

Text Summarization through Entailment-based Minimum Vertex Cover

Anand Gupta¹, Manpreet Kaur², Adarsh Singh², Aseem Goel², Shachar Mirkin³

¹Dept. of Information Technology, NSIT, New Delhi, India

²Dept. of Computer Engineering, NSIT, New Delhi, India

³Xerox Research Centre Europe, Meylan, France

Abstract

Sentence Connectivity is a textual characteristic that may be incorporated intelligently for the selection of sentences of a well meaning summary. However, the existing summarization methods do not utilize its potential fully. The present paper introduces a novel method for single-document text summarization. It poses the text summarization task as an optimization problem, and attempts to solve it using *Weighted Minimum Vertex Cover* (WMVC), a graph-based algorithm. Textual entailment, an established indicator of semantic relationships between text units, is used to measure sentence connectivity and construct the graph on which WMVC operates. Experiments on a standard summarization dataset show that the suggested algorithm outperforms related methods.

1 Introduction

In the present age of digital revolution with proliferating numbers of internet-connected devices, we are facing an exponential rise in the volume of available information. Users are constantly facing the problem of deciding what to read and what to skip. Text summarization provides a practical solution to this problem, causing a resurgence in research in this field.

Given a topic of interest, a standard search often yields a large number of documents. Many of them are not of the user's interest. Rather than going through the entire result-set, the reader may read a gist of a document, produced via summarization tools, and then decide whether to fully read the document or not, thus saving a substantial amount of time. According to Jones (2007), a summary can be defined as "a reductive transformation of source text to summary text through

content reduction by selection and/or generalization on what is important in the source". Summarization based on content reduction by selection is referred to as *extraction* (identifying and including the important sentences in the final summary), whereas a summary involving content reduction by generalization is called *abstraction* (reproducing the most informative content in a new way).

The present paper focuses on extraction-based single-document summarization. We formulate the task as a graph-based optimization problem, where vertices represent the sentences and edges the connections between sentences. Textual entailment (Giampiccolo et al., 2007) is employed to estimate the degree of connectivity between sentences, and subsequently to assign a weight to each vertex of the graph. Then, the Weighted Minimum Vertex Cover, a classical graph algorithm, is used to find the minimal set of vertices (that is – sentences) that forms a cover. The idea is that such cover of well-connected vertices would correspond to a cover of the salient content of the document.

The rest of the paper is organized as follows: In Section 2, we discuss related work and describe the WMVC algorithm. In Section 3, we propose a novel summarization method, and in Section 4, experiments and results are presented. Finally, in Section 5, we conclude and outline future research directions.

2 Background

Extractive text summarization is the task of identifying those text segments which provide important information about the gist of the document – the salient units of the text. In (Marcu, 2008), salient units are determined as the ones that contain frequently-used words, contain words that are within titles and headings, are located at the beginning or at the end of sections, contain key phrases and are *the most highly connected to other parts*

of the text. In this work we focus on the last of the above criteria, connectivity, to find highly connected sentences in a document. Such sentences often contain information that is found in other sentences, and are therefore natural candidates to be included in the summary.

2.1 Related Work

The connectivity between sentences has been previously exploited for extraction-based summarization. Salton et al. (1997) generate intra-document links between passages of a document using automatic hypertext link generation algorithms. Mani and Bloedorn (1997) use the number of shared words, phrases and co-references to measure connectedness among sentences. In (Barzilay and Elhadad, 1999), lexical chains are constructed based on words relatedness.

Textual entailment (TE) was exploited recently for text summarization in order to find the highly connected sentences in the document. Textual entailment is an asymmetric relation between two text fragments specifying whether one fragment can be inferred from the other. Tatar et al. (2008) have proposed a method called Logic Text Tiling (LTT), which uses TE for sentence scoring that is equal to the number of entailed sentences and to form text segments comprising of highly connected sentences. Another method called Analog Textual Entailment and Spectral Clustering (ATESC), suggested in (Gupta et al., 2012), also uses TE for sentence scoring, using analog scores.

We use a graph-based algorithm to produce the summary. Graph-based ranking algorithms have been employed for text summarization in the past, with similar representation to ours. Vertices represent text units (words, phrases or sentences) and an edge between two vertices represent any kind of relationship between two text units. Scores are assigned to the vertices using some relevant criteria to select the vertices with the highest scores. In (Mihalcea and Tarau, 2004), content overlap between sentences is used to add edges between two vertices and Page Rank (Page et al., 1999) is used for scoring the vertices. Erkan and Radev (2004) use inter-sentence cosine similarity based on word overlap and *tf-idf* weighting to identify relations between sentences. In our paper, we use TE to compute connectivity between nodes of the graph and apply the weighted minimum vertex cover (WMVC) algorithm on the graph to select

the sentences for the summary.

2.2 Weighted MVC

WMVC is a combinatorial optimization problem listed within the classical NP-complete problems (Garey and Johnson, 1979; Cormen et al., 2001). Over the years, it has caught the attention of many researchers, due to its NP-completeness, and also because its formulation complies with many real world problems.

Weighted Minimum Vertex Cover Given a weighted graph $G = (V, E, w)$, such that w is a positive weight (cost) function on the vertices, $w : V \rightarrow \mathbb{R}$, a weighted minimum vertex cover of G is a subset of the vertices, $C \subseteq V$ such that for every edge $(u, v) \in E$ either $u \in C$ or $v \in C$ (or both), and the total sum of the weights is minimized.

$$C = \operatorname{argmin}_{C'} \sum_{v \in C'} w(v) \quad (1)$$

3 Weighted MVC for text summarization

We formulate the text summarization task as a WMVC problem. The input document to be summarized is represented as a weighted graph $G = (V, E, w)$, where each of $v \in V$ corresponds to a sentence in the document; an edge $(u, v) \in E$ exists if either u entails v or v entails u with a value at least as high as an empirically-set threshold. A weight w is then assigned to each sentence based on (negated) TE values (see Section 3.2 for further details). WMVC returns a cover C which is a subset of the sentences with a minimum total weight, corresponding to the best connected sentences in the document. The cover is our output – the summary of the input document.

Our proposed method, shown in Figure 1, consists of the following main steps.

1. Intra-sentence textual entailment score computation
2. Entailment-based connectivity scoring
3. Entailment connectivity graph construction
4. Application of WMVC to the graph

We elaborate on each of these steps in the following sections.

3.1 Computing entailment scores

Given a document d for which summary is to be generated, we represent d as an array of sentences

Id	Sentence
S_1	A representative of the African National Congress said Saturday the South African government may release black nationalist leader Nelson Mandela as early as Tuesday.
S_2	“There are very strong rumors in South Africa today that on Nov. 15 Nelson Mandela will be released,” said Yusef Saloojee, chief representative in Canada for the ANC, which is fighting to end white-minority rule in South Africa.
S_3	Mandela the 70-year-old leader of the ANC jailed 27 years ago, was sentenced to life in prison for conspiring to overthrow the South African government.
S_4	He was transferred from prison to a hospital in August for treatment of tuberculosis.
S_5	Since then, it has been widely rumoured Mandela will be released by Christmas in a move to win strong international support for the South African government.
S_6	“It will be a victory for the people of South Africa and indeed a victory for the whole of Africa,” Saloojee told an audience at the University of Toronto.
S_7	A South African government source last week indicated recent rumours of Mandela’s impending release were orchestrated by members of the anti-apartheid movement to pressure the government into taking some action.
S_8	And a prominent anti-apartheid activist in South Africa said there has been “no indication (Mandela) would be released today or in the near future.”
S_9	Apartheid is South Africa’s policy of racial separation.
Summary	“There are very strong rumors in South Africa today that on Nov.15 Nelson Mandela will be released,” said Yusef Saloojee, chief representative in Canada for the ANC, which is fighting to end white-minority rule in South Africa. He was transferred from prison to a hospital in August for treatment of tuberculosis. A South African government source last week indicated recent rumours of Mandela’s impending release were orchestrated by members of the anti-apartheid movement to pressure the government into taking some action. Apartheid is South Africa’s policy of racial separation.

Table 1: The sentence array of article *AP881113-0007* of cluster *do106* in the DUC’02 dataset.

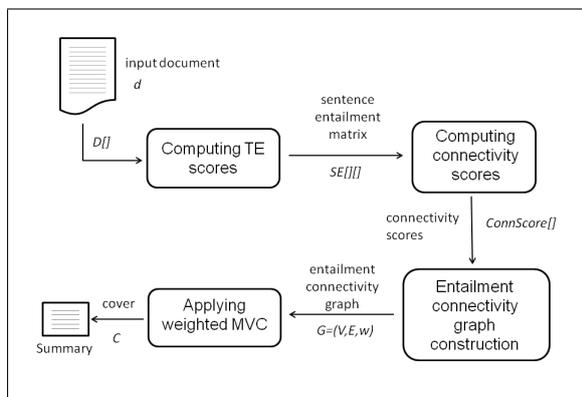


Figure 1: Outline of the proposed method.

$D_{1 \times N}$. An example article is shown in Table 1. We use this article to demonstrate the steps of our algorithm.

Then, we compute a TE score between every possible pair of sentences in D using a textual entailment tool. TE scores for all the pairs are stored in a *sentence entailment matrix*, $SE_{N \times N}$. An entry $SE[i, j]$ in the matrix represents the extent by which sentence i entails sentence j . The sentence entailment matrix produced for our example document is shown in Table 2.

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
S_1	-	0	0	0.04	0	0	0.001	0.02	0.02
S_2	0.02	-	0.01	0.04	0.06	0.01	0	0.01	0.04
S_3	0	0	-	0.09	0	0	0	0	0.04
S_4	0	0	0	-	0	0	0	0	0.01
S_5	0	0	0	0.04	-	0	0.01	0.01	0.04
S_6	0	0	0	0.04	0	-	0	0	0.02
S_7	0	0	0	0.04	0.06	0	-	0.02	0.27
S_8	0	0	0	0.04	0	0	0.01	-	0.02
S_9	0	0	0	0.04	0	0	0	0	-

Table 2: The sentence entailment matrix of the example article.

Id	<i>ConnScore</i>	Id	<i>ConnScore</i>
S_1	0.08	S_6	0.06
S_2	0.19	S_7	0.39
S_3	0.13	S_8	0.07
S_4	0.01	S_9	0.04
S_5	0.1		

Table 3: Connectivity Scores of the sentences of article *AP881113-0007*.

3.2 Connectivity scores

Our assumption is that entailment between sentences indicates connectivity, that – as mentioned above – is an indicator of sentence salience. More specifically, salience of a sentence is determined by the degree by which it entails other sentences in the document. We thus use the sentence entailment matrix to compute a connectivity score for each sentence by summing the entailment scores of the sentence with respect to the rest of the sentences in the document, and denote this sum as *ConnScore*. Formally, *ConnScore* for sentence i is computed as follows.

$$ConnScore[i] = \sum_{i \neq j} SE[i, j] \quad (2)$$

Applying it to each sentence in the document, we obtain the $ConnScore_{1 \times N}$ vector. The sentence connectivity scores corresponding to Table 2 are shown in Table 3.

3.3 Entailment connectivity graph construction

The more a sentence is connected, the higher its connectivity score. To adapt the scores to the WMVC algorithm, that searches for a *minimal* solution, we convert the scores into positive weights

in inverted order:

$$w[i] = -ConnScore[i] + Z \quad (3)$$

$w[i]$ is the score that is assigned to the vertex of sentence i ; Z is a large constant, meant to keep the scores positive. In this paper, Z has been assigned value = 100. Now, the better a sentence is connected, the lower its weight.

Given the weights, we construct an undirected weighted entailment connectivity graph, $G(V, E, w)$, for the document d . V consists of vertices for the document’s sentences, and E are edges that correspond to the entailment relations between the sentences. w is the weight explained above. We create an edge between two vertices as explained below. Suppose that S_i and S_j are two sentences in d , with entailment scores $SE[i, j]$ and $SE[j, i]$ between them. We set a threshold τ for the entailment scores as the mean of all entailment values in the matrix SE . We add an edge (i, j) to G if $SE[i, j] \geq \tau$ OR $SE[j, i] \geq \tau$, i.e. if at least one of them is as high as the threshold.

Figure 2 shows the connectivity graph constructed for the example in Table 1.

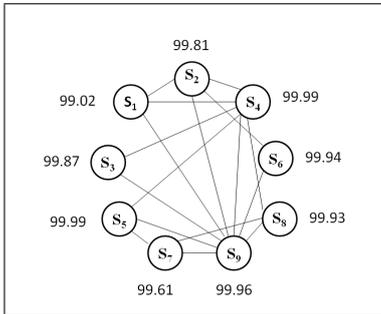


Figure 2: The Entailment connectivity graph of the considered example with associated *Score* of each node shown.

3.4 Applying wMVC

Finally, we apply the weighted minimum vertex cover algorithm to find the minimal vertex cover, which would be the document’s summary. We use integer linear programming (ILP) for finding a minimum cover. This algorithm is a 2-approximation for the problem, meaning it is an efficient (polynomial-time) algorithm, guaranteed to find a solution that is no more than 2 times bigger than the optimal solution.¹ The algorithm’s

¹We have used an implementation of ILP for wMVC in MATLAB, *grMinVerCover*.

input is $G = (V, E, w)$, a weighted graph where each vertex $v_i \in V (1 \leq i \leq n)$ has weight w_i . Its output is a minimal vertex cover C of G , containing a subset of the vertices V . We then list these sentences as our summary, according to their original order in the document.

After applying wMVC to the graph in Figure 2, the cover C returned by the algorithm is $\{S_2, S_4, S_7, S_9\}$ (highlighted in Figure 2).

Whenever a summary is required, a word-limit on the summary is specified. We find the threshold which results with a cover that matches the word limit through binary search.

4 Experiments and results

4.1 Experimental settings

We have conducted experiments on the single-document summarization task of the *DUC 2002* dataset², using a random sample that contains 60 news articles picked from each of the 60 clusters available in the dataset. The target summary length limit has been set to 100 words. We used version 2.1.1 of BIUTEE (Stern and Dagan, 2012), a transformation-based TE system to compute textual entailment score between pairs of sentences.³ BIUTEE was trained with 600 text-hypothesis pairs of the RTE-5 dataset (Bentivogli et al., 2009).

4.1.1 Baselines

We have compared our method’s performance with the following re-implemented methods:

1. **Sentence selection with tf-idf**: In this baseline, sentences are ranked based on the sum of the *tf-idf* scores of all the words except stopwords they contain, where *idf* figures are computed from the dataset of 60 documents. Top ranking sentences are added to the summary one by one, until the word limit is reached.
2. **LTT**: (see Section 2)
3. **ATESC**: (see Section 2)

4.1.2 Evaluation metrics

We have evaluated the method’s performance using *ROUGE* (Lin, 2004). *ROUGE* measures the

²http://www-nlpir.nist.gov/projects/duc/data/2002_data.html

³Available at: <http://www.cs.biu.ac.il/~nlp/downloads/biutee>.

Method	P (%)	R (%)	F_1 (%)
TF-IDF	13.3	17.6	15.1
LTT	39.9	34.6	37.1
ATESC	37.7	32.5	34.9
wMVC	39.8	38.8	39.2

Table 4: *ROUGE-1* results.

Method	P (%)	R (%)	F_1 (%)
TF-IDF	7.4	9.6	8.4
LTT	18.4	15.2	16.6
ATESC	16.3	11.7	13.6
wMVC	16.7	16.8	16.8

Table 5: *ROUGE-2* results.

quality of an automatically-generated summary by comparing it to a “gold-standard”, typically a human generated summary. *ROUGE-n* measures n -gram precision and recall of a candidate summary with respect to a set of *reference summaries*. We compare the system-generated summary with two reference summaries for each article in the dataset, and show the results for *ROUGE-1*, *ROUGE-2* and *ROUGE-SU4* that allows skips within n -grams. These metrics were shown to perform well for single document text summarization, especially for short summaries. Specifically, Lin and Hovy (2003) showed that *ROUGE-1* achieves high correlation with human judgments.⁴

4.2 Results

The results for *ROUGE-1*, *ROUGE-2* and *ROUGE-SU4* are shown in Tables 4, 5 and 6, respectively. For each, we show the precision (P), recall (R) and F_1 scores. Boldface marks the highest score in each table. As shown in the tables, our method achieves the best score for each of the three metrics.

4.3 Analysis

The entailment connectivity graph generated conveys information about the connectivity of sentences in the document, an important parameter for indicating the salience of a sentences.

The purpose of the wMVC is therefore to find a subset of the sentences that are well-connected and cover all the content of all the sentences. Note that merely selecting the sentences on the basis of a *greedy approach*, that picks the those sentences with the highest connectivity score, does not ensure that all edges of the graph are cov-

⁴See (Lin, 2004) for formal definitions of these metrics.

Method	P (%)	R (%)	F_1 (%)
TF-IDF	2.2	4.2	2.9
LTT	16	11.8	13.6
ATESC	15.5	11.1	12.9
wMVC	14.1	14.2	14.2

Table 6: *ROUGE-SU4* results.

ered, i.e. it does not ensure that all the information is covered in the summary. In Figure 3, we illustrate the difference between wMVC (left) and a greedy algorithm (right) over our example document. The vertices selected by each algorithm are highlighted. The selected set by wMVC, $\{S_2, S_4, S_7, S_9\}$, covers all the edges in the graph. In contrast, using the greedy algorithm, the subset of vertices selected on the basis of highest scores is $\{S_2, S_3, S_7, S_8\}$. There, several edges are not covered (e.g. $(S_1 \rightarrow S_9)$).

It is therefore much more in sync with the summarization goal of finding a subset of sentences that conveys the important information of the document in a compressed manner.

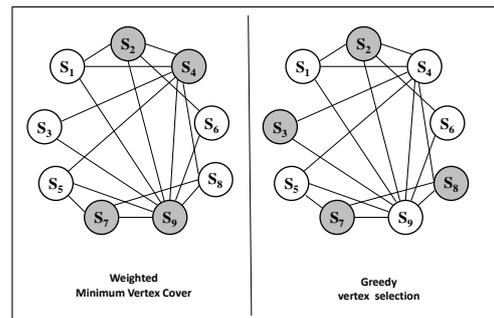


Figure 3: Minimum Vertex Cover vs. Greedy selection of sentences.

5 Conclusions and future work

The paper presents a novel method for single-document extractive summarization. We formulate the summarization task as an optimization problem and employ the weighted minimum vertex cover algorithm on a graph based on textual entailment relations between sentences. Our method has outperformed previous methods that employed TE for summarization as well as a frequency-based baseline. For future work, we wish to apply our algorithm on smaller segments of the sentences, using *partial textual entailment* Levy et al. (2013), where we may obtain more reliable entailment measurements, and to apply the same approach for multi-document summarization.

References

- Regina Barzilay and Michael Elhadad. 1999. Using lexical chains for text summarization. In *In Inderjeet Mani and Mark T. Maybury, editors, Advances in Automatic Text Summarization*, pages 111–121, The MIT Press, 1999.
- Luisa Bentivogli, Ido Dagan, Hoa Trang Dang, Danilo Giampiccolo, and Bernardo Magnini. 2009. The fifth pascal recognizing textual entailment challenge. In *Proceedings of Text Analysis Conference*, pages 14–24, Gaithersburg, Maryland USA.
- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. 2001. *Introduction to Algorithms*. McGraw-Hill, New York, 2nd edition.
- Gunes Erkan and Dragomir R. Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research (JAIR)*, 22(1):457–479.
- Michael R. Garey and David S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. FREEMAN, New York.
- Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third pascal recognizing textual entailment challenge. In *Proceedings of the Association for Computational Linguistics, ACL'07*, pages 1–9, Prague, Czech Republic.
- Anand Gupta, Manpreet Kaur, Arjun Singh, Ashish Sachdeva, and Shruti Bhati. 2012. Analog textual entailment and spectral clustering (atesc) based summarization. In *Lecture Notes in Computer Science, Springer*, pages 101–110, New Delhi, India.
- Karen Spark Jones. 2007. Automatic summarizing: The state of the art. *Information Processing and Management*, 43:1449–1481.
- Omer Levy, Torsten Zesch, Ido Dagan, and Iryna Gurevych. 2013. Recognizing partial textual entailment. In *Proceedings of the Association for Computational Linguistics*, pages 17–23, Sofia, Bulgaria.
- Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 71–78, Edmonta, Canada, 27 May- June 1.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out*, pages 25–26, Barcelona, Spain.
- Inderjeet Mani and Eric Bloedorn. 1997. Multi-document summarization by graph search and matching. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97)*, American Association for Artificial Intelligence, pages 622–628, Providence, Rhode Island.
- Daniel Marcu. 2008. From discourse structure to text summaries. In *Proceedings of the ACL/EACL '97, Workshop on Intelligent Scalable Text Summarization*, pages 82–88, Madrid, Spain.
- Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into texts. In *Proceedings of EMNLP*, volume 4(4), page 275, Barcelona, Spain.
- Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web. *Technical Report*.
- Gerard Salton, Amit Singhal, Mandar Mitra, and Chris Buckley. 1997. Automatic text structuring and summarization. *Information Processing and Management*, 33:193–207.
- Asher Stern and Ido Dagan. 2012. BIUTEE: A modular open-source system for recognizing textual entailment. In *Proceedings of the ACL 2012 System Demonstrations*, pages 73–78, Jeju, Korea.
- Doina Tatar, Emma Tamaianu Morita, Andreea Mihis, and Dana Lupsa. 2008. Summarization by logic segmentation and text entailment. In *Conference on Intelligent Text Processing and Computational Linguistics (CICLing 08)*, pages 15–26, Haifa, Israel.