

Mining the Gaps: Towards Polynomial Summarization

Marina Litvak

Sami Shamoon College of Engineering
Beer Sheva, Israel
marinal@sce.ac.il

Natalia Vanetik

Sami Shamoon College of Engineering
Beer Sheva, Israel
natalyav@sce.ac.il

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 summarization¹, 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 (polynomial-time) linear programming. The experimental results show the superiority of our approach over most of the systems participating in the generic multi-document summarization task (MultiLing) of the TAC 2011 competition.

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.,

¹This work was partially funded by U.S. Department of Navy, Office of Naval Research.

2011). However, most mentioned works use exponential number of constraints or Integer Linear Programming which is an NP-hard problem.

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 terms² 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 or 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.

2 Our Method

2.1 Definitions

We are given a set of sentences S_1, \dots, S_n derived from a document or a cluster of related documents. Meaningful words in these sentences are entirely described by terms T_1, \dots, T_m . Our goal is to find a

²normalized meaningful words

subset S_{i_1}, \dots, 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. Here, 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. In this case columns of A describe representatives of sentence clusters. The total number of words (term appearances) in the document, denoted by S , can be computed from the matrix A as

$$S = \sum_i \sum_j a_{ij} \quad (1)$$

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

$S_1 = A \text{ fat cat is a cat that eats fat meat.}$
 $S_2 = My \text{ cat eats fish but he is a fat cat.}$
 $S_3 = All \text{ fat cats eat fish and meat.}$

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

$$\begin{array}{l} T_1 = \text{"fat"} \\ T_2 = \text{"cat"} \\ T_3 = \text{"eat"} \\ T_4 = \text{"fish"} \\ T_5 = \text{"meat"} \end{array} \begin{bmatrix} S_1 & S_2 & S_3 \\ a_{11} = 2 & a_{12} = 1 & a_{13} = 1 \\ a_{21} = 2 & a_{22} = 2 & a_{23} = 1 \\ a_{31} = 1 & a_{32} = 1 & a_{33} = 1 \\ a_{41} = 0 & a_{42} = 1 & a_{43} = 1 \\ a_{51} = 1 & a_{52} = 0 & a_{53} = 1 \end{bmatrix}$$

where a_{ij} are term counts. The total count of terms in this matrix is

$$S = \sum_{i=1}^5 \sum_{j=1}^3 a_{ij} = 16$$

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

2.2 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, 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.

2.3 Polytope as a document representation

We represent every sentence by a 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. Intuitively, 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 hyperplane in \mathbb{R}^m . 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.

Example 2. This example demonstrates variables corresponding to the 5×3 matrix A of Example 1.

$$\begin{array}{l} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \end{array} \begin{bmatrix} S_1 & S_2 & S_3 \\ x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \\ x_{41} & x_{42} & x_{43} \\ x_{51} & x_{52} & x_{53} \end{bmatrix}$$

Every sentence in our document is a hyperplane in \mathbb{R}^m , defined with columns of A and variables representing terms in sentences:

$$\begin{aligned} A[[j]] &= [a_{1j}, \dots, a_{mj}] \\ \mathbf{x}_j &= [x_{1j}, \dots, x_{mj}] \text{ for all } 1 \leq j \leq n \end{aligned}$$

We define a system of linear inequalities

$$\begin{aligned} A[[j]] \cdot \mathbf{x}_j^T &= \sum_{i=1}^m a_{ij} x_{ij} \leq \\ &\leq A[[j]] \cdot \mathbf{1}^T = \sum_{i=1}^m a_{ij} \end{aligned} \quad (2)$$

Every inequality of this form defines a hyperplane H_i and its lower half-space specified by equation (2):

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

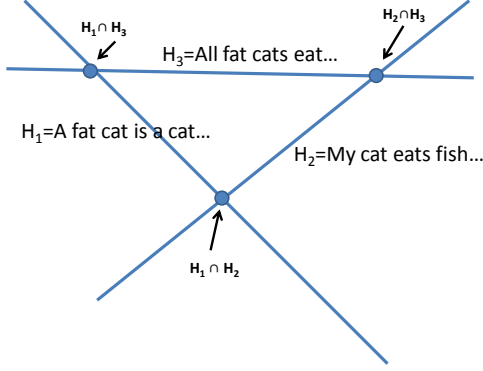


Figure 1: Two-dimensional projection of hyperplane intersection.

and with normal vector $\mathbf{n} = (\mathbf{n}_{xy})$

$$\mathbf{n}_{xy} = \begin{cases} a_{xy} & 1 \leq x \leq m \wedge y = j \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

To say that every term is either present or absent from the chosen extract, we add constraints $0 \leq x_{ij} \leq 1$. Intuitively, 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.

Example 3. Sentence-term matrix A of Example 1 defines the following hyperplane equations.

$$H_1 : 2x_{11} + 2x_{21} + x_{31} + x_{51} = 2 + 2 + 1 + 1 = 6$$

$$H_2 : x_{12} + 2x_{22} + x_{32} + x_{42} = 5$$

$$H_3 : x_{13} + x_{23} + x_{33} + x_{43} + x_{53} = 5$$

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.

2.4 Summary constraints

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_{ij} .

$$\sum_{i=1}^m \sum_{j=1}^n x_{ij} \leq T_{\max} \quad (4)$$

Example 4. Equation (4) for Example 1, $T_{\max} = 11$ has the form

$$\begin{aligned} 0 \leq x_{ij} \leq 1, \forall i, j \\ \sum_{i=1}^5 \sum_{j=1}^3 x_{ij} \leq 11 \end{aligned}$$

Additionally, we may have constraints on the maximal W_{\max} number of words in the summary. We take into account only words that remain in the text after stop-word removal and stemming. The difference between the number terms and the number of words in a summary is that a single term can appear more than once in a sentence. Therefore, the total number of words in the text is expressed by summing up the elements of its term-count matrix. Therefore, maximality constraints for words are expressed by the following linear inequality.

$$\sum_{i=1}^m \sum_{j=1}^n a_{ij} x_{ij} \leq W_{\max} \quad (5)$$

Example 5. Equation (5) for the sentence-term matrix of Example 1 for $W_{\max} = 11$ has the form

$$\begin{aligned} 2x_{11} + 2x_{21} + x_{31} + x_{51} + \\ + x_{12} + x_{22} + 2x_{32} + x_{42} + \\ + x_{13} + x_{23} + x_{33} + x_{43} + x_{53} \leq 11 \end{aligned}$$

2.5 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:

$$\begin{cases} \sum_{i=1}^m a_{i1} x_{i1} \leq \sum_{i=1}^m a_{i1} \\ \dots \\ \sum_{i=1}^m a_{in} x_{in} \leq \sum_{i=1}^m a_{in} \\ \sum_{i=1}^m \sum_{j=1}^n x_{ij} \leq T_{\max} \\ \sum_{i=1}^m \sum_{j=1}^n a_{ij} x_{ij} \leq W_{\max} \\ 0 \leq x_{ij} \leq 1 \end{cases} \quad (6)$$

First n inequalities describe sentences S_1, \dots, 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. Since every inequality in the system (6) is linear, the entire system describes a convex polyhedron in \mathbb{R}^{mn} , which we denote by \mathbf{P} . Faces of \mathbf{P} are determined by intersections of hyperplanes defined in (6).

2.6 Objectives and summary extraction

We assume here that the surface of the polyhedron \mathbf{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 \mathbf{P} can reach $O(2^n)$). Fortunately, we do not need to scan the whole set of \mathbf{P} 's surfaces but rather to find

Function	Formula	Description
Maximal Weighted Term Sum (<i>OBJ</i> 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 $w_i = tf(i)$ and $tf(i)$ is the term frequency of term T_i ; (5) TFISF, where $w_i = tf(i) * isf(i)$ and $isf(i)$ is the inverse sentence frequency of T_i
Distance Function (<i>OBJ</i> ₂)	$\min \sum_{i=1}^m (\hat{t}_i - p_i)^2,$ (1) $\hat{t}_i = t_i = \sum_{j=1}^n x_{ji}$ and $\forall i p_i = 1$, or (2) $\hat{t}_i = \frac{t_i}{\sum_{j=1}^n t_j}$ and $p_i = tf(i)$	Minimizes the Euclidian distance between terms $t = (t_1, \dots, t_m)$ (a point on the polytope \mathbf{P} representing a generated summary) and the vector $p = (p_1, \dots, p_m)$ (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 (<i>OBJ</i> ₃)	$\min \sum_{j=1}^n \sum_{k=j+1}^n ovl_{jk},$ $ovl_{jk} = \frac{ S_j \cap S_k }{ S_j \cup S_k } = \frac{\sum_{i=1}^m w(a_{ij}, a_{ik})(x_{ij} + x_{ik})}{\sum_{i=1}^m (a_{ij} + a_{ik})}$	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 (<i>OBJ</i> ₄)	$\max \sum_{i,j} bi_{ij},$ where $\forall i, j, 0 \leq bi_{ij} \leq 1$	Maximizes the information coverage as a bigram sum. Variable bi_{ij} is defined for every bigram (T_i, T_j) in the text.

Table 1: Objective functions for summarization using polytope model.

the point on \mathbf{P} that optimizes the chosen objective function. Table 1 contains four different objective functions³ that we used for summarization, along with descriptions of the changes in the model that were required for each function.

Since the LP method not only finds the minimal distance but also presents an evidence to that minimality in the form of a point $x = (x_{ij})$, we use the point's data to find what sentences belong to the chosen summary. We check which equations of H_i the point x satisfies as equalities. If an equality holds, x lies on H_i and therefore the sentence S_i is contained in the summary. This test is straightforward and takes $O(mn)$ time. In a case of insufficient summary length, the sentences nearest to the point x are extracted to a summary in a greedy manner.

3 Experiments

In order to evaluate the quality of our approach, we compared our approach to multiple summarizers participated in the generic multi-document summarization task of the TAC 2011 competition (Giannakopoulos et al., 2011) and human

³Since our approach is unsupervised, there is no possibility and meaning to use ROUGE, that needs Gold Standard, as an objective.

performance as well. Our software was implemented in Java using Ipsolve (Berkelaar, 1999)⁴. We used the following objective functions, described in Table 1.

- (1) Maximal weighted term sum $OBJ_1^{weight_type}$, where $weight_type$ is one of POS_EQ, POS_F, POS_B, TF, TFISF;
- (2) Minimal distance $OBJ_2^{vector_type}$, where $vector_type$ is either MTC (Maximal Term Coverage) or MTF (Maximal Term Frequency);
- (3) Minimal sentence overlap OBJ_3 ;
- (4) Maximal bigram sum OBJ_4 .

We conducted the experiments on the MultiLing 2011 (Giannakopoulos et al., 2011) English dataset. MultiLing dataset consists of 10 document sets, 10 documents each one, in seven languages. The original news articles in English were taken from WikiNews⁵, organized into 10 sets, and then summarized. According to the MultiLing summarization task, all systems must generate summaries in size of 250 words at most. Eight systems (ID1-ID8) participated in the pilot and compared to the global baseline (ID9) and the global topline (ID10) systems. Systems A,B and C de-

⁴The software is available upon request.

⁵<http://en.wikinews.org/wiki/>

note summaries manually created by human experts. The choice of this dataset is argued by future plans to adapt and evaluate the introduced system to multiple languages.

The automatic summarization evaluation package, ROUGE (Lin, 2004), is used to evaluate the effectiveness of our approach vs. 10 summarizers participated in the MultiLing pilot of the TAC 2011 competition. For fair comparison, only first 250 words⁶ were considered in ROUGE statistics. The recall scores of ROUGE- N for $N \in \{1, 2, 3, 4\}$, ROUGE- W -1.2, and ROUGE- SU 4 which are based on N -gram, Weighted Longest Common Subsequence (WLCS), and Skip-bigram plus unigram, with maximum skip-distance of 4, matching between system summaries and reference summaries, respectively, are reported in Table 2 below.

3.1 Experimental Results

As it can be seen from Table 2, our model using unweighted term sum ($OBJ_1^{POS.EQ}$) as an objective function outperforms most of the systems – 6 systems in terms of ROUGE-1, ROUGE-2, ROUGE- SU 4 and ROUGE- W -1.2, and 8 systems in terms of ROUGE-3 and ROUGE-4. Conversely to our expectations, adding any type of weights to OBJ_1 reduces its performance. Minimizing repetition while increasing terms coverage (OBJ_2^{MTC}) shares the same rank with $OBJ_1^{POS.EQ}$ for most ROUGE metrics. Minimizing distance to a document term frequency vector (OBJ_2^{MTF}) performs worse – it outperforms 3, 4, 5, 6, 5 and 3 systems in terms of ROUGE-1, ROUGE-2, ROUGE-3, ROUGE-4, ROUGE- SU 4 and ROUGE- W -1.2, respectively. Sentence overlap (OBJ_3) and maximal bigram sum (OBJ_4) have very close scores, outperforming 3, 5, 5, 6, 5 and 3 systems in terms of ROUGE-1, ROUGE-2, ROUGE-3, ROUGE-4, ROUGE- SU 4 and ROUGE- W -1.2, respectively. Generally, optimizing the most of introduced functions generates the near-quality summaries. All functions perform better than the baseline (ID9) system.⁷

4 Conclusions and Future Work

In this paper we present a linear programming model for the problem of extractive summariza-

⁶ROUGE.pl -a -x -2 4 -u -c 95 -e data -r 1000 -n 4 -f A -p 0.5 -t 0 -d -l 250

⁷We did not perform tests of statistical significance due to too many comparisons (10 systems vs. 10 objective functions), leaving it as a future work.

tion. We represent the document as a set of intersecting hyperplanes. Every possible summary of a document is represented as an intersection of two or more hyperplanes. 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 linear programming problem can be solved in polynomial time (see (Karmarkar, 1984), (Khachiyan, 1996; Khachiyan and Todd, 1993)), the time complexity of our approach is polynomial (quadratic, being more precise).

The results of experiments show that our method outperforms most of the systems participated in the MultiLing pilot in terms of various ROUGE metrics. In future, we intend to (1) improve the system's performance by introducing more objective functions and their combinations, (2) adapt our system to multiple languages, and (3) extend our model to query-based summarization.

Acknowledgments

Authors thank Igor Vinokur for implementing the introduced approach and performing experiments.

REFERENCES

- Berkelaar, M. (1999). lp-solve free software. <http://lpsolve.sourceforge.net/5.5/>.
- Giannakopoulos, G., El-Haj, M., Favre, B., Litvak, M., Steinberger, J., and Varma, V. (2011). TAC 2011 MultiLing Pilot Overview. In *TAC 2011: Proceedings of Text Analysis Conference*.
- 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.
- 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.

system	rouge-1	system	rouge-2	system	rouge-3	system	rouge-4	system	rouge-SU4	system	rouge-W-1.2
A	0.6690	A	0.4725	A	0.4169	A	0.3951	A	0.4938	A	0.2205
C	0.6519	C	0.4578	C	0.4050	C	0.3851	C	0.4812	C	0.2103
B	0.6457	B	0.4388	B	0.3797	B	0.3589	B	0.4636	B	0.2102
ID10	0.5269	ID10	0.2560	ID10	0.1743	ID10	0.1349	ID10	0.2754	ID10	0.1458
ID2	0.4641	ID2	0.1715	ID3	0.0932	ID3	0.0639	ID2	0.2024	ID2	0.1283
ID4	0.4436	ID3	0.1655	$OBJ_1^{POS.EQ}$	0.0868	$OBJ_1^{POS.EQ}$	0.0636	ID3	0.1941	ID4	0.1227
ID3	0.4266	ID4	0.1507	ID2	0.0849	ID2	0.0551	ID4	0.1900	ID3	0.1199
$OBJ_1^{POS.EQ}$	0.4166	$OBJ_1^{POS.EQ}$	0.1463	ID4	0.0784	$OBJ_1^{POS.B}$	0.0546	$OBJ_1^{POS.EQ}$	0.1819	$OBJ_1^{POS.EQ}$	0.1140
OBJ_2^{MTC}	0.4143	OBJ_2^{MTC}	0.1426	OBJ_2^{MTC}	0.0776	OBJ_2^{MTC}	0.0545	OBJ_2^{MTC}	0.1777	OBJ_2^{MTC}	0.1136
ID5	0.4068	ID5	0.1343	$OBJ_1^{POS.B}$	0.0744	ID4	0.0518	ID5	0.1720	ID1	0.1113
ID1	0.4029	$OBJ_1^{POS.B}$	0.1293	OBJ_1^{TFISF}	0.0660	OBJ_1^{MTF}	0.0437	$OBJ_1^{POS.B}$	0.1652	ID5	0.1111
OBJ_1^{TFISF}	0.3959	OBJ_1^{TFISF}	0.1266	ID5	0.0655	OBJ_1^{TF}	0.0437	OBJ_1^{TFISF}	0.1624	ID8	0.1098
$OBJ_1^{POS.B}$	0.3932	OBJ_4	0.1238	OBJ_3	0.0649	OBJ_1^{TFISF}	0.0436	OBJ_3	0.1602	$OBJ_1^{POS.B}$	0.1089
ID7	0.3911	OBJ_3	0.1236	OBJ_4	0.0649	OBJ_3	0.0435	OBJ_4	0.1600	OBJ_3	0.1079
OBJ_3	0.3907	ID8	0.1230	OBJ_3^{MTF}	0.0646	OBJ_4	0.0435	OBJ_3^{MTF}	0.1589	OBJ_1^{TFISF}	0.1075
OBJ_4	0.3894	OBJ_2^{MTF}	0.1222	OBJ_1^{TF}	0.0646	$OBJ_1^{POS.F}$	0.0433	OBJ_1^{TF}	0.1589	OBJ_4	0.1072
OBJ_2^{MTF}	0.3874	OBJ_1^{TF}	0.1222	$OBJ_1^{POS.F}$	0.0641	ID5	0.0429	ID1	0.1588	OBJ_2^{MTF}	0.1067
OBJ_1^{TF}	0.3874	$OBJ_1^{POS.F}$	0.1216	ID8	0.0613	ID8	0.0409	$OBJ_1^{POS.F}$	0.1587	$OBJ_1^{POS.F}$	0.1063
$OBJ_1^{POS.F}$	0.3872	ID1	0.1191	ID1	0.0552	ID1	0.0367	ID8	0.1579	ID7	0.1038
ID8	0.3860	ID9	0.1053	ID6	0.0489	ID6	0.0331	ID9	0.1444	OBJ_1^{TF}	0.1000
ID9	0.3737	ID6	0.1043	ID9	0.0455	ID9	0.0277	ID6	0.1434	ID9	0.0983
ID6	0.3543	ID7	0.0923	ID7	0.0301	ID7	0.0165	ID7	0.1423	ID6	0.0948

Table 2: Evaluation results. MultiLing 2011. English.

- 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.
- Khachiyan, L. G. and Todd, M. J. (1993). On the complexity of approximating the maximal inscribed ellipsoid for a polytope. *Mathematical Programming*, 61:137–159.
- Khuller, S., Moss, A., and Naor, J. S. (1999). The budgeted maximum coverage problem. *Information Processing Letters*, 70(1):39–45.
- Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004)*, pages 25–26.
- Makino, T., Takamura, H., and Okumura, M. (2011). Balanced coverage of aspects for text summarization. In *TAC '11: Proceedings of Text Analysis Conference*.
- 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.