human summarizers [19]. Radev et al. give the technical definition, for a text to be considered a summary as, "a text that is produced from one or more texts, that conveys important information in the original text(s), and that is no longer than half of the original text(s) and usually, significantly less than that" [15].Assimilating key pieces of information from a huge document, to produce an abridged and compact version without any human bias, is very challenging and still remains an active research problem.

This process of distilling the most important information from a source (or sources), to successfully generate and imitate summaries written by human beings, has 3 different dimensions or aspects to it, allowing us to categorize different techniques for document summarization. The categories are defined based on the output type- Extractive or Abstractive; the input type - single input document or multiple documents, and the purpose, whether it is specific to a certain domain, query-based or generic. In this paper we focus on the input and the output types.

Single document summarization which summarizes information present in one input document, is considered a much more difficult problem and has been less researched, than the multi-document counterpart, which generates a summary of multiple input documents, that often pertain to the same topic. Several graph and neural network based algorithms for multi-document summarization have been introduced recently( [12], [6], [7], [21]). Extracting information from a single document can be very tricky, when you don't have similar accompanying documents giving you an idea of which content is redundant. Single document summarization hasn't received as much attention as it did in the past( [20], [14], [17], [11]).

The two approaches based on output type are Extractive and Abstractive summarization. Like the name suggests, extractive methods assemble summaries by choosing particular sentences from the document, while abstractive methods involve generating new words or phrases not originally present in the document, to convey the content as a summary. The distinction between these is tantamount to the difference between highlighting important contents in a text document and writing your own abridged version of the text document. The extractive model at the very least ensures baseline levels of grammar and relevance. In this project we push the limits of the extractive models by compressing the extracted content further , from a sentence-based to a textual unit extraction.

## 1.1. Extractive Summarization

An aggressive approach to sub-sentential extraction can be viewed as deletion of unnecessary, Elementary Discourse Units(EDUs), that are contiguous parts of sentences. To increase the expressive capacity of our model we select the textual units that are scored according to a rich set of sparse features f, and model parameters or weights w, learned on the New York Times Annotated corpus (Sandhaus, 2008). The compression of individual sentences is licensed by a discursive formalism implemented as dependencies between EDUs. These units for compression are derived from the Rhetorical Structure Theory-based compressions of Hirao et al. (2013) where each sentence is composed of EDUs with RST relations like ELABORATION or SAME-UNIT between them indicating which units can be deleted and which have a dependency with the surrounding units, ensuring both are either present or absent in the summary. This ensures a tree structure where nodes can be deleted as long as we do not delete the parent of an included node.The model then is sufficiently constrained to have the expressive capacity to extract important content from units, but not at the expense of the summary's fluency. It views the input text as a set of EDUs and optimizes over binary variables indicating whether that particular unit is part of the summary.

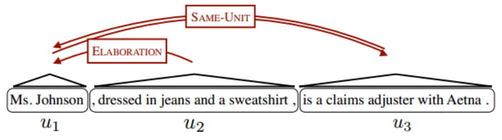
SAME-UNIT
ELABORATION
Ms. Johnson
, dressed in jeans and a sweatshirt ,
is a claims adjuster with Aetna .
$u_1$
$u_2$
$u_3$

Figure 1: Discourse parse showing different EDUs in a sentence

## 1.2. Recursive Cardinality Potentials

Cardinality potentials are useful class of high order potential that affect probabilities based on the number of active variables in D binary variables. The difficulty of computations associated the partition function for binary pairwise submodular models has led to relatively less progress, in efficient high order marginalization and sampling.The most common restriction to ensure tractability for exact or approximate inference is that the model should have low treewidth. Since cardinality potentials express constraints over the number of variables that take on a particular value, they are not limited by this. [18] introduces a high order class of potential that generalizes cardinality potentials, called Recursive Cardinality (RC) potentials, which are used for drawing exact joint samples and computing marginal probabilities in $O(Dlog^2D)$ time,to do efficient exact inference, in models composed of a tree structure and a cardinality potentials. They frame this as a belief propagation algorithm, to do MAP inference with efficient computations, in a low order tree-structured model that includes additional auxiliary variables, taking $O(DlogD)$ time.

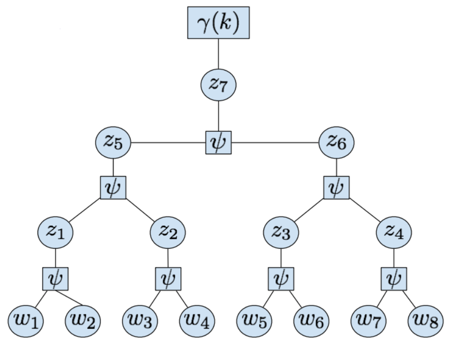

Figure 2: Cardinality potentials. $\psi$ enforces constraint that parent is the sum of children

RC potentials are defined in terms of a set of subsets S of a set of D binary variables, where for every pair of subsets in S has one either contained in or disjoint from another. In terms of a graphical modeling framework, a cardinality potential model over binary variables, where the probability is a Gibbs distribution, based on an energy function consisting of unary potentials $_d$ and one cardinality potential f() would look like

$$-E(y) = \sum_d \theta_d y_d + f(\sum_d y_d) \quad (1)$$

$$p(y) = \frac{e^{-E(y)}}{\sum_{y'} e^{-E(y')}} \quad (2)$$

## 1.3. FFT - Fast Fourier Transform

Chen and Liu (1997) speak of marginal distributions over cardinalities in models with only unary potentials, called Poisson-Binomial(PB) distributions.

Belfore (1995) provides a divide-and-conquer algorithm that recursively calls an FFT routine(equivalent to the upward pass of message passing in case of ordinary cardinality potentials),computing the probability that exactly k elements or between k to l elements, are chosen to be on, in $O(Dlog^2D)$ time. According to Fernandez and Williams(2010) and Hong (2011), computing the cumulative distribution function of a PB distribution that uses FFT, can be done by computing the characteristic function which takes $O(D^2)$ time and applying the inverse FFT in order to recover the CDF, effectively applying FFT only once. Given that efficient inference of marginal probabilities in such models takes $O(D^3logD)$ even with the best-known algorithm in machine learning from Potetz and Lee(2008), Tarlow et al. propose a fast FFT algorithm.

| Method | Runtime |
|---|---|
| PL(2008) Inference of marginal probabilities | $O(D^3logD)$ |
| Exact marginal computations in balanced RC | $O(Dlog^2D)$ |
| Computing all D messages or marginals | $O(D^3logD)$ |

Table 1: Inference run times

They include original and auxiliary variables in the tree-structured model which allows for exact maximum likelihood learning by formulating the problem as passing messages . These integer-valued auxiliary variables representing the count over subsets of original variables y, are constructed to keep distribution over y unchanged, while inference in the expanded model $q(y, z)$ where $\sum_z q(y, z) = p(y)$ can now be done very efficiently, as marginals are computed in q for both y and z variables. For each joint setting of y there will be exactly one joint setting of z with nonzero probability. The terms with high order cardinality like $f_k(\sum_{y_d \subset s_k} y_d)$ can be re-expressed as unary potentials $z_k$ on auxiliary variables and an internal node in the tree is created for every subset. This leads to a binary tree structure of depth $logD$, over y and z, with leaf nodes representing original y variables and sums of increasingly large subsets of y represented by z variables as intermediate nodes. Due to this, we can run sum-product belief propagation even when z variables have large cardinality. Tarlow et al. set a deterministic relationship between the auxiliary variable $z_p$ to be the sum of its children $z_l$ and $z_r$

$$q(y, z) \propto p(y) \prod_{p \subset P} 1_{z_p = z_{l_p} + z_{r_p}} \tag{3}$$

Given $P(z_l)$ and $P(z_r)$, $P(z_p)$ is simply the convolution of their PMFs.

Which makes the upward and downward messages computations as follows
Upward :

$$m_{g,z_p}(z_p) = \sum_{z_p = z_r + z_r}^{c_l} m_{z_l,g,}(z_l) m_{z_r,g}(z_p - z_l) \tag{4}$$

Downward:

$$m_{g,z_l}(z_l) = \sum_{z_r=0}^{c_l} m_{z_p,g,}(z_l + z_r) m_{z_r,g}(z_r) \tag{5}$$

### 1.4. Our Aim for this project

Supporting such single document summarization, that is very specific to a certain domain, is the main motivation for this paper. Some interesting applications would be to generate abstracts for scientific literature. Personalized summaries are also quite useful in question-answering systems [19]. Probabilistic graphical models have efficient algorithms for inference and learning. They are widely used due to their representational power. Durrett et al. 2016 use an ILP solver to do MAP inference in their model, but as it turns out, the "compression" and "syntactic constraint" parts of the model can be done with a certain type of dynamic program like recursive cardinality potentials. In this project, given a document and a set of parses to get constraints, we make a probabilistic model over possible extractions, including restricting the maximum potential size of the extraction. We train it based on MLE and Rouge-reinforce. Our goal is to use our model to generate abstracts for scientific literature without losing important tokens in the condensed version, similar to [2].

## 2. Related Work

Inspired by the application of deep learning methods for automatic machine translation, approaches framing the problem of text summarization as a sequence-to-sequence learning problem have shown promising results. Using RNN is standard approach for several NLP tasks

The closest related work is Durrett et al. 2016, where they present a discriminative model for single-document summarization that uses syntactic and discursive parsing formalisms to allow a more aggressive compression of individual sentences and improve cross-sentence coherence by coupling it with anaphoricity constraints. They frame the learning problem as a primal form of the structured SVM, train the model via stochastic subgradient descent and also optimize explicitly for ROUGE-1. They evaluate this system on content selection, clarity of language and referential structure.

Pointer Generator model in [16], is an abstractive summarizer that demonstrates the effectiveness of sequence-to-sequence model with attention, to produce a context vector as the weighted sum of the encoder hidden states, that calculates the vocabulary distribution. This chooses the word with the highest probability as the decoder output in each step, enabling the model to freely generate words in any order. To deal with inaccurate facts or repeated tokens, See et al 2017 introduce this hybrid network that penalizes repeated generation of words from a fixed vocabulary by using the attention distribution, to keep track of generated words via coverage, while copying tokens present in the article via pointing.

When it comes to using graph-based models, [11] introduced supervised and unsupervised approaches with graph-based syntactic representation of source text, for identifying the keywords to be used in extractive summarization, as it enhances the vector-space model by considering structural document features. [9] frames summarization as combinatorial optimization problem where they exploited rhetorical structures,in joint methods to utilize dependencies between words and between sentences to constructing a nested tree, which is trimmed without losing important content in the source document to create summaries.The nodes in the document tree represents the dependency between sentences were replaced by a sentence tree representing dependency between words

## 3. Methodology

Inference:

1. Receive document, compute RNN features
2. Compute sparse substructure scoring with RNN features P(extraction—document)
3. Output extraction or (optional) P(cleanup—extraction) from another RNN

Learning:

- (supervised) push extraction and/or cleanup towards ground truth summary
- (unsupervised) predict auxiliary variable (headline or auto-encode) with extraction, no cleanup model

### 3.1. Model

Refer to Figure 3

## 4. Dataset

The corpus used for training and evaluating the summarization models is the New York Times Annotated Corpus (Sandhaus, 2007), LDC2008T19. This dataset contained 110,540 articles with their corresponding abstractive summaries which we split these into 100,834 training and 9706 test examples, based on date of publication. To avoid including some summaries that were extremely short, we filtered this raw dataset by constraining the minimum length of the summaries to be no more than 50 words only contain documents with fewer than 100 EDUs. As a result, this dataset has a length distribution that ensures sufficient number of tokens to produce nontrivial summaries. This filtered test set called NYT50, has 3,452 test examples and 17,651

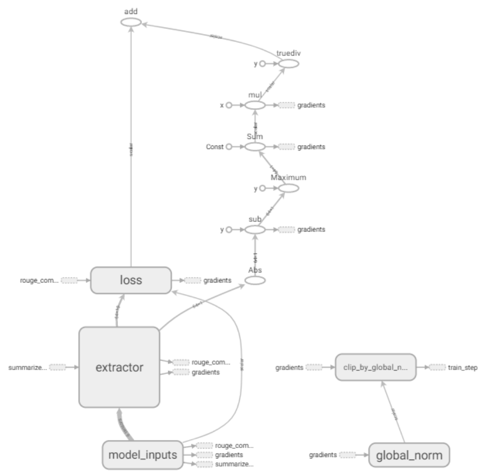

Figure 3: Model graph

train examples while a validation set with 4413 examples was also created. We used the GPU accelerated version of the Berkeley parser from [8], and the Berkeley Entity Resolution System from [4].We also used number and gender data from [1].The summarizer model in [3] gave us the data in the CoNLL format with constituency parses, NER, and coreference and some additional processing with EDU segmentation and discourse parsing. This split the data into abstracts, extractions and parses. Their discourse parser in the style of Hirao et al. (2013) follows the conditions from [22] as closely as possible and out performs the discourse dependency parser from Yoshida et al. (2014).The whole process took roughly 28 hours to shard the data into 10 sub-sampled data partitions and prepare it for our model. We maintain a vocabulary file with entries like ”token,count,token stem and whether token is a stop word”, using a porter stemmer and a stop word list. We also have mapping dictionaries from token to their stems and indices.

The Pointer Generator model for abstractive summarization was done on the CNN and DailyMail corpus [16]. We process the NYT corpus and discarded those files with missing text.The text articles and abstracts were then tokenized using the PTB Tokenizer from the Stanford CoreNLP package [13] to generate a vocabulary file and documents were split into chunks of 1000 examples per chunk and stored as Binary format files (*.bin) which are a lot smaller than their text equivalents. These were then converted to sequence of serialized tensorflow.core.example.example_pb2.Example objects to work with their code since file formats TensorFlow are based on Protocol Buffers.

## 5. Training

We are training on a large corpus of documents that are paired with their corresponding reference summaries and use an efficient deep bidirectional rnn to get the features. We optimize using Adam with learning rate 1e-5 and epsilon 1e-4. Training starts with creating batches of SummaryExample objects that contain example ids, article sentences, edu ids, parent edu ids, extraction labels, abstract sentences. Cardinality-constrained inference can be numerically unstable so we come up with the following hacks

1. Message damping and auxiliary constraints

$$\mu_{z_l,z_r \to z_p}(i) \leftarrow (1-\lambda_{damp}*)\mu_{z_l,z_r \to z_p}(i) + \lambda_{damp} \tag{6}$$

2. Regularize logZ

$$Loss \leftarrow Loss + \lambda_{logz} * max(0, |logz| - 100.0) \tag{7}$$

- Turn sequences of words with extraction labels into a bag of words
- Compute micro/macro ROUGE scores for samples and predictions
- Get sufficient statistics for ROUGE-1 recall computation

Main summarizer model class with learning and inference graphs has different choices for the extractor and the loss.
- Extractor that uses cardinality potentials to limit number of extractions with the constraint that number of selected things must be exactly the length number of the gold abstract, with log-score close to -inf on other counts.
- Extractor that uses discourse-parse constrained inference to limit number of extractions.
- Cross entropy loss with supervised extraction labels.
- REINFORCE loss to optimize ROUGE-1 recall score with summary.

The number of epochs and iterations wasn't set to a hard limit. The training steps ran on an infinite loop and we used Early stopping as a way to avoid over-fitting and stop training. Batch size is set to 64.

## 6. Evaluation

Essentially a summary is generated text and there isn't any supposedly ideal gold standard summary, that is free from bias and encompasses all the important factual content. Human evaluation is not only expensive but also subjective and therefore prone to high variance. Since our process is extractive, we need to also keep in mind the grammaticality, readability and coherence of the resulting summary. ROUGE [10]or "Recall Oriented Understudy for Gisting Evaluation" is the most widely used metric for automatic evaluation of a summary by comparing it to gold standard human summaries. It is recall-based measure that compares how many n-grams match between a candidate and the reference summary [5]. The result of calculating ROUGE for the NYT50 dataset summaries is presented in Table 2.

## 7. Results

We wanted to compare our results against both the state-of-the-art extractive and abstractive single document summarization baselines. For baselines, we obtain the rouge-1 and rouge-2 scores for ILP based extraction model from the Compression and Anaphoricity Constraints based Single-Document summarization [4].We run the Pointer-generator with and without coverage on the processed NYT50 dataset to establish a baseline with an Abstractive model.

| Model results on NYT50 | Rouge-1 | Rouge-2 |
|---|---|---|
| ILP Extractive Summarizer (Berkeley) | 42.2 | 25.9 |
| Pointer generator (Abstractive) | 41.79 | 24.29 |
| PG with coverage | 42.06 | 24.25 |
| Independent Cardinality potentials | Rouge-1 | Rouge-2 |
| MLE | 38.7 | - |
| First k-cardinality | 40.6 | - |
| Rouge Reinforce | 48.1 | - |
| Tree constrained (FFT) | Rouge-1 | Rouge-2 |
| MLE |  | - |
| First k-cardinality |  | - |
| Rouge Reinforce |  | - |

Table 2: Results. The first result is from Durrett et.al 2016

## 8. Conclusions and Future work

In the imminent information overload age, there is an immense and vital need for summarization tools. We present a summarization system that tackles two difficult and relatively less researched areas of document summarization. Our single document summarizer shows promising results on rouge evaluation of the summaries.

For future work, we hope to combine our model with an abstractive pipeline to generate higher quality summaries and make it capable of generalization, paraphrasing using sub-textual cues and the incorporating real-world knowledge which is thought possible only in an abstractive framework. Another avenue to pursue is some sort of unsupervised or semi-supervised learning to extract EDUs based on another objective, such as reconstruction of the original document (which might have problems because it'll over-emphasize rare words) or prediction of meta-data like headlines, etc. We can also try pyramid evaluation method to see how much factual content is present in the summary.

## References

[1] S. Bergsma and D. Lin. Bootstrapping path-based pronoun resolution. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 33–40. Association for Computational Linguistics, 2006.

[2] E. Collins, I. Augenstein, and S. Riedel. A supervised approach to extractive summarisation of scientific papers. *arXiv preprint arXiv:1706.03946*, 2017.

[3] G. Durrett, T. Berg-Kirkpatrick, and D. Klein. Learning-based single-document summarization with compression and anaphoricity constraints. *arXiv preprint arXiv:1603.08887*, 2016.

[4] G. Durrett and D. Klein. A joint model for entity analysis: Coreference, typing, and linking. *Transactions of the Association for Computational Linguistics*, 2:477–490, 2014.

[5] R. Ferreira, L. de Souza Cabral, R. D. Lins, G. P. e Silva, F. Freitas, G. D. Cavalcanti, R. Lima, S. J. Simske, and L. Favaro. Assessing sentence scoring techniques for extractive text summarization. *Expert systems with applications*, 40(14):5755–5764, 2013.

[6] J. Goldstein, V. Mittal, J. Carbonell, and M. Kantrowitz. Multi-document summarization by sentence extraction. In *Proceedings of the 2000 NAACL-ANLP Workshop on Automatic summarization*, pages 40–48. Association for Computational Linguistics, 2000.

[7] A. Haghighi and L. Vanderwende. Exploring content models for multi-document summarization. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 362–370. Association for Computational Linguistics, 2009.

[8] D. Hall, T. Berg-Kirkpatrick, and D. Klein. Sparser, better, faster gpu parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 208–217, 2014.

[9] Y. Kikuchi, T. Hirao, H. Takamura, M. Okumura, and M. Nagata. Single document summarization based on nested tree structure. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 315–320, 2014.

[10] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*, 2004.

[11] M. Litvak and M. Last. Graph-based keyword extraction for single-document summarization. In *Proceedings of the workshop on Multi-source Multilingual Information Extraction and Summarization*, pages 17–24. Association for Computational Linguistics, 2008.

[12] K. Mani, I. Verma, and L. Dey. Multi-document summarization using distributed bag-of-words model. *arXiv preprint arXiv:1710.02745*, 2017.

[13] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014.

[14] D. R. Radev, S. Blair-Goldensohn, and Z. Zhang. Experiments in single and multi-document summarization using mead. *Ann Arbor*, 1001:48109, 2001.

[15] D. R. Radev, E. Hovy, and K. McKeown. Introduction to the special issue on summarization. *Computational linguistics*, 28(4):399–408, 2002.

[16] A. See, P. J. Liu, and C. D. Manning. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*, 2017.

[17] K. Svore, L. Vanderwende, and C. Burges. Enhancing single-document summarization by combining ranknet and third-party sources. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, 2007.

[18] D. Tarlow, K. Swersky, R. S. Zemel, R. P. Adams, and B. J. Frey. Fast exact inference for recursive cardinality models. *arXiv preprint arXiv:1210.4899*, 2012.

[19] J.-M. Torres-Moreno. Automatic Text Summarization, Pages 4-5, 2014.

[20] X. Wan and J. Xiao. Exploiting neighborhood knowledge for single document summarization and keyphrase extraction. *ACM Transactions on Information Systems (TOIS)*, 28(2):8, 2010.

[21] M. Yasunaga, R. Zhang, K. Meelu, A. Pareek, K. Srinivasan, and D. Radev. Graph-based neural multi-document summarization. *arXiv preprint arXiv:1706.06681*, 2017.

[22] Y. Yoshida, J. Suzuki, T. Hirao, and M. Nagata. Dependency-based discourse parser for single-document summarization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1834–1839, 2014.