# Multilingual Multi-Document Summarization with POLY$^2$

**Marina Litvak**
Department of Software Engineering
Shamoon College of Engineering
Beer Sheva, Israel
marinal@sce.ac.il

**Natalia Vanetik**
Department of Software Engineering
Shamoon College of Engineering
Beer Sheva, Israel
natalyav@sce.ac.il

## Abstract

In this paper we present a linear model for the problem of text summarization, where a summary preserves the information coverage as much as possible in comparison to the original document set. We reduce the problem of finding the best summary to the problem of finding the point on a convex polytope closest to the given hyperplane, and solve it efficiently with the help of fractional linear programming. We supply here an overview of our system, titled POLY$^2$, that participated in the MultiLing contest at ACL 2013.

## 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).

Some authors reduce summarization to the maximum coverage problem (Takamura and Okumura, 2009; Gillick and Favre, 2009) that, despite a great performance, is known as NP-hard (Khuller et al., 1999). Linear Programming helps to find an accurate approximated solution to this 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). However, most mentioned works use exponential number of constraints.

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 linear programming in polynomial time. We measure information coverage by terms[1] and strive to obtain a summary that preserves the optimal value of the chosen objective function as much as possible in comparison to the original document. Various objective functions combining different parameters like term's position and its frequency are introduced and evaluated.

Our method ranks and extracts significant sentences into a summary and it can be generalized for both single-document and multi-document summarization. Also, it can be easily adapted to cross-lingual/multilingual summarization.

Formally speaking, in this paper we introduce (1) a novel text representation model expanding a classic Vector Space Model (Salton et al., 1975) to Hyperplane and Half-spaces, (2) re-formulated extractive summarization problem as an optimization task and (3) its solution using linear 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.

## 2 Our Method

### 2.1 Definitions

We are given a set of sentences $S_1, ..., S_n$ derived from a document or a cluster of related documents. Meaningful words in these sentences are entirely described by terms $T_1, ..., T_m$. 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.

We use the standard sentence-term matrix, $A = (a_{ij})$ of size $m \times n$, for initial data representation, where $a_{ij} = k$ if term $T_i$ appears in the sentence $S_j$ precisely $k$ times.

Our goal is to find subset $i_1, ..., i_k$ of $A$'s

[1] normalized meaningful words



*Proceedings of the MultiLing 2013 Workshop on Multilingual Multi-document Summarization*, pages 45–49,
Sofia, Bulgaria, August 9 2013. 

columns so that the chosen submatrix represents the best possible summary under some constraints. Since it is hard to determine what is the best summary mathematically (this task is usually left to human experts), we wish to express summary quality as a linear function of the underlying matrix. We strive to find a summary that gives an optimal value once the function in question has been determined.

Basic text preprocessing includes sentence splitting and tokenization. Also, additional steps like stopwords removal, stemming, synonym resolution, etc. may be performed for resource-rich languages.

### 2.2 Polytope as a document representation

We represent every sentence by a hyperplane and the lower half-space of that hyperplane. In a way, the hyperplane bounding each half-space is the sentence itself, and a half-space below it is an approximation of that sentence. An intersection of lower half-spaces in Euclidean space forms a convex polyhedron, and in our case the faces of this polyhedron are intersections of hyperplanes bounding lower half-spaces that stand for document sentences. We add trivial constraints so that the polyhedron representing the entire document is bounded, i.e. is a *polytope*. All possible extracts from the document can be represented by hyperplane intersections. Thus the boundary of the resulting polytope is a good approximation for extracts that can be generated from the given document.

We view every column of the sentence-term matrix as a *linear constraint* representing a lower half-space in $\mathbb{R}^{mn}$. An occurrence of term $t_i$ in sentence $S_j$ is represented by variable $x_{ij}$. The maximality constraint on the number of terms in the summary can be easily expressed as a constraint on the sum of these variables. Note that each sentence constraint uses its $n$ unique variables, thus making sure that the intersection of every subset of sentence hyperplane is not empty and is well-defined.

Every sentence $S_j$ in our document is a lower half-space of a hyperlane in $\mathbb{R}^{mn}$, defined with columns of $A$ and variables $x_{1j}, \ldots, x_{mj}$ representing the terms in this sentence:

$$A[][j] = [a_{1j}, \ldots, a_{mj}]$$
$$\mathbf{x}_j = [x_{1j}, \ldots, x_{mj}] \text{ for all } 1 \leq j \leq n$$

We define a system of linear inequalities

$$\begin{array}{l} A[][j] \cdot \mathbf{x}_j^T = \sum_{i=1}^m a_{ij} x_{ij} \leq \\ \leq A[][j] \cdot \mathbf{1}^T = \sum_{i=1}^m a_{ij} \end{array} \quad (1)$$

Every inequality of this form defines a lower half-space of a hyperplane

$$H_i := (A[][j] \cdot \mathbf{x}_j^T = A[][j] \cdot \mathbf{1}^T)$$

To say that every term is either present or absent from the chosen extract, we add constraints $0 \leq x_{ij} \leq 1$. Intuitively, the entire hyperplane $H_i$ and therefore every point $p \in H_i$ represents sentence $S_i$. Then a subset of $r$ sentences is represented by intersection of $r$ hyperplanes.

### 2.3 Summary constraints

We express summarization constraints in the form of linear inequalities. Maximality constraint on the number of terms in the summary can be easily expressed as a constraint on the sum of all term variables $x_{ij}$, and the same goes for minimality constraint.

### 2.4 The polytope model

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:

$$\begin{cases} \sum_{i=1}^m a_{i1} x_{i1} \leq \sum_{i=1}^m a_{i1} \\ \quad \cdots \\ \sum_{i=1}^m a_{in} x_{in} \leq \sum_{i=1}^m a_{in} \\ T_{\min} \leq \sum_{i=1}^m \sum_{j=1}^n x_{ij} \leq T_{\max} \\ W_{\min} \leq \sum_{i=1}^m \sum_{j=1}^n a_{ij} x_{ij} \leq W_{\max} \\ 0 \leq x_{ij} \leq 1 \end{cases} \quad (2)$$

First $n$ inequalities describe sentences $S_1, \ldots, S_n$, the next two inequalities describes constraints on the total number of terms and words in a summary, and the final constraint determines upper and lower boundaries for all sentence-term variables. Intersections of $S_1, \ldots, S_n$ are well-defined, since every pair of sentences is described by a linear constraints on different $n$-tuple of variables $x_{ij}$ out of total $mn$ variables. Since every inequality in the system (2) is linear, the entire system describes a convex polyhedron, which we denote by $\mathbf{P}$.

### 2.5 Objectives and summary extraction

We assume here that the surface of the polyhedron $\mathbf{P}$ is a suitable representation of all the possible

| Function | Formula | Description |
|---|---|---|
| Maximal Weighted Term Sum ($OBJ_1$) | $\max \sum_{i=1}^{m} w_i t_i$, $t_i = \sum_{j=1}^{n} x_{ji}$ | Maximizes the information coverage as a weighted term sum. We used the following types of term weights $w_i$. (1) POS_EQ, where $w_i = 1$ for all $i$; (2) POS_F, where $w_i = \frac{1}{app(i)}$ and $app(i)$ is the index of a sentence in the document where the term $T_i$ first appeared; (3) POS_B, where $w_i = \max\{\frac{1}{app(i)}, \frac{1}{n-app(i)+1}\}$; (4) TF, where $w_i = \mathrm{tf}(i)$ and $\mathrm{tf}(i)$ is the term frequency of term $T_i$; (5) TFISF, where $w_i = \mathrm{tf}(i) * \mathrm{isf}(i)$ and $isf(i)$ is the inverse sentence frequency of $T_i$ |
| Distance Function ($OBJ_2$) | $\min \sum_{i=1}^{m} (\widehat{t_i} - p_i)^2$, (1) $\widehat{t_i} = t_i = \sum_{j=1}^{n} x_{ji}$ and $\forall i\ p_i = 1$, or (2) $\widehat{t_i} = \frac{t_i}{\sum_{j=1}^{m} t_j}$ and $p_i = \mathrm{tf}(i)$ | Minimizes the Eucledian distance between terms $t = (t_1, \ldots, t_m)$ (a point on the polytope $\mathbf{P}$ representing a generated summary) and the vector $p = (p_1, \ldots, p_m)$ (expressing document properties we wish to preserve and representing the "ideal" summary). We used the following options for $t$ and $p$ representation. (1) MTC, where $t$ is a summary term count vector and $p$ contains all the terms precisely once, thus minimizing repetition but increasing terms coverage. (2) MTF, where $t$ contains term frequency of terms in a summary and $p$ contains term frequency for terms in documents. |
| Sentence Overlap ($OBJ_3$) | $\min \sum_{j=1}^{n} \sum_{k=j+1}^{n} ovl_{jk}$, $ovl_{jk} = \frac{\lvert S_j \cap S_k \rvert}{\lvert S_j \cup S_k \rvert} = = \frac{\sum_{i=1}^{m} w(a_{ij}, a_{ik})(x_{ij} + x_{ik})}{\sum_{i=1}^{m} (a_{ij} + a_{ik})}$ | Minimizes the Jaccard similarity between sentences in a summary (denoted by $ovl_{jk}$ for $S_j$ and $S_k$). $w(a_{ij}, a_{ik})$ is 1 if the term $T_i$ is present in both sentences $S_j$ and $S_k$ and is 0 otherwise. |
| Maximal Bigram Sum ($OBJ_4$) | $\max \sum_{i,j} \mathrm{bi}_{ij}$, where $\forall i, j\ 0 \leq \mathrm{bi}_{ij} \leq 1$ | Maximizes the information coverage as a bigram sum. Variable $\mathrm{bi}_{ij}$ is defined for every bigram $(T_i, T_j)$ in the text. |

Table 1: Objective functions for summarization using polytope model.

sentence subsets. Fortunately, we do not need to scan the whole set of $\mathbf{P}$'s surfaces but rather to find the point on $\mathbf{P}$ that optimizes the chosen objective function. Table 1 contains four different objective functions that we used for summarization, along with descriptions of the changes in the model that were required for each function.

Since fractional LP method not only finds the optimal value of objective function but also presents an evidence to that optimality in the form of a point $x = (x_{ij})$, we use the point's data to find what sentences belong to the chosen summary. The point $x$ may satisfy several of the sentence inequalities as equalities, while other inequalities may not turn into equalities. If sentence inequality is turned into equality by $x$, the sentence necessarily belongs to the chosen summary. Otherwise, the point $x$ that optimizes the chosen objective function represents a summary that does not contain sufficient number of words or terms. In this case we add additional sentences to the summary and we choose these sentences on the basis of their distance from the point $x$. Sentence hyperplanes that are the nearest to the point $x$ are chosen in a greedy manner and added to the summary. This test is straightforward and takes $O(mn)$ time.

## 3 POLY$^2$: system description

We title our system POLY$^2$ as a double POLY occured in "POLYnomial Summarization using POLYtope model". POLY$^2$ is implemented in Java and it uses lp-solve software (Berkelaar, 1999) in order to perform Linear Programming. The input for our system is initial collection of texts to be summarized. Four main parts of POLY$^2$ are: preprocessing, linear program generation, linear program application and testing. Data flow of the system is depicted in Figure 1.

The **preprocessing step** consists of following parts. During the very first step, initial documents undergo stop word removal, stemming and synonym resolution (if available for the chosen language). Then, a sentence-term matrix is generated in the form of a text file, where every line describes a term and every column – a sentence. Also, the index list for multi-document summarization is generated. In this list serial number of each sentence is paired up with its serial number in its document. This information is used later in order to decide how close each sentence is to its document's boundaries. Finally, we generate list of bigrams (a consecutive appearance of two terms) for every sentence. All of the files gener-

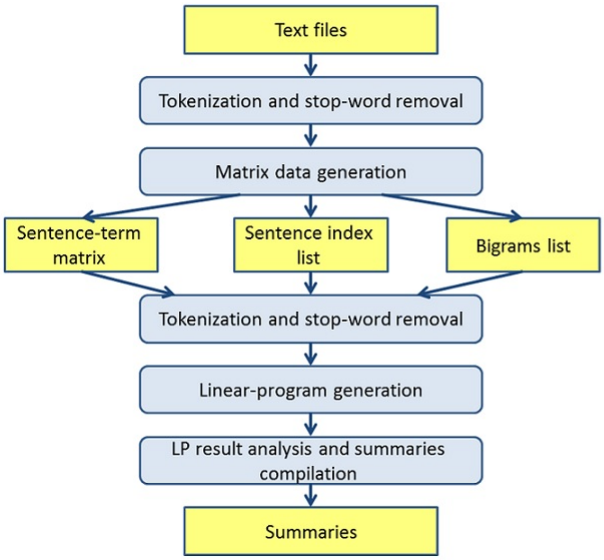


Figure 1: Data flow of the POLY$^2$ system.

```
max :
tm0+tm1+tm2+tm3+tm4+tm5+tm6+tm7+tm8+tm9+tm10+tm11+tm12
10tm0=tx0_0+tx0_12+tx0_38+tx0_45+tx0_50+tx0_55+tx0_63+tx0_92+
tm1=tx1_0;
tm2=tx2_0;
tm3=tx3_0;
tm4=tx4_0;
tm5=tx5_0;
64tm6=tx6_0+tx6_6+tx6_12+tx6_17+tx6_18+tx6_19+tx6_20+tx6_21+t
tm7=tx7_0;
tm8=tx8_0;
tm9=tx9_0;
tm10=tx10_0;
```

Figure 2: Linear program generated by POLY$^2$ system.

ated during preprocessing are text files.

The main part of POLY$^2$ is **linear program generation**. The system allows to select document matrix files and auxiliary files and objective function and generates a system of linear inequalities together with an objective function in format acceptable by lp-solve software. Figure 2 shows a sample LP file contents. Note that file generation for every one of our objective function takes only seconds for a single documents.

The next step is to **run linear program** and extract its results. We use lp-solve Java API to perform this task and extract coordinates of point $x$ that optimizes the chosen objective function. In order to construct the summary, we measure the normalized distance from point $x$ to every one of the sentence hyperplanes. Since this information is readily available through lp-solve API, the task requires sorting of $n$ real numbers, where $n$ is the number of sentences in all of the documents together. Hyperplanes whose distance to $x$ is minimal represent the sentences participating in the final summary. Running time of this step for a single file does not exceed three seconds.

The final (optional) step is to verify generated summaries, that can be performed with the help of any evaluation system. In our case, the ROUGE (Lin, 2004) system has been used.

## 4 Conclusions and Future Work

In this paper we present an extractive summarization system POLY$^2$ based on a linear programming model. We represent the document as a set of intersecting hyperplanes. Every possible summary of a document is represented as the intersection of two or more hyperlanes. We consider the summary to be the best if the optimal value of objective function is preserved during summarization, and translate the summarization problem into a problem of finding a point on a convex polytope which is the closest to the hyperplane describing the "ideal" summary. We introduce multiple objective functions describing the distance between a summary (a point on a convex polytope) and the best summary (the hyperplane). Since the introduced objectives behaves differently on different languages, only two of them were indicated as primary systems and evaluated by MultiLing 2013 organizers–$OBJ_1^{POS\_F}$ and $OBJ_3$–denoted by ID5 and ID51.

Below we summarize the results of automated evaluations in MultiLing 2013 (ROUGE-1,2,3,4 and three N-gram graph methods) for POLY$^2$ in three languages.

**English**: $7^{th}$ place in all ROUGE metrics (stemmed) and AutoSummENG, and $8^{th}$ place in MeMoG and NPowER (out of 10 systems);

**Hebrew**: $3^{rd}$ place in ROUGE-1, $5^{th}$ place in ROUGE-2 and MeMoG, $4^{th}$ rank in ROUGE-3 and ROUGE-4, and only $6^{th}$ rank in terms of AutoSummENG and NPowER (out of 9 participants);

**Arabic**: $7^{th}$ rank in ROUGE-1,2 and all NGG metrics, $6^{th}$ rank in terms of ROUGE-3, and $5^{th}$ place in ROUGE-4 (out of 10 summarizers).

As it can be seen, the best performance for POLY$^2$ has been achieved on the dataset of Hebrew documents.

Since fractional linear programming problem can be solved in polynomial time (Karmarkar, 1984), the time complexity of our approach is polynomial.

## Acknowledgments

Authors thank Igor Vinokur for the software maintenance and running of experiments.

## References

Michel Berkelaar. 1999. lp-solve free software. `http://lpsolve.sourceforge.net/5.5/`.

Dan Gillick and Benoit Favre. 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.

Yoshihiro Matsuo Hitoshi Nishikawa, Takaaki Hasegawa and Genichiro Kikui. 2010. Opinion Summarization with Integer Linear Programming Formulation for Sentence Extraction and Ordering. In *Coling 2010: Poster Volume*, pages 910–918.

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

L. G. Khachiyan and M. J. Todd. 1993. On the complexity of approximating the maximal inscribed ellipsoid for a polytope. *Mathematical Programming*, 61:137–159.

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

Samir Khuller, Anna Moss, and Joseph (Seffi) Naor. 1999. The budgeted maximum coverage problem. *Information Precessing Letters*, 70(1):39–45.

Chin-Yew Lin. 2004. Rouge: a package for automatic evaluation of summaries. In *Workshop on Text Summarization Branches Out, Post-Conference Workshop of ACL 2004*, pages 25–26.

Takuya Makino, Hiroya Takamura, and Manabu Okumura. 2011. Balanced coverage of aspects for text summarization. In *TAC '11: Proceedings of Text Analysis Conference*.

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

Hiroya Takamura and Manabu Okumura. 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.

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