Polytope Model for Extractive Summarization

Marina Litvak and Natalia Vanetik
Department of Software Engineering, Sami Shamoon College of Engineering, Beer Sheva, Israel

Keywords: Text Summarization, Quadratic Programming, Polytope Model.

Abstract: The problem of text summarization for a collection of documents is defined as the problem of selecting a small subset of sentences so that the contents and meaning of the original document set are preserved in the best possible way. In this paper we present a linear model for the problem of text summarization, where we strive to obtain a summary that preserves the information coverage as much as possible in comparison to the original document set. We construct a system of linear inequalities that describes the given document set and its possible summaries and translate the problem of finding the best summary to the problem of finding the point on a convex polytope closest to the given hyperplane. This re-formulated problem can be solved efficiently with the help of quadratic programming.

1 INTRODUCTION

Automated text summarization is an active field of research in various communities like Information Retrieval (IR), Natural Language Processing (NLP), and Text Mining (TM). Summarization is important for IR since it helps to access large repositories of textual data efficiently by identifying the essence of a document and indexing a repository. Taxonomically, we distinguish between *single-document*, where a summary per single document is generated, and *multi-document*, where a summary per cluster of related documents is generated, summarization. Also, we distinguish between automatically generated *extract*—the most salient fragments of the input document/s (e.g., sentences, paragraphs, etc.) and *abstract*—re-formulated synopsis expressing the main idea of the input document/s. Since generating abstracts requires a deep linguistic analysis of the input documents, most existing summarizers work in extractive manner (Mani and Maybury, 1999). Moreover, extractive summarization can be applied to cross-lingual/multilingual domains (Litvak et al., 2010).

In this paper we deal with the problem of extractive summarization. Our method can be generalized for both single-document and multi-document summarization. Since the method includes only very basic linguistic analysis (see section 5.4), it can be applied to cross-lingual/multilingual summarization.

Formally speaking, in this paper we introduce:

- A novel text representation model expanding a classic Vector Space Model (Salton et al., 1975) to Hyperplane and Half-spaces;
- A distance measure between text and information coverage we wish to preserve;
- A re-formulated extractive summarization problem as a distance minimizing task and its solution using quadratic programming.

The main challenge of this paper is a new text representation model making possible to represent an exponential number of extracts without computing them explicitly, and finding the optimal one by simple minimizing a distance function in polynomial time.

This paper is organized as follows: section 2 depicts related work, section 3 describes problem setting and definitions, section 4 introduces a new text representation model and a possible distance measure between text and information coverage, section 5 refers summarization task as a distance optimization in a new text representation model. We discuss as *unsupervised* as *supervised* approaches. Last section contains our future work and conclusions.

2 RELATED WORK

Numerous techniques for automated summarization have been introduced in the last decades, trying to reduce the constant information overload of professionals in a variety of fields. Many works formulated

Litvak M. and Vanetik N..
Polytope Model for Extractive Summarization.
DOI: 10.5220/0004170902810286
In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval* (KDIR-2012), pages 281-286
ISBN: 978-989-8565-29-7

the summarization as optimization problem, solving it using such techniques like a standard hill-climbing algorithm(Hassel and Sjobergh, 2006), regression models (Ouyang et al., 2011), and Evolutionary algorithms (Alfonseca and Rodriguez, 2003; Liu et al., 2006).

Some authors reduce summarization to the maximum coverage problem (Takamura and Okumura, 2009). The maximum coverage model extracts sentences to a summary to cover as many information as possible, where information can be measured by text units like terms, n-grams, etc. Despite a great performance (Takamura and Okumura, 2009; Gillick and Favre, 2009) in summarization field, maximum coverage problem is known as NP-hard (Khuller et al., 1999). Some works attempt to find a near-optimum solution by greedy approach (Filatova, 2004; Takamura and Okumura, 2009). Linear Programming helps to find a more accurate approximated solution to the maximum coverage problem and became very popular in summarization field in the last years (Gillick and Favre, 2009; Woodsend and Lapata, 2010; Hitoshi Nishikawa and Kikui, 2010; Makino et al., 2011).

Trying to solve a trade-off between summary quality and time complexity, we propose a novel summarization model solving the approximated maximum coverage problem by quadratic programming in polynomial time. We measure information coverage by terms (normalized meaningful words) and strive to obtain a summary that preserves the term frequency as much as possible in comparison to the original document.

3 DEFINITIONS

3.1 Problem Setting

We are given a set on sentences[1] $S_1,...,S_m$ derived from a document or a cluster of related documents speaking on some subject. Meaningful words in these sentences are entirely described by terms $T_1,...,T_n$. Our goal is to find a subset $S_{i_1},...,S_{i_k}$ consisting of sentences such that (1) there are at most N terms in these sentences; (2) term frequency is preserved as much as possible w.r.t. the original sentence set; (3) redundant information among k selected sentences is minimized.

[1]Since an extractive summarization usually deals with sentence extraction, this paper also focuses on sentences. Generally, our method can be used for extracting any other text units like phrases, paragraphs, etc..

3.2 The Matrix Model

We describe sets of sentences and terms by real matrix $A = (a_{i,j})$ of size $n \times m$ where

> $a_{i,j} = k$ if term T_i appears in the sentence S_j precisely k times.

Then columns of A describe sentences and rows describe terms. Since we are not interested in redundant sentences, in the case of multi-document summarization, we can initially select meaningful sentences by clustering all the columns as vectors in $\mathbb{R}^n$ and choose a single representative from each cluster. Then columns describe representatives of sentence clusters.

Here and further, we refer to A as the **sentence-term matrix** corresponding to the given document/s.

Example 1. *Given the following text of $m = 3$ sentences and $n = 5$ (normalized) terms:*

> *S_1 = A fat cat is a cat that eats fat meat.*
> *S_2 = My cat eats fish but he is a fat cat.*
> *S_3 = All fat cats eat fish and meat.*

A matrix corresponding to the text above has the following shape:

$$\begin{array}{l} \\ T_1 = \text{“fat”} \\ T_2 = \text{“cat”} \\ T_3 = \text{“eat”} \\ T_4 = \text{“fish”} \\ T_5 = \text{“meat”} \end{array} \begin{array}{c} \begin{array}{ccc} S_1 & S_2 & S_3 \end{array} \\ \begin{bmatrix} a_{1,1}=2 & a_{1,2}=1 & a_{1,3}=1 \\ a_{2,1}=2 & a_{2,2}=2 & a_{2,3}=1 \\ a_{3,1}=1 & a_{3,2}=1 & a_{3,3}=1 \\ a_{4,1}=0 & a_{4,2}=1 & a_{4,3}=1 \\ a_{5,1}=1 & a_{5,2}=0 & a_{5,3}=1 \end{bmatrix} \end{array}$$

where $a_{i,j}$ are term frequencies.

Let s be the total number of terms in all the sentences. We can derive s from the sentence-term matrix S. Formally, we compute

$$s = \sum_{i=1}^{m} \sum_{j=1}^{n} a_{i,j}$$

Example 2. *For the matrix of Example 1 we have $s = \sum_{i=1}^{5} \sum_{j=1}^{3} a_{i,j} = 16$.*

3.3 Term Frequencies

We can use the sentence-term matrix to compute term frequency for each term. Indeed, for n terms in the document their term count is a real vector C of size n, where $C[i] = k$ stands for term count. Then term frequency is a real vector

$$F = \frac{1}{s} C$$

obtained from C by dividing each of its elements by the total term count.

Computing the vector F requires application of a simple linear transformation to A. We have

$$F^T = \frac{1}{s} A \times J_{m\times 1}$$

where $J_{m\times 1}$ is the all-1 vector. Vector F consists of term frequences $[tf(T_1),\ldots,tf(T_n)]$ and is easily computed using matrix-vector multiplication.

Example 3. *For the matrix of Example 1 we have*

$$C = \begin{bmatrix} 2 & 1 & 1 \\ 2 & 2 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \frac{1}{16} \begin{bmatrix} 4 \\ 5 \\ 3 \\ 2 \\ 2 \end{bmatrix}$$

and $s = 16$. *Then*

$$F^T = \frac{1}{16} C^T = \frac{1}{16} [4\ 5\ 3\ 2\ 2]$$

3.4 The Goal

In this setting, our goal can be reformulated as the problem of finding subset $i_1,\ldots,i_k$ of matrix columns from A, so that for the resulting submatrix A' the distance from F to the vector

$$(F')^T = \frac{1}{s'} A' \times J_{m'\times 1}$$

is as small as possible. Here, the number s' denotes the total count of terms in selected sentences. Distance functions in this case can vary – for example, Manhattan distance, Euclidean distance, cosine similarity, mutual information etc.

4 FROM MATRIX MODEL TO POLYTOPE

4.1 Hyperplanes and Half-spaces

Extractive summarization aims at extracting a subset of sentences that covers as much non-redundant information as possible w.r.t. the source document/documents. Here we introduce a new efficient text representation model with purpose of representing all possible extracts without computing them explicitly. Since the number of potential extracts is exponential in the number of sentences, we would be saving a great portion of computation time. Finding an optimal extract of text units is a general problem for various Information Retrieval tasks like: Question Answering, Literature Search, etc., and our model can be efficiently applied on all these tasks.

In our representation model, each sentence is represented by hyperplane, and all sentences derived from a document form a hyperplane intersections (polytope). Then, all possible extracts can be represented by subplanes of our hyperplane intersections and as such that are not located far from the boundary of the polytope. Therefore, intuitively, the boundary of the resulting polytope is a good approximation for extracts that can be generated from the given document.

4.2 The Approach

We view every column of the sentence-term matrix as a linear constraint representing a hyperplane in $\mathbb{R}^{mn}$. A term t_i in sentence S_j is represented by variable $x_{i,j}$.

Example 4. *This example demonstrates variables corresponding to the* 4×3 *sentence-term matrix A from Example 1.*

$$\begin{array}{c} \\ T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \end{array} \begin{array}{c} \begin{array}{ccc} S_1 & S_2 & S_3 \end{array} \\ \begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,3} \\ x_{2,1} & x_{2,2} & x_{2,3} \\ x_{3,1} & x_{3,2} & x_{3,3} \\ x_{4,1} & x_{4,2} & x_{4,3} \\ x_{5,1} & x_{5,2} & x_{5,3} \end{bmatrix} \end{array}$$

Together all the columns will define a system of linear inequalities, we also express constrains on the number of terms in the extract we seek. Then every sentence in our document is a hyperlane in $\mathbb{R}^{mn}$ expressed with the help of elements in columns $A[][i]$ of A and variables $x_{i,j}$ representing appearances of terms in sentences.

We define linear inequality

$$A[][i] \cdot [x_{i,1},\ldots,x_{i,n}]^T = \sum_{j=1}^{n} a_{j,i} x_{j,i} \le A[][i] \cdot \mathbf{1}^T \quad (1)$$

Every inequality of this form defines a hyperplane H_i and it lower half-space specified by equation (1):

$$H_i := \sum_{j=1}^{n} a_{j,i} x_{j,i} = \sum_{j=1}^{n} a_{j,i}$$

and has normal vector $\tilde{\mathbf{n}}_i$ with

$$\tilde{\mathbf{n}}_i[k] = \begin{cases} a_{j,i} & k = j \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

To express the fact that every term is either present or absent from the chosen extract, we add constraints

$$0 \le x_{i,j} \le 1 \quad (3)$$

Intuitively, a point p on H_i represents a sentence with the same term counts as S_i. To study subsets of sentences, we observe intersections of hyperplanes

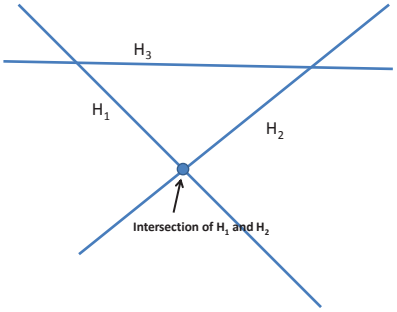

Figure 1: Two-dimensional projection of hyperplane intersection.

H_i. In this case, we say that the intersection of two hyperplanes H_i and H_j represents a set of two sentences S_i and S_j. Then a subset of sentences of size r is represented by intersection of r hyperplanes.

Example 5. *Sentence-term matrix of Example 1 defines the following hyperplane equations.*

$$H_1 := 2x_{1,1} + 2x_{2,1} + x_{3,1} + x_{5,1} = 2 + 2 + 1 + 1 = 6$$
$$H_2 := x_{1,2} + 2x_{2,2} + x_{3,2} + x_{4,2} = 5$$
$$H_3 := x_{1,3} + x_{2,3} + x_{3,3} + x_{4,3} + x_{5,3} = 5$$

Here, a summary consisting of the first and the second sentence is expressed by the intersection of hyperplanes H_1 and H_2. Figure 1 shows how a two-dimensional projection of hyperplanes H_1, H_2, H_3 and their intersections look like.

5 SUMMARIZATION AS A DISTANCE FUNCTION

We express summarization constraints in the form of linear inequalities in $\mathbb{R}^{mn}$, using the columns of the sentence-term matrix A as linear constraints. Maximality constraint on the number of terms in the summary can be easily expressed as a constraint on the sum of term variables $x_{i,j}$. Since we are looking for summaries that consist of at most N terms, we introduce the following linear constraint

$$\sum_{i=1}^{m} \sum_{j=1}^{n} x_{i,j} \leq N \tag{4}$$

Indeed, every variable $x_{i,j}$ stands for a separate term in specific sentence, and we intend for their sum to express the number of terms in selected sentences.

Example 6. *Equation* (4) *for the sentence-term matrix of Example 1 for $N = 10$ has the form*

$$\sum_{i=1}^{4} \sum_{j=1}^{3} x_{i,j} \leq 10$$

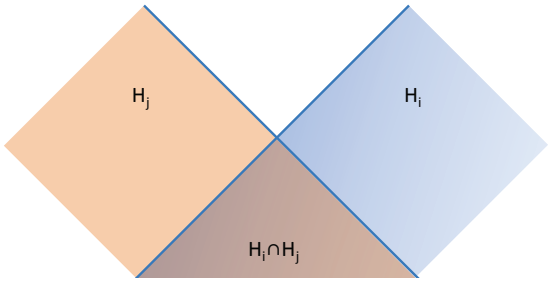

Figure 2: Intersection of hyperplanes.

Having defined linear inequalities that describe each sentence in a document separately and the total number of terms in sentence subset, we can now look at them together as a system:

$$\begin{cases} \sum_{j=1}^{n} a_{1,i} x_{1,j} \leq \sum_{j=1}^{n} a_{j,1} \\ \dots \\ \sum_{j=1}^{n} a_{1,m} x_{m,j} \leq \sum_{j=1}^{n} a_{j,m} \\ \sum_{i=1}^{m} \sum_{j=1}^{n} x_{i,j} \leq N \\ 0 \leq x_{i,j} \leq 1 \end{cases} \tag{5}$$

First m inequalities describe sentences $S_1, \dots, S_m$, and the next inequality describes constraints on the total number of terms in a summary.

Since every inequality in the system (5) is linear, the entire system describes a convex polyhedron in $\mathbb{R}^{mn}$, which we denote by **P**. Faces of **P** are determined by intersections of hyperplanes H_i, $x_1 + \dots + x_n = N$ and $x_i = 0, x_i = 1, y_i = 0, y_i = 1$. Intersections of H_i's represent subsets of sentences (see Figure 2 for illustration), as the following property shows.

Property 1. *Equation of the intersection $H_{1,\dots,k} = H_1 \cap \dots \cap H_k$ (which is a hyperplane by itself) satisfies $\sum_{j=1}^{k} \sum_{i=1}^{n} a_{j,i} x_{j,i} = \sum_{j=1}^{k} \sum_{i=1}^{n} a_{j,i}$.*

Proof. This property is trivial, since the intersection $H_{1,\dots,k}$ has to satisfy all the equations of $H_1, \dots, H_k$. Therefore, summing up equalities (1) for $H_1, \dots, H_k$ we have $\sum_{j=1}^{k} \sum_{i=1}^{n} a_{j,i} x_{j,i} = \sum_{j=1}^{k} \sum_{i=1}^{n} a_{j,i}$. Note that the choice of indexes $1, \dots, k$ was arbitrary and the property holds for any subset of indexes. □

Therefore, the hypersurfaces representing sentence sets we seek are in fact hyperplane intersections that form the boundaries of the polytope **P**.

5.1 Finding the Closest Point

We assume here that the surface of the polyhedron **P** is a suitable representation of all the possible sentence subsets (its size, of course, is not polynomial in m and n since the number of vertices of **P** can be very large). Fortunately, we do not need to scan the whole set of

P's surfaces but rather to find the point on **P** which is the closest to the term frequency we wish to preserve.

We use the fact that the term frequency vector F is in fact a point in $\mathbb{R}^n$. Polytope $\mathbf{P} \subseteq \mathbb{R}^{mn}$, and we need to breach the gap between dimensions of the two spaces.

Let H be a hyperplane representing a summary. Then by Property 1 H is w.l.o.g. an intersection of several hyperplanes $H_1, \ldots, H_k$ of the form (1) and its normal vector $\tilde{\mathbf{n}}$ satisfies the following condition:

$$\tilde{\mathbf{n}}[i,j] = \begin{cases} a_{i,j} & i = 1 \ldots k, \\ 0 & \text{otherwise.} \end{cases}$$

Then term count of term T_i in this summary is precisely

$$\sum_{j=1}^{m} \tilde{\mathbf{n}}[i,j] \tag{6}$$

To find the distance from F to possible summaries as Euclidean distance in $\mathbb{R}^n$, we

- look at linear transformation of **P** that transforms a hyperplane with normal vector $\tilde{\mathbf{n}} = (\tilde{\mathbf{n}}_{i,j})$ into a hyperplane with normal vector

$$\tilde{\mathbf{m}} = (\sum_{j=1}^{m} \tilde{\mathbf{n}}[1,j], \ldots, \sum_{j=1}^{m} \tilde{\mathbf{n}}[n,j]) \tag{7}$$

- observe $F = (f_1, \ldots, f_n)$ as a point in $\mathbb{R}^n$;
- search for points $p = (p_1 := \sum_{j=1}^{m} x_{1,j}, \ldots, p_n := \sum_{j=1}^{m} x_{n,j})$ on the transformed polytope whose distance to F is minimal;
- such a point p is a transformation of point $x = (x_{ij}) \in \mathbf{R}^{mn}$ which lies on the boundary of **P**. It holds that $x_{k,1} = \cdots = x_{k,m} = 1$ if the point belongs to hyperplane H_k.

Distance between the two is computed as

$$d(p,F) = \sqrt{\sum_{i=1}^{n} (f_i - p_i)^2} \tag{8}$$

Since vector F is constant for given document or document collection, values of f_i's are constant and therefore $d(p,F)$ is a quadratic function.

The problem of finding the required summary can now be reformulated as **finding the point on P closest to the point F** (see Figure 3 for illustration). Since our polytope is defined by a system of linear inequalities and the distance from F to **P** is expressed as a quadratic function, the minimum of this function is achieved on the boundary of **P**.

Formally speaking, we are looking for the minimum of the following function under constraints defined in (5).

$$d(\mathbf{P},F) = \sqrt{\sum_{i=1}^{n} (\sum_{j=1}^{m} x_{i,j}) - f_i^2} \tag{9}$$

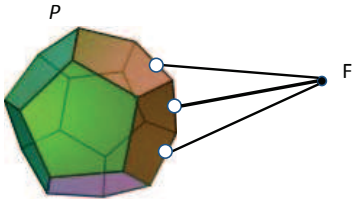

Figure 3: Distance from F to **P**.

Minimizing function (9) is a quadratic programming problem. Therefore, the original summarization problem can be posed as a quadratic programming problem of minimizing the function $d(\mathbf{P},F)$ under constraints of (5). Quadratic programming problem of this type can be solved efficiently both theoretically and practically (see (Karmarkar, 1984; Khachiyan, 1996; Berkelaar, 1999)).

5.2 Extracting the Summary

Since the LP method not only finds the minimal distance but also presents an evidence to that minimality in the form of a point $x = (x_{i,j})$, we use the point's data to find what sentences belong to the chosen summary. Viewing x as a matrix, we check whether or not each column $x[][i]$ equals to **1**. If this equality holds, x lies on an intersection of hyperplanes that includes H_i and therefore sentence S_i is contained in the summary. Otherwise, S_i does not belong to the chosen summary. This test is straightforward and takes $O(mn)$ time.

Applying of our method with lp-solve (Berkelaar, 1999) to Example 1 under length constraint of 11 terms resulted in a summary consisting of the first and the second sentences.

5.3 Supervised Approach

As described above, finding the closest point is relative to the term frequency we wish to preserve. Trying to preserve original term frequency of the source document, we get *unsupervised* method. Given gold standard summaries, we can train our model to find the closest point to the term frequency of gold standard and apply a trained model on new documents, as in *supervised* method.

5.4 Text Preprocessing

In order to build the matrix and then the polytope model, one needs to perform the basic text preprocessing including sentence splitting and tokenization. Also, such additional steps like stopwords removal, stemming, synonym resolution, etc. may be performed for resource-rich languages. Since the main

purpose of these methods is to reduce the matrix dimensionality, the resulted model will be more efficient.

6 CONCLUSIONS AND FUTURE WORK

In this paper we present a linear programming model for the problem of extractive summarization. We represent the document as a sentence-term matrix whose entries contain term count values and view this matrix as a set of intersecting hyperplanes. Every possible summary of a document is represented as an intersection of two or more hyperlanes, and one additional constraint is used to limit the number of terms used in a summary. We consider the summary to be the best if term frequency is preserved during summarization, and in this case the summarization problem translates into a problem of finding a point on a convex polytope (defined by linear inequalities) which is the closest to the hyperplane describing overall term frequencies in the document.

Linear programming problem can be solved in polynomial time (see (Karmarkar, 1984), (Khachiyan, 1996)). Numerous packages and applications are available, such as (Berkelaar, 1999), (Makhorin, 2000) etc. In future research, we plan to implement and test our approach, as in *unsupervised* as in *supervised* learning. Also, we'd like to extend our model to query-based summarization by adapting the distance function and apply our text representation model to such text mining tasks like text clustering and text categorization.

ACKNOWLEDGEMENTS

The authors thank Ruvim Lipyansky for ideas that led to development of their approach.

REFERENCES

Alfonseca, E. and Rodriguez, P. (2003). Generating extracts with genetic algorithms. In *Proceedings of the 2003 European Conference on Information Retrieval (ECIR'2003)*, pages 511–519.

Berkelaar, M. (1999). lp-solve free software. http://lpsolve.sourceforge.net/5.5/.

Filatova, E. (2004). Event-based extractive summarization. In *In Proceedings of ACL Workshop on Summarization*, pages 104–111.

Gillick, D. and Favre, B. (2009). A Scalable Global Model for Summarization. In *Proceedings of the NAACL HLT Workshop on Integer Linear Programming for Natural Language Processing*, pages 10–18.

Hassel, M. and Sjobergh, J. (2006). Towards holistic summarization: Selecting summaries, not sentences. In *Proceedings of LREC - International Conference on Language Resources and Evaluation*.

Hitoshi Nishikawa, Takaaki Hasegawa, Y. M. and Kikui, G. (2010). Opinion Summarization with Integer Linear Programming Formulation for Sentence Extraction and Ordering. In *Coling 2010: Poster Volume*, pages 910–918.

Karmarkar, N. (1984). New polynomial-time algorithm for linear programming. *Combinatorica*, 4:373–395.

Khachiyan, L. G. (1996). Rounding of polytopes in the real number model of computation. *Mathematics of Operations Research*, 21:307–320.

Khuller, S., Moss, A., and Naor, J. S. (1999). The budgeted maximum coverage problem. *Information Precessing Letters*, 70(1):39–45.

Litvak, M., Last, M., and Friedman, M. (2010). A new approach to improving multilingual summarization using a Genetic Algorithm. In *ACL '10: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 927–936.

Liu, D., Wang, Y., Liu, C., and Wang, Z. (2006). Multiple Documents Summarization Based on Genetic Algorithm. In *Fuzzy Systems and Knowledge Discovery*, volume 4223 of *Lecture Notes in Computer Science*, pages 355–364.

Makhorin, A. O. (2000). GNU Linear Programming Kit. http://www.gnu.org/software/glpk/.

Makino, T., Takamura, H., and Okumura, M. (2011). Balanced coverage of aspects for text summarization. In *TAC '11: Proceedings of Text Analysis Conference*.

Mani, I. and Maybury, M. (1999). *Advances in Automatic Text Summarization*. MIT Press, Cambridge, MA.

Ouyang, Y., Li, W., Li, S., and Lu, Q. (2011). Applying regression models to query-focused multi-document summarization. *Information Processing and Management*, 47:227–237.

Salton, G., Yang, C., and Wong, A. (1975). A vector-space model for information retrieval. *Communications of the ACM*, 18.

Takamura, H. and Okumura, M. (2009). Text summarization model based on maximum coverage problem and its variant. In *EACL '09: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 781–789.

Woodsend, K. and Lapata, M. (2010). Automatic Generation of Story Highlights. In *ACL '10: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 565–574.