Language-independent Techniques for Automated Text Summarization

Mark LAST [a,1] and Marina LITVAK [a]
[a] *Ben-Gurion University of the Negev, Beer-Sheva, Israel*

Abstract. Text summarization is *the process of distilling the most important information from source/sources to produce an abridged version for a particular user/users and task/tasks*. Automatically generated summaries can significantly reduce the information overload on intelligence analysts in their daily work. Moreover, automated text summarization can be utilized for automated classification and filtering of text documents, information search over the Internet, content recommendation systems, online social networks, etc.

The increasing trend of cross-border globalization accompanied by the growing multi-linguality of the Internet requires text summarization techniques to work equally well on multiple languages. However, only some of the automated summarization methods proposed in the literature can be defined as “multi-lingual" or “language-independent," as they are not based on any morphological analysis of the summarized text.

In this chapter, we present a novel approach called MUSE (MUltilingual Sentence Extractor) to “language-independent" extractive summarization, which represents the summary as a collection of the most informative fragments of the summarized document without any language-specific text analysis. We use a Genetic Algorithm to find the best linear combination of 31 sentence scoring metrics based on vector and graph representations of text documents. Our summarization methodology is evaluated on two monolingual corpora of English and Hebrew documents, and, in addition, on a bilingual collection of English and Hebrew documents. The results are compared to 15 statistical sentence scoring methods for extractive single-document summarization found in the literature and to several state-of-the-art summarization tools. These bilingual experiments show that the MUSE methodology significantly outperforms the existing approaches and tools in both languages.

Keywords. extractive summarization, sentence extraction, language-independent summarization, genetic algorithm, optimization

1. Introduction

Document summarization is aimed at all types of electronic documents including Internet text files with the purpose of generating a summary - main document information expressed in “a few words." Automatically generated summaries can significantly reduce the information overload on professionals in various fields [1]. They may also be useful for automated classification and filtering of documents, information search

[1]Corresponding Author: Prof. Mark Last, Department of Information Systems Engineering, Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel; E-mail: mlast@bgu.ac.il

over the Internet, content recommendation systems, online social networks and other applications. Automated text summarization is known as a complex and challenging area. All approaches for automated summarization may be categorized according to a genre of generated summaries into *extractive* vs. *abstractive*, *single-document* vs. *multi-document*, *query-oriented* vs. *generic*, *language-dependent* vs. *multilingual*, etc. [2]. Extractive summarization, which is the focus of this chapter, is aimed at selecting a subset of the most relevant fragments from a source text into the summary. These fragments may be paragraphs [3], sentences [4], keyphrases [5,6,7] or keywords [8].

The methodologies used by researchers in the summarization area include *statistic*-based ones, using various text representations such as the classical vector space model or agraph representation, and *semantic*-based ones, using ontologies and language-specific knowledge. Although language-specific knowledge is expected to improve the quality of automated summaries in that language, it restricts the usage of the summarizer in multiple languages. Only systems that perform equally well on different languages without any language-specific knowledge can be considered as language-independent summarizers. *Statistic*-based methods not depending on the morphological analysis of a text may be applied to any language, and therefore, can be considered language-independent.

Today, when more and more languages are being used to publish information on the Internet[2], we need more multilingual tools that would allow the intelligence experts to analyze the potentially illegal content, which may be hidden under the shelter of some rare languages. Application of a complex *syntactical* and *morphological analysis* to the source document in a multilingual domain is computationally expensive, language-dependent, and requires a tremendous amount of computer resources [3]. Therefore, development of multilingual summarization approaches is becoming more essential. Particularly, there is a need of language-independent statistical techniques that can be readily applied to the text in any language without using language-specific morphological tools. In the absence of such tools, the most common alternative to language-independent summarization would be the manual translation of the entire document into one of the more common languages, which is a labor-intensive process, especially in case of dynamically changing content. On the other hand, the performance of existing automatic translation tools is still quite limited for semantically rich and complex languages, mainly due to the word-sense disambiguation problem [10].

However, only some of the automated summarization methods proposed in the literature can be defined as “cross-lingual" or “language-independent," as they are not based on any morphological analysis of the summarized text. The best-known examples of language-independent sentence scoring methods are:

- *Position*. There are several *position*-based methods for sentence scoring, where each sentence gets a score that is proportional to its relative position in the text. For example, *lead* method or *position-first* scores sentences proportionally to their closeness to the beginning of a document text.
- *Frequency*. In the *frequency*-based methods, the sentence score depends on the frequency weights of its words. For example, *coverage* method scores sentences by a fraction of keywords (frequent words) contained in this sentence.

[2] Gulli and Signorini [9] used Web searches in 75 different languages to estimate the Web size as of the end of January 2005.

[3] For example, running the Stanford Part-of-Speech (POS) tagger on the document collection from DUC 2002 (533 news documents) on a standard PC (with 2.67 GHz CPU and 2.00 GB of RAM) takes a few hours.

- *Title*. In the *title*-based methods, the sentence score is proportional to its similarity to the document title. Different similarity functions can be applied.
- *Length*. In this category of methods the sentence is scored by its length. The length can be measured by the number of words, characters, n-grams, etc.

Sections 2.2 and 3.3 describe the existing metrics and the metrics introduced in our research, respectively.

It has been empirically shown that combinations of several scoring metrics outperforms individual metrics. For example, the *position*-based method, which was declared to be the best individual method in [11,12], increased performance by 10% when combined with the cue-based method in [13]; the *title*-based method increased performance by 8% when combined with the title- and cue-based methods and by 3% when combined with cue-, location-, position-, and word-frequency-based methods in [11] and [13] respectively; the *cue*-based method was declared as the best individual method by the study [13] and increased performance by 7% when combined with the title and position methods and by 9% when combined with the position method in [11] and [12] respectively. All previous works [11,12,13,14,15] combined a relatively small amount of sentence features ($4 - 5$ on average, except the paper [16], which combined about 7 statistic metrics with several linguistic ones), and, when a linear combination was considered [11,16,14,15], no attempts to find the optimal weights were made. Since linguistic methods prevent a summarization system from being applied to multiple languages, we do not consider them in our research.

In this chapter, we introduce MUSE (MUltilingual Sentence Extractor) – a new approach to multilingual single-document extractive summarization, considering summarization as an optimization or a search problem. We use a Genetic Algorithm (GA) to find an optimal weighted linear combination of various (31 in total) sentence scoring methods, which are all language-independent and based on a vector or a graph representation of a document. We have evaluated our approach on two monolingual corpora of English and Hebrew documents, and, additionally, on one bilingual corpus, containing a mix of English and Hebrew documents. The major goals of our evaluation experiments can be summarized as follows:

- To compare the GA-based approach for single-document extractive summarization (MUSE) to the best known sentence scoring methods.
- To determine whether the same weighting model is applicable across two different languages.

We used the ROUGE evaluation toolkit (see Subsection 2.3) for quality evaluation in both experiments.

This chapter is organized as follows: the next section describes related work in extractive summarization, focusing mainly on multilingual sentence extraction; sections 3 and 4 describe the proposed language-independent sentence extraction approaches along with their experimental results. The last section presents conclusions and future work.

2. Related Work

2.1. Language-Independent Summarization

Multilingual or language-independent summarization is an ambiguous term used by scientists in different contexts. We will use this term here to denote a summarization process performed on a multilingual document collection (collection of documents written in several languages) by a single summarization tool. One important requirement for any multilingual/language-independent summarizer is that it demonstrates an equally good performance on different languages without any special adaptations, such as modifications of the algorithm and/or requirements for additional data, in each language.

To date, some methods have been elaborated and used to cross the barrier between different languages in information retrieval:

- In case of bilingual retrieval, index the documents by two languages [17];
- Interlingual mapping: monolingual technique inducing latent semantic axes/variables can be straightforwardly applied:
  - Latent semantic indexing and Singular value decomposition [18,19,20],
  - Generalized vector space model [21],
  - Probabilistic latent semantic analysis [22];
- Bilingual mapping of semantic spaces:
  - Monolingual LSI, bilingual mapping [23],
  - Canonical correlation analysis [24];
- Translation of document representation [25];
- Word level Alignment (probabilistic translation model) [26];
- Machine (aided) translation [27] – the “Holy Grail" of computational linguistics.

Most works dealing with multilingual summarization of documents/queries written in a foreign language use automatic (and/or manual) translation as a preprocessing or/and a post-processing step [27,25,28,29]. In these works, the quality of translation may impact summarization accuracy a lot. Since automatic translation is known as a very complex and challenging area, and existing tools usually suffer from the low accuracy of results, summarization systems face a noisy output from these tools.

Only few works do not employ translation or any other complementary tools in order to deal with multilingual content. The authors of [3,30,31] introduce a simple technique using a graph representation of the text and a similarity measure between text units that can be easily applied to multiple languages. Section 2.2 lists all sentence scoring metrics found in the literature that can be applied to multiple languages.

2.2. Language-Independent Sentence Scoring Metrics

Most of language-independent sentence scoring methods introduced in the literature can be categorized as *frequency*, *position*, *length* or *title*-based. *Frequency* and *title* are calculated using the vector representation of a text, whereas *position* and *length*-based methods calculate scores using the overall structure of the document.

We list and briefly describe below 15 existing methods for multilingual sentence scoring. The methods are divided into three main categories: *structure*, *vector* and *graph*-

based methods, according to the text representation model they use, where each category has an internal taxonomy as well. For example, *structure*-based methods include *position*-based methods, where each sentence receives a score proportional to its relative position in the text. The description of each approach includes a reference to the original work where this method was proposed for extractive summarization. We denote sentence by S, the total number of words in S by N, and text document by D. Some method descriptions contain the term *"keywords"*, referring to words used in the score calculations. Usually frequent but non-common words are considered as keywords.

- **Structure-based methods**:
  - *Position* [32]:
    - **POS_L**: the sentence score is proportional to its closeness to the end of the document: $SCORE(S_i) = i$, where i is the sequential number of the sentence in the document;
    - **POS_F**: the sentence score is proportional to its closeness to the beginning of the document: $SCORES(S_i) = \frac{1}{i}$;
    - **POS_B**: the sentence score is proportional to its closeness to the borders of the document: $SCORE(S_i) = max(\frac{1}{i}, \frac{1}{n-i+1})$, where n is the total number of sentences.
  - *Length* [33]:
    - **LEN_W**: the sentence score is equal to the number of *words* in the sentence;
    - **LEN_CH**: the sentence score is equal to the number of *characters* in the sentence.
- **Vector-based methods**:
  - *Frequency*-based:
    - **LUHN** [4]: the significance factor of a sentence in the Luhn method is derived from an analysis of its words and based on a combination of two measurements: the frequency of word occurrence, and the relative position of keywords within the sentence. The score reflects the number of occurrences of keywords within a sentence and the linear distance between them due to the intervention of non-significant words:

$$SCORE(S) = max_{i \in \{clusters(S)\}}\{SCORE_i\}, \quad (1)$$

      where $clusters(S)$ are portions of the sentence S bracketed by keywords[4]. $SCORE_i = \frac{S_i^2}{N_i}$, where S_i is the number of keywords in the i^{th} cluster, where N_i is the total number of words in the i^{th} cluster.
    - **KEY** [11]: the Key method compiles a Key glossary for each document, ideally consisting of topic words statistically selected from the body of that document – keywords. The sentence score is calculated as a sum of its keyword frequencies:

$$SCORE(S) = \sum_{i \in \{Keywords(S)\}} tf_i, \quad (2)$$

[4]Luhn's experiments suggest an optimal limit of 4 or 5 non-significant words between keywords.

where tf_i stands for the term frequency of the i^{th} keyword in S.

* **COV** [34]: in the Coverage method, the sentence score is calculated as a ratio of keywords number:

$$SCORE(S) = \frac{|Keywords(S)|}{|Keywords(D)|}. \quad (3)$$

* **TF** [35]: this method differs from the Key method by calculating the average term frequency for *all* the words in the sentence. The sentence score is calculated as an average term frequency for all sentence words:

$$SCORE(S) = \sum_{i \in \{words(S)\}} \frac{tf_i}{N}. \quad (4)$$

* **TFISF**, Term Frequency Inverse Sentence Frequency [36] represents a product of term frequency and inverted sentence frequency. The sentence score is calculated as an average TFISF for all the words in the sentence:

$$SCORE(S) = \sum_{t \in S} f(t) \times isf(t), \quad (5)$$

where $f(t)$ stands for term t frequency and $isf(t) = 1 - \frac{log(n(t))}{log(n)}$, where n is the number of sentences in the document and $n(t)$ is the number of sentences containing t.

* **SVD**, Singular Value Decomposition [20] is a central component of LSA (Latent Semantic Analysis) that is a fully automatic algebraic-statistical technique for extracting and representing the contextual usage of word meanings in passages of discourse. According to [19], one should create a term by sentences $m \times n$ matrix $A = [A_1, A_2, \ldots, A_n]$, where each column A_i represents the weighted term-frequency vector of the i^{th} sentence in the document, and apply SVD to the matrix A: $A = U\Sigma V^T$. The summarization method proposed by [19] chooses the summary sentences on the basis of the relative importance of the 'topics' they mention, described by the matrix V^T. The summarization algorithm simply chooses for each 'topic' the most important sentence for that topic: i.e., the k^{th} sentence chosen is the one with the largest index value in the k^{th} right singular vector in matrix V^T. The improved summarization method introduced by [20] selects the sentences whose vectorial representation in the matrix $\Sigma^2 \cdot V^T$ has the greatest 'length'. Intuitively, the idea is to choose the sentences with the greatest combined weight across all important topics. More formally, sentence score is calculated as a length of the sentence vector in $\Sigma^2 \cdot V^T$ after computing SVD.

* *Title* [11]: according to the Title method, every sentence gets a score that measures its similarity to the title of the document: $SCORE(S) = sim(S, T)$. The similarity function can be one of various functions:

  * **TITLE_O** using *Overlap* similarity:

$$sim(S, T) = \frac{|S \cap T|}{min\{|S|, |T|\}}, \quad (6)$$

where $|S|$ and $|T|$ are the number of words in the sentence S and the title T respectively, and $|S \cap T|$ is the number of common words in S and T;

* **TITLE_J** using *Jaccard* similarity:

$$sim(S,T) = \frac{|S \cap T|}{|S \cup T|}, \tag{7}$$

where $|S \cap T|$ is the number of common words in S and T, and $|S \cup T|$ is the total number of words in S and T;

* **TITLE_C** using *Cosine* similarity:

$$sim(\vec{S},\vec{T}) = cos(\vec{S},\vec{T}) = \frac{\vec{S} \times \vec{T}}{\left|\vec{S}\right| \times \left|\vec{T}\right|}, \tag{8}$$

where $\vec{S}$ and $\vec{T}$ are vector representations for S and T using the term frequency respectively.

- **Graph-based methods**:

  * **ML_TR** is a multilingual version of TextRank [31] without morphological analysis. Each document is represented as a graph of nodes that stand for sentences interconnected by similarity (*overlap*) relationship. The overlap of two sentences is determined simply as the number of common tokens between the two sentences, normalized by the length of these sentences. Formally, given two sentences S_i and S_j, where the sentence S_i is represented by the set of N_i words: $w_1, w_2, \cdots, w_{N_i}$, the similarity of S_i and S_j is defined as: $Similarity(S_i, S_j) = \frac{|\{w_k | w_k \in S_i \& w_k \in S_j\}|}{log(|S_i|)+log(|S_j|)}$. The sentence score is equal to the PageRank [37] score of its node in the representation graph:

$$WS(V_i) = (1-d) + d * \sum_{V_j \in In(V_i)} \frac{w_{ji}}{\sum_{V_k \in Out(V_j)} w_{jk}} WS(V_j), \tag{9}$$

where $In(V_i)$ is the set of vertices that point to V_i (predecessors), $Out(V_j)$ is the set of vertices that vertex V_j points to (successors), d is the damping factor, which integrates into the model the probability of jumping from a given vertex to another random vertex in the graph (we used $d = 0.85$, setting the probability of jumping to a completely new node at 0.15), and w_{ji} is the weight assigned to the edge connecting the two vertices: V_j and V_i and equal to the similarity value between the corresponding sentences.

2.3. Summary Evaluation with ROUGE

The problem of evaluating text summarization is a very complex one, and serious questions remain open concerning the appropriate methods and types of such evaluation. There are a variety of methods to compare summarization systems' performance. In general, methods for evaluating text summarization approaches can be broadly classified into two categories:

- *Extrinsic evaluation*, in which the quality of a summary is judged on the basis of how it affects the completion of some other task. The tasks have included humans

determining the relevance of documents to topics [38,39,40] as well as humans answering questions based on reading the summaries [41].

- *Intrinsic evaluation*, where humans judge the quality of the summarization directly, based on an analysis of the summary. This analysis can involve user judgments of fluency (language complexity, presence of dangling anaphors, preservation of structured environments, grammatical stylistic features, etc.) of the summary, coverage of key/essential ideas [38,39] or similarity to an "ideal" summary [11,12]. Many of fluency measures can be automatically detected. Judging coverage of key ideas, though, is more subjective, even if humans agree in advance as to what constitutes the set of key ideas. The problem with matching a system summary against an ideal summary is that the ideal summary is hard to establish. The human-produced summary may be supplied by the author or constructed by a judge. There can be a large number of different generic and user-focused abstracts that could summarize a given document.

In our experiments, we have used ROUGE (Recall-Oriented Understudy for Gisting Evaluation) toolkit for automatic intrinsic evaluation of induced summaries. It includes measures for automatically determining the quality of a summary by comparing it to other (gold standard) summaries created by humans. The measures count the number of overlapping units such as n-gram, word sequences, and word pairs between the computer-generated summary to be evaluated and the gold standard summaries created by humans. ROUGE summarization evaluation package provides several different ROUGE measures: ROUGE-N, ROUGE-L, ROUGE-W, and ROUGE-S. Paper [42] shows how these measures correlate with human judgments using DUC 2001, 2002, and 2003 data. We tested all metrics on our dataset and chose to use ROUGE-N with $N = 1$ (ROUGE-1) as a metric allowing the greatest variation between evaluated summarizers. Also, ROUGE-1 performed well in most of summarization tasks according to the results published in [42], and was found to have the highest correlation with human judgments, at a confidence level of 95%. Formally, ROUGE-N is an n-gram recall between a candidate summary and a set of reference summaries. ROUGE-N is computed as follows:

$$ROUGE - N = \frac{\Sigma_{S \in ReferenceSummaries} \Sigma_{gram_n \in S} Count_{match}(gram_n)}{\Sigma_{S \in ReferenceSummaries} \Sigma_{gram_n \in S} Count(gram_n)}, \quad (10)$$

where n stands for the length of the n-gram (sequence of n words), $gram_n$, and $Count_{match}(gram_n)$ is the maximum number of n-grams co-occurring in the candidate summary and the set of reference summaries.

The number of n-grams in the denominator of the ROUGE-N formula increases as we add more references. The numerator sums over all reference summaries. This effectively gives more weight to matching n-grams occurring in multiple references. Therefore, a candidate summary that contains words shared by more references is favored by the ROUGE-N measure. Intuitively, we normally prefer a candidate summary that is more similar to the consensus among reference summaries.

Currently, ROUGE package contains all metrics implemented as recall, precision and f-measure based metrics.

The best published result for ROUGE-1 (Recall) among DUC 2002 systems (including language-specific summarizers tailored to the English language) was 50% [43].

3. Sentence Extraction

3.1. Overview

In this chapter we deal with sentence extraction applied to document summarization. Section 3.2 describes the text representation models used by our methodology. In Section 3.3 we present 16 new and modified language-independent sentence scoring metrics that are based on vector- and graph-based representations of text documents. Then, in Section 3.4 we present MUSE (MUltilingual Sentence Extractor) – a new approach to multilingual single-document extractive summarization, considering summarization as an optimization or a search problem aimed at finding the best linear combination of sentence scoring features. We use a Genetic Algorithm (GA) to find an optimal linear combination of 31 sentence scoring methods, which are all language-independent and based on a vector or a graph representation of the document. In the linear combination we use 15 existing (described in Section 2.2 above) and 16 new (presented in Section 3.3) methods. The MUSE algorithm is evaluated on an English and a Hebrew corpus.

3.2. Text Representation Models

The vector-based scoring methods listed below use *tf* or *tf-idf* term weights for evaluation of sentence importance, whereas the graph-based methods use a slightly modified version of the *simple* word-based graph representation of text and web documents introduced in [44,45]. The *simple* graph representation is built from the words segmentation and holds unlabeled edges representing order-relationship between the words represented by nodes. The stemming and stopword removal operations can be performed before the graph construction[5]. Only a single vertex for each distinct word is created, even if it appears more than once in the text. Thus, each node label in the graph is unique. Each node has a rank. The rank type is a configurable parameter and can be set to out-degree, in-degree, degree, HITS or PageRank. By default, the node rank is set to degree, which was shown as the best measure according to experiments presented in [8]. Unlike the original *simple* representation, where only a specified number of most frequent terms are included in the graph, we do not impose any limit on the number of graph nodes.

Edges represent order-relationships between two terms (words): there is a directed edge from A to B if an A's term immediately precedes a B's term in any sentence of the document. We label each edge by the IDs of sentences that contain both words in the specified order. This definition of graph edges is slightly different from the co-occurrence relations used in [6] for building undirected document graphs with unlabeled edges, where the order of word occurrence is ignored and the size of the co-occurrence window varies between 2 and 10.

A small example of the graph representation used here is shown in Figure 1, which shows a sample text (enumerated sentences) and its graph representation respectively.

[5]This part may be skipped for language-independent processing unless appropriate stemmers and stopword lists are available for a given language

0	Hurricane Gilbert Heads Toward Dominican Coast.
1	Hurricane Gilbert swept toward the Dominican Republic Sunday, and the Civil Defense alerted its heavily populated south coast to prepare for high winds, heavy rains and high seas.
2	The storm was approaching from the southeast with sustained winds of 75 mph gusting to 92 mph.

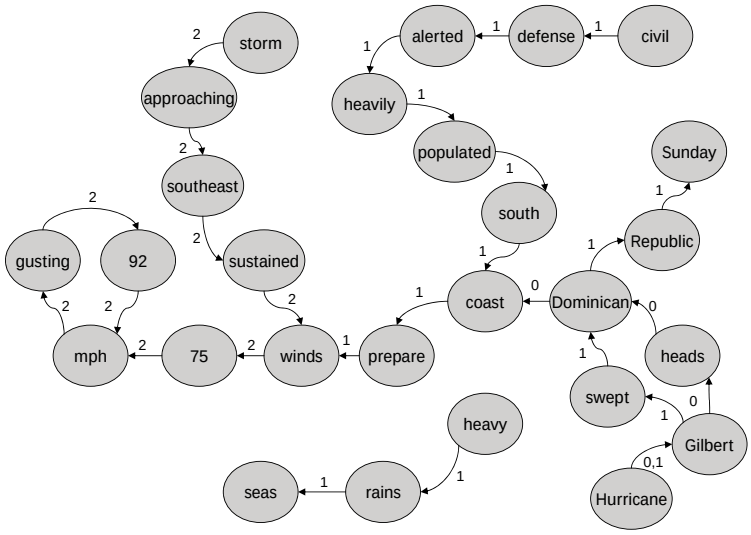

Figure 1. Text and its graph representation.

3.3. New Language-Independent Scoring Metrics for Sentence Extraction

In this section we describe 16 new and modified methods for sentence scoring, including 13 methods based on the word-based graph representation. The description of each modified approach includes a reference to the original method. The methods are divided into three main categories: *structure*, *vector* and *graph*-based methods, according to the text representation model, while each category has an internal taxonomy as well. For example, *structure* features contains *position*-based methods, where each sentence gets a score that is proportional to its relative position in the text.

- **Vector-based methods**:
  - *Document Coverage*. These methods score a sentence by its similarity to the rest of the document sentences ($D - S$). The intuition behind this approach is that the more document content is covered by a particular sentence, the more important the sentence must be for the summary. $SCORE(S) = sim(S, D - S)$. The similarity function may be one of the following:
    - **D_COV_O** using *Overlap* similarity:

$$\frac{|S \cap T|}{min\{|S|, |D - S|\}}; \tag{11}$$

* **D_COV_J** using *Jaccard* similarity:

$$\frac{|S \cap T|}{|S \cup D - S|}; \tag{12}$$

* **D_COV_C** using *Cosine* similarity:

$$cos(\vec{S}, \vec{D - S}) = \frac{\vec{S} \bullet \vec{D - S}}{\left|\vec{S}\right| \bullet \left|\vec{D - S}\right|}. \tag{13}$$

- **Graph-based methods**:
  * *Degree*-based:
    * **LUHN_DEG**. This metric is a graph-based extension of LUHN measure (see formula 1). The *Degree* value of the node standing for the word is used instead of the word frequency: words are considered as keywords if they are represented by nodes having a degree higher than the predefined threshold. The score reflects the number of occurrences of keywords within a sentence and the linear distance between them due to the intervention of non-significant words:

$$SCORE(S) = max_{i \in \{clusters(S)\}}\{SCORE_i\}, \tag{14}$$

      where $clusters(S)$ are portions of the sentence S bracketed by keywords. $SCORE_i = \frac{S_i^2}{N_i}$, where S_i is the number of keywords in the i^{th} cluster, where N_i is the total number of words/nodes in the i^{th} cluster.
    * **KEY_DEG**. This metric is a graph-based extension of KEY measure (see formula 2). As in LUHN_DEG, the *Degree* value of the node standing for the word is used instead of the word frequency, in order to determine keywords:

$$SCORE(S) = \sum_{i \in \{Keywords(S)\}} Deg_i, \tag{15}$$

      where Deg_i stands for the *Degree* of the i^{th} keyword in S.
    * **COV_DEG**. This metric is a graph-based extension of COV measure (see formula 3). The *Degree* value of the node standing for the word is used instead of the word frequency, in order to determine keywords:

$$SCORE(S) = \frac{|Keywords(S)|}{|Keywords(D)|}. \tag{16}$$

    * **DEG**. The sentence score is calculated as an average degree for all sentence nodes: $SCORE(S) = \frac{\sum_{i \in \{words(S)\}} Deg_i}{N}$.
    * **GRASE** (GRaph-based Automated Sentence Extractor) is a modification of the Salton's algorithm [3], using the graph representation defined in Section 3.2 above. In the word-based graph representation, all sentences are represented by paths, completely or partially. In order to identify the relevant sentences, we search for the *bushy* paths and extract the most frequent sentences from them. According to Salton et al. [3], the *bushiness* of a node

equals to its *Degree*, and a *bushy* path is constructed mostly out of *bushy* nodes. The path score can be calculated as a normalized sum of *Degree*[6] for all its nodes. The top-ranked paths are extracted. Figure 2 shows five most-ranked paths for the document graph depicted in Figure 1. In contrast to [3], we search among *all* paths in graph. For efficiency, the "longest increasing subsequences" algorithm from dynamic programming [46] was adopted. Each sentence in the *bushy* path gets a domination score that equals the number of edges with its label in this path, normalized by the sentence length. The relevance score for the sentence is calculated as a sum of its domination scores over all the extracted paths.

* *PageRank*-based:
  * **LUHN_PR**. This metric is a graph-based extension of LUHN measure (see formula 1). The *PageRank* score of the node representing the word is used instead of the word frequency: words are considered as keywords if they are represented by nodes with a PageRank score higher than a predefined threshold . The PageRank score is calculated by the formula:

$$PR(V_i) = (1-d) + d * \sum_{V_j \in In(V_i)} \frac{1}{|Out(V_j)|} PR(V_j), \tag{17}$$

    where $In(V_i)$ is the set of vertices that point to V_i (predecessors), $Out(V_j)$ is the set of vertices that vertex V_j points to (successors), and d is the damping factor, integrating into the model the probability of jumping from a given vertex to another random vertex in the graph (we used $d = 0.85$ which sets the probability of jumping to a completely new page at 0.15)
  * **KEY_PR**. This metric is a graph-based extension of KEY measure (see formula 2). The *PageRank* score of the node standing for the word is used instead of the word frequency, in order to identify keywords:

$$SCORE(S) = \sum_{i \in \{Keywords(S)\}} PR_i, \tag{18}$$

    where PR_i stands for the *PageRank* of the i^{th} keyword in S.
  * **COV_PR**. This metric is a graph-based extension of COV measure (see formula 3). The *PageRank* score of the node standing for the word is used instead of the word frequency, in order to identify keywords.
  * **PR**. The sentence score is calculated as an average PageRank score for all sentence nodes: $SCORE(S) = \frac{\sum_{i \in \{words(S)\}} PR_i}{N}$.
* *Similarity*-based. These metrics use similarity functions based on graph edge matching techniques similar to [47]. Edge matching is an alternative approach to measure the similarity between graphs based on the number of common edges. Since these are extensions of similarity vector-based methods, they use the same similarity functions that can be applied to graph matching techniques: *Overlap* (see formula 6) and *Jaccard* (see formula 7).

[6] Actually, any ranks for nodes can be used.

* **TITLE_E_O**. This metric is a graph-based extension of TITLE_O, using the *Overlap* similarity function (see formula 6) and edge matching between the title and the sentence graphs:

$$sim(S,T) = \frac{|S \cap T|}{min\{|S|, |T|\}}, \tag{19}$$

where $|S|$ and $|T|$ are the number of edges in the document graph labeled by S and ones labeled by T respectively, and $|S \cap T|$ is the number of common edges (labeled by both IDs);

* **TITLE_E_J**. This metric is a graph-based extension of TITLE_J, using the *Jaccard* similarity function (see formula 7) and edge matching between the title and the sentence graphs:

$$sim(S,T) = \frac{|S \cap T|}{|S \cup T|}, \tag{20}$$

where $|S \cap T|$ is the number of common edges in S and T, and $|S \cup T|$ is the total number of edges labeled by either S or T;

* **D_COV_E_O**. This metric is a graph-based extension of D_COV_O, using the *Overlap* similarity function (see formula 11) and edge matching between the sentence and the document complement (the rest of the document sentences) graphs.
* **D_COV_E_J**. This metric is a graph-based extension of D_COV_J, using the *Jaccard* similarity function (see formula 12) and edge matching between the sentence and the document complement graphs.

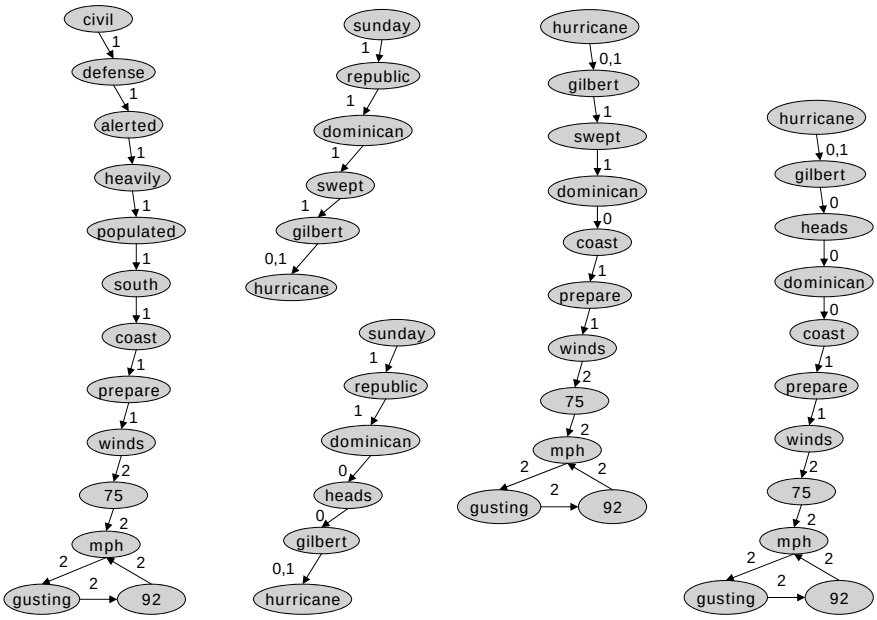

Figure 2. Five top-ranked paths for text from Figure 1.

A complete taxonomy of sentence scoring methods described above and in Section 2.2 (31 in total), is presented in Figure 3.

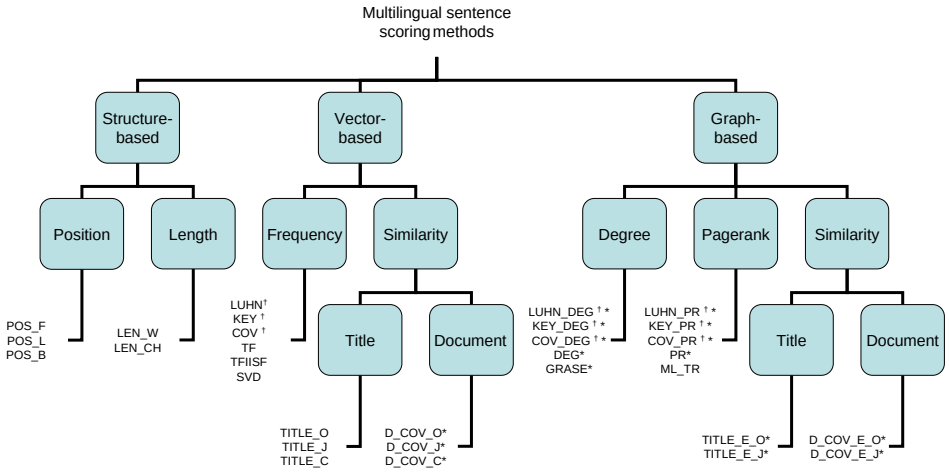

Figure 3. Taxonomy of multilingual sentence scoring methods.

1	Statistics on Six Decades of Oscar With PM-Oscar Nominations Bjt
2	Statistics on Six Decades of Oscar With PM-Oscar Nominations Bjt The motion picture industry's most coveted award, Oscar, was created 60 years ago and 1,816 of the statuettes have been produced so far.
3	Weighing 8 pounds and standing 13 inches tall, Oscar was created by Metro-Goldwyn-Mayer studios art director Cedric Gibbons, who went on to win 11 of the trophies.
4	Oscar, manufactured by the R.S. Owens Co., Chicago, is made of Britannia metal, copper plate, nickel plate and gold plate.
5	From 1942 to 1944, the trophy was made of plaster, but winners were later presented with the real thing.
6	According to the Academy of Motion Pictures Arts and Sciences, the only engraving mistake was in 1938 when the best actor trophy given to Spencer Tracy for Boy's Town" read: Best Actor: Dick Tracy."
7	The Academy holds all the rights on the statue and reserves the right to buy back an Oscar before someone takes it to a pawn shop," said Academy spokesman Bob Werden.
8	The most-nominated film was All About Eve" in 1950.
9	It got 14 nominations.
10	Ben-Hur" in 1959 was the most-awarded film with 11, and Walt Disney was the most-awarded person with 32.

Figure 4. Sample AP880217-0100 document from DUC 2002 collection.

Table 1 contains scores for AP880217-0100 document sentences from Figure 4 for each of the 31 single metrics (not normalized). All new and modified methods introduced above are denoted by asterisk (*). All score values are rounded to the numbers with three decimal places. We used a sentence splitter provided with the MEAD summarizer [14], producing DOCSENT-formatted documents for plain text or HTML source documents. As a result of splitting, the title of the document appears twice in its docsent file: as the first sentence and as a subsentence of the second one.

Table 1. Sentence scores.

Score	1	2	3	4	5	6	7	8	9	10
LUHN	1.333	1.333	0.000	1.800	0.000	0.000	0.000	0.000	0.000	0.000
KEY	0.163	0.279	0.202	0.183	0.067	0.221	0.212	0.038	0.029	0.077
TF	0.027	0.017	0.013	0.015	0.011	0.014	0.015	0.013	0.029	0.013
TFISF	1.494	1.235	1.080	1.234	1.066	1.306	1.207	1.133	1.569	1.233
COV	0.075	0.200	0.188	0.150	0.075	0.200	0.175	0.038	0.013	0.075
POS_F	1.000	0.500	0.333	0.250	0.200	0.167	0.143	0.125	0.111	0.100
POS_L	1	2	3	4	5	6	7	8	9	10
POS_B	1.000	0.500	0.333	0.250	0.200	0.200	0.250	0.333	0.500	1.000
TITLE_C	1.000	0.890	0.633	0.622	0.000	0.000	0.595	0.000	0.384	0.000
TITLE_O	1.000	0.375	0.067	0.083	0.000	0.000	0.071	0.000	1.000	0.000
TITLE_J	1.000	0.375	0.050	0.059	0.000	0.000	0.053	0.000	0.167	0.000
LEN_W	10	33	27	20	19	34	31	9	4	18
LEN_CH	64	201	163	122	104	200	167	52	22	105
D_COV_O*	1.000	0.500	0.133	0.083	0.167	0.188	0.143	0.333	1.000	0.167
D_COV_C*	0.585	0.603	0.428	0.385	0.101	0.218	0.455	0.123	0.225	0.089
D_COV_J*	0.075	0.100	0.025	0.013	0.013	0.038	0.025	0.013	0.013	0.013
SVD	0.181	0.121	0.074	0.035	0.069	0.000	0.084	0.155	0.229	0.000
LUHN_DEG*	0.000	0.000	0.000	1.800	0.000	2.000	1.333	0.000	0.000	2.000
KEY_DEG*	8.000	15.000	11.000	13.000	3.000	20.000	16.000	4.000	0.000	8.000
DEG*	2.833	2.438	2.333	2.750	2.000	2.500	2.643	2.000	2.000	2.333
COV_DEG*	0.063	0.175	0.163	0.150	0.063	0.200	0.163	0.013	0.013	0.050
LUHN_PR*	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
KEY_PR*	0.014	0.014	0.014	0.028	0.000	0.026	0.027	0.000	0.000	0.013
PR*	0.013	0.013	0.013	0.013	0.012	0.013	0.013	0.012	0.013	0.012
COV_PR*	0.013	0.013	0.013	0.025	0.000	0.025	0.025	0.000	0.000	0.013
TITLE_E_O*	1.000	0.313	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
TITLE_E_J*	1.000	0.313	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
D_COV_E_O*	1.000	0.375	0.071	0.000	0.000	0.000	0.000	0.000	0.000	0.000
D_COV_E_J*	0.057	0.069	0.011	0.000	0.000	0.000	0.000	0.000	0.000	0.000
GRASE*	0.100	0.030	0.037	0.050	0.053	0.029	0.032	0.111	0.000	0.056
ML_TR	0.088	0.153	0.142	0.105	0.111	0.120	0.105	0.054	0.022	0.101

3.4. Improving Multilingual Summarization using a Combination of Sentence Scoring Metrics

Every single metric for sentence scoring (listed in sections 2.2 and 3.3) indicates only one specific characteristic of the sentence (i.e. similar to the title, close to the beginning, containing a big fraction of keywords, etc.), which usually does not perform equally well on various documents of different lengths, genres, etc. The question that the researcher should ask when she/he tries to find a “universal" solution is which sentence extraction method to use in a general case? Important sentences usually combine several characteristics (i.e. similar to the title AND close to the beginning AND containing a big fraction of keywords AND ...). Assuming the independence of different extraction methods, a combination of their evidence by merging individual sentence scores might outperform all individual metrics on a large collection of documents in several languages. Previous works [11,14,15] used a linear combination of sentence features as a relevance sentence score, considered a small amount of features ($4-5$), and made no attempt to find the best

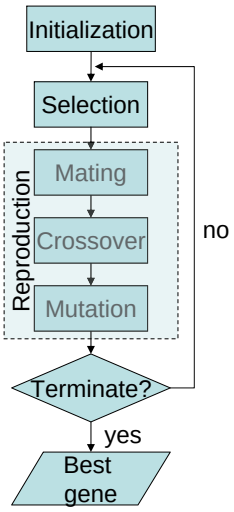

Figure 5. Simplified flowchart of a Genetic Algorithm.

weights for a combination. The search for the best linear combination of features can be performed by different optimization algorithms, such as Hill climbing, Evolutionary programming, etc. Genetic Algorithms (GA) are known as a prominent search and optimization technique [48]. In [49], we have introduced MUSE (MUltilingual Sentence Extractor), a learning approach to language-independent extractive summarization, where the best set of weights for a linear combination of sentence scoring metrics is found by a genetic algorithm trained on a collection of document summaries. The obtained weighting vector can be used for sentence scoring in future summarization. Since most sentence scoring methods have a linear computational complexity, only the training phase of our approach is time-consuming.

GAs are categorized as global search heuristics motivated by the principles of biological evolution. Figure 5 demonstrates a simplified flowchart of a GA. A typical genetic algorithm requires: (1) a genetic representation of the solution domain, (2) a fitness function to evaluate the solution domain, (3) specifications of the basic operations for reproduction: crossover and mutation.

In the MUSE algorithm, we represent the solution as a vector of weights for a linear combination of sentence scoring methods – real-valued numbers in an unlimited range normalized in such a way that they sum up to 1. The vector size is fixed and it equals to the number of methods used in the combination.

The fitness function is defined over the genetic representation and it measures the quality of the represented solution. We use ROUGE-1 Recall (described in Subsection 2.3 above) as a fitness function measuring the summarization quality which is being maximized during the optimization procedure.

Below we describe each phase of the optimization procedure in detail.

- **Initialization**. We start from N randomly generated genes/solutions as an initial population. Each gene is represented by a weighting vector $v_i = w_1, \ldots, w_D$ having a fixed number of D elements. All elements are generated from a standard normal distribution with $\mu = 0, \sigma^2 = 1$ and normalized to sum up to 1. A neg-

ative weight in such solution representation, if it occurs, can be considered as a "penalty" for the associated metric.

- **Selection**. During each successive generation, a proportion of the existing population is selected to breed a new generation. We use a truncation selection method that rates the fitness of each solution and selects the N_1 best individual solutions – getting the maximal ROUGE value.
- **Reproduction**. In this step we generate the next generation population of solutions from those selected through genetic operators: *mating*, *crossover*, and *mutation*.
  In *mating*, a pair of "parent" solutions is randomly selected. A new solution is created using *crossover* and *mutation*. New parents are selected for each new child, and the process continues until a new population of solutions of the appropriate size of N is generated.
  Our *crossover* operator includes a probability P_χ that a new different offspring solution will be generated by calculating the weighted average of two "parent" vectors, according to [50]. Formally, a new vector v will be created from two vectors: v_1 and v_2 according to the formula $v = \lambda * v_1 + (1 - \lambda) * v_2$. With the probability of $1 - P_\chi$, the offspring will be a duplicate of one of the parents.
  The purpose of *mutation* in GAs is preserving and introducing diversity. Our mutation operator involves a probability P_μ that an arbitrary weight w_i in a vector will be changed by uniformly randomized factor φ in some specified range τ from its original value. In other words, at probability P_μ, w_i for some arbitrary i will be updated according to the formula: $w_i = \varphi * w_i$, where $\varphi \in \tau$.
- **Termination**. The generational process is repeated until a termination condition, a plateau of solution/combination fitness, has been reached, such that successive iterations no longer produce better results. In our experiments, we stopped when the improvement in the best solution no longer exceeded the minimal improvement ϵ.

We used the following MUSE settings in our experiments:

- population size $N = 500$,
- dimension (number of elements) of a solution vector (gene) $D \leq 31$ (we experimented with a subset of 31 scoring metrics, filtering highly correlated ones – see Section 4),
- size of selection $N_1 = 100$ (fifth out of $N = 500$),
- crossover probability $P_\chi = 0.8$,
- $\lambda = 0.5$ for a new gene calculation,
- mutation probability $P_\mu = 0.03$,
- range of randomized factor used in mutation $\tau = [-0.3, 0.3]$,
- minimal improvement $\epsilon = 1.0E - 21$

4. Experiments

4.1. Overview

In order to evaluate MUSE, the new summarization approach proposed in Section 3.4, a comparative experiment was conducted on two monolingual corpora of English and

Hebrew texts, and also on a bilingual corpus of texts on both languages. These two languages belong to different language families (Indo-European and Semitic languages, respectively), that increased the generality of our evaluation. The experimental procedure for MUSE had the following goals:

- To evaluate the optimal sentence scoring model produced from the corpora of summarized documents in two languages.
- To compare the performance of the GA-based multilingual summarization method proposed in this work to the best state-of-the-art approaches.
- To compare the method performance on each language.
- To determine whether the same sentence scoring model can be effectively applied to both languages for extractive summarization.

4.2. Text Preprocessing

In extractive summarization, sentence segmentation plays a crucial role, while improper segmentation may impact the quality of summarization results. We used a sentence splitter provided with the MEAD summarizer [14] for English and a simple splitter for Hebrew which splits the text by period, exclamation or question punctuation marks. Obviously, the same set of splitting rules may be used for many different languages. The reason for using separate splitters for English and Hebrew was the implementation restriction of the MEAD splitter to European languages.

4.3. Experimental Data

For English texts, we performed experiments on the corpus of document abstracts provided for the single document summarization task at the Document Understanding Conference 2002 [43]. We chose the corpus from DUC 2002 as it had a similar task – single-document summarization with summaries up to 100 words. This benchmark dataset contains 533 unique text documents on 567 different topics[7]. The document sets have been produced using data from the Wall Street Journal 1987-1992, AP newswire 1989-1990, Financial Times 1991-1994, etc. Each document is at least 10 sentences long and has two (in rare cases three) human-written abstracts of approximately 100 words on average. Despite the fact that evaluation against gold standard extracts rather than abstracts is more appropriate in our case, we have not succeeded in obtaining such data in English.

For Hebrew, we have prepared an annotated collection of *Haaretz* articles and their extractive summaries. We set up an experiment, where 50 news articles, of 250 to 830 words, from the *Haaretz*[8] newspaper site were summarized by human assessors. We provided assessors with the Tool Assisting Human Assessors (TAHA) software,[9] enabling easy selection and storage of sentences to be included in the document extract. In total, 70 undergraduate students from the Department of Information Systems Engineering participated in the experiment with ten different documents randomly assigned to every student. The experiment participants were instructed:
(1) to spend at least five minutes on each document;

[7] Some documents are repeated under different topics.
[8] http://www.haaretz.co.il
[9] TAHA can be provided upon request

Table 2. Length statistics for generated summaries

System	min	max	avg	St.Dev.
TextRank	85	139	99.99	9.97
SUMMA	18	293	100.95	16.35
MEAD	18	293	103.46	20.16
MS	1	122	80.11	29.55

(2) to ignore dialogs and citations;
(3) to read the whole document before starting sentence extraction;
(4) to ignore redundant, repeated and too detailed information;
(5) to follow the constraints of the minimal and maximal size for a summary, namely 95 and 100 words, respectively.

A quality assessment was performed by comparing the summary of every student to the summaries of all other students using the ROUGE evaluation toolkit adapted for Hebrew[10] and ROUGE-1 Recall metric [51]. We filtered out all the summaries produced by assessors that received an average ROUGE score below 0.5, i. e. agreed with the rest of assessors in less than 50% of cases. Finally, we were left with the summaries made by about 60% of the most consistent assessors, with ten extracts on average per single document[11]. The minimal and the maximal ROUGE scores of the most consistent assessors were 50% and 57% respectively.

The third, bilingual corpus has been assembled from the documents of both monolingual corpora.

4.4. State-of-the-Art Summarizers: Comparative Results

We evaluated and compared four state-of-the-art summarization systems: SUMMA [15], MEAD [14], AutoSummarize tool in Microsoft ®Office Word 2003 [52], and TextRank [31]. We evaluated all systems using the Basic Elements method introduced by Hovy et al. [53,54]. An evaluation has been performed using only an English corpus – the Document Understanding Conference, [43] dataset from 2002. Since the manual abstracts are approximately 100 words long, we generated all summaries under the constraint of 100 or less words. In each system the default settings were used. Using the AutoSummarize tool of MS-Word we encountered an apparent bug – the summaries generated by using MS-Word AutoSummarize dialog box differ from those generated by running the Document.AutoSummarize Method from VBA with the same parameter settings. Since we needed to generate summaries for quite a large number of documents, we used the results of the Document.AutoSummarize Method implemented in a VBA script. Table 2 presents statistical information about the length of the summaries generated for each system. Table 3 demonstrates the average BE scores on DUC 2002 data for each human summarizer (A-J), SUMMA, MEAD, MS, and TextRank. The BE platform was ran with default settings. Figure 6 demonstrates the summarization quality (BE results) as a function of document length for all evaluated systems and four baselines: LOCATION FIRST, LOCATION LAST, LOCATION MIDDLE and RANDOM SELECTION, where

[10]The regular expressions specifying “word" were adapted to Hebrew alphabet. We are grateful to Dr. Michael Elhadad and his student, Galina Volk, for providing us the adapted toolkit. The same toolkit was used for summaries evaluation on the Hebrew corpus.

[11]The dataset is available at http://www.cs.bgu.ac.il/ litvakm/research/

Table 3. Average BE scores on DUC 2002

Summarizer	BE score
G	0.2943
H	0.2677
D	0.2641
J	0.2548
TextRank	0.2466
B	0.2433
I	0.2366
MEAD	0.2337
F	0.2312
SUMMA	0.2310
A	0.2181
C	0.2095
E	0.2071
MS	0.1464

first, last, middle and random sentences respectively were extracted. All documents from DUC 2002 collection were divided into 10 groups, where the minimal length threshold was set to $100 * n, n \in [1, 10]$. It can be seen that the summarizers' relative performance mostly remains unchanged for all length thresholds. Also, we can see that most systems (except one – TextRank) perform worse than POSITION FIRST, whereas the advantage of TextRank is not significant. These results only confirm the conclusion made in the previous works [11,12,13], where it was empirically shown that selection of the first sentences is the best individual sentence scoring metric. Table 4 presents for each system

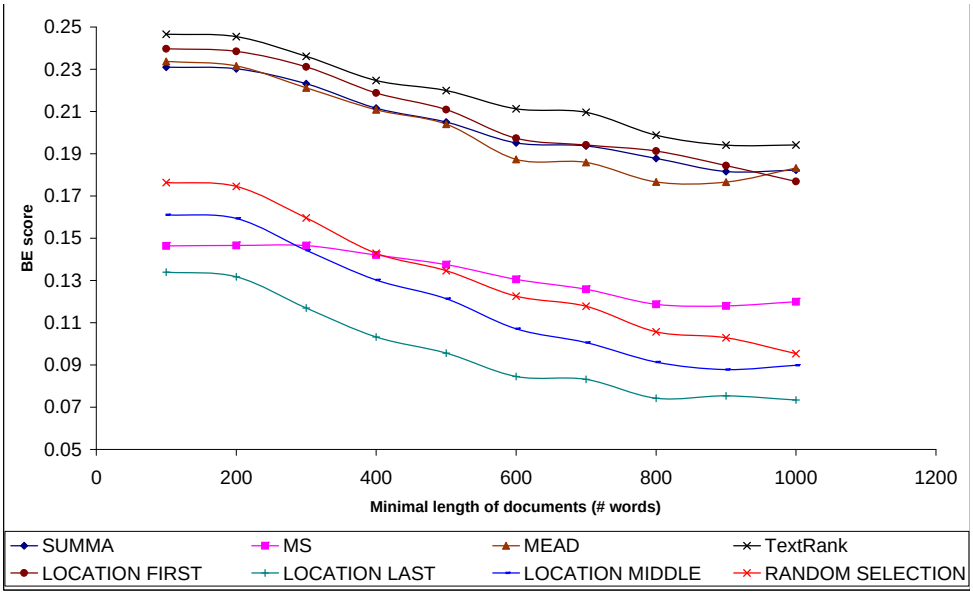

Figure 6. Does the length matter?

Table 4. Percentage of best BE scores

System	# documents	Percentage (%)
TextRank	188	35
MEAD	124	23
SUMMA	113	21
MS	22	4

Table 5. BE scores for DUC 2002

Summarizer	BE score	Ratio(TextRank)	Ratio(MEAD)	Ratio(SUMMA)	Ratio(MS)
Best (G)	0.2943	0.84	0.79	0.78	0.50
Avg	0.2426	1.02	0.96	0.95	0.60
Worst (E)	0.2071	1.19	1.13	1.12	0.71

the absolute number and percentage of documents on which the system performed best. There are cases where more than one system gets the same best score. In such a situation we give a point to all these systems, but we do not count the cases when all the systems get a zero score. The systems in the table are ordered from the best to the worst. Table 5 contains relative ratios between the system score and the best, average and worst scores for human summarizers from DUC 2002 and each evaluated system accordingly.

According to our comparative results, TextRank is the best summarizer among the evaluated systems. Moreover, TextRank can be applied to multiple languages if it does not perform any morphological analysis of the text. We use language-independent implementation of TextRank in the comparative evaluation of the MUSE algorithm presented in this chapter.

4.5. The MUSE Algorithm: Experimental Results

The MUSE algorithm evaluation is based on the ROUGE-1 Recall metric computed using 10-fold cross-validation. Table 6 contains the average ROUGE values for the results of training and testing on three corpora: English, Hebrew and bilingual respectively. Since we have a different number of English and Hebrew documents (533 and 50 respectively), in order to evaluate our approach on bilingual corpora, 10 balanced corpora with the same number of English and Hebrew documents have been created by combining approximately 50 randomly selected English documents with all 50 Hebrew documents. Then, 10-fold cross-validation was performed on each corpus and the average results for training and testing were calculated.

We compared our approach with the multilingual version of TextRank (denoted by *ML_TR*) [31] as the best known multilingual summarizer, English and Hebrew versions of Microsoft Word Autosummarize method (denoted by *MS_SUM*) as a widely used commercial summarizer, and the best individual scoring method in each corpus. As a baseline we took summaries combined from the first sentences (denoted by *POS_F*). Table 7 demonstrates the comparative results (ROUGE mean value) on English, Hebrew and bilingual corpora respectively, with the best summarizers on top. Pairwise comparisons between summarizers indicate that all compared methods are significantly different at the 95% confidence level, except *POS_F* and *ML_TR* in English and bilingual corpora, and *D_COV_J* and *POS_F* in the Hebrew corpus. MUSE significantly outperforms

TextRank and the best single metrics in all three corpora: *COV_DEG* [55] in English and *D_COV_J* [55] in Hebrew corpora respectively.

Two sets of features were evaluated on the bilingual corpus: the full set of 31 sentence scoring metrics and 10 best multilingual metrics determined in [55] by a clustering analysis of the methods' results on both corpora. The experimental results show that the optimized combination of 10 best metrics is not significantly distinguishable from the best single metric in the bilingual corpus, *COV_DEG*, based on our word-based graph representation. The difference between the combination of all 31 metrics and *COV_DEG* is significant only with a one-tailed p value of 0.0798 (considered not quite statistically significant). Both combinations significantly outperform all other compared summarizers. Table 7 contains results of *MUSE* using the full set of 31 sentence scoring metrics.

Our experiments show that removing highly correlated metrics (one metric, with lower ROUGE value, out of each pair of highly correlated metrics) from the linear combination slightly improves the summarization quality, though the improvement is not statistically significant. Discarding bottom ranked features (up to 50%) also does not affect the results significantly.

Table 8 demonstrates the best vectors induced from training *MUSE* on all the documents in English, Hebrew and bilingual (one of 10 balanced) corpora along with their training ROUGE scores and the number of GA iterations. While one might expect the optimal values of the weights to be nonnegative numbers, the actual result is such that some of the weights are definitely negative. Although there is no simple explanation for this amazing fact, it could be compared to the well-known phenomenon from Numerical Analysis called *over-relaxation* [56]. For example, solving the Laplace equation $\phi_{xx} + \phi_{yy} = 0$ over a grid of points is done iteratively as follows: at each grid point let $\phi^{(n)}, \overline{\phi}^{(n)}$ denote the n^{th} iteration as calculated from the differential equation and its *modified* final value respectively. The final value is chosen as $\omega\phi^{(n)} + (1-\omega)\overline{\phi}^{(n-1)}$. While the sum of the two weights is obviously 1, the *optimal* value of ω, which minimizes the number of iterations needed for convergence, usually satisfies $1 < \omega < 2$ (i.e., the second weight $1-\omega$ is negative) and approaches 2 the finer the grid gets. This surprising result, though somewhat unexpected, can be rigorously proved [57].

We have found that "good" features (having a high rating in the experiment described in [55]) have relatively high positive weights, whereas "bad" features have near-zero or low negative weights. Also, some features, that may be considered as unimportant, have a "neutral" behavior expressed by the near-zero weights in the optimized vectors. For example:

- *POS_L* and *GRASE* have negative weights in all vectors. Intuitively, we may expect these features to get near-zero weights as features that received low rank in both corpora and do not contribute to the quality of summarization. In case of confirmation of our expectations, we could recommend not to include them in the linear combination at all. Yet, the fact that they got negative but different from zero weights means that they affect the results and cannot be excluded from the combination.
- *LEN_CH* and *TITLE_E_J* have close ratings in both monolingual corpora and relatively close (negative or near-zero) values in optimized vectors for both corpora.
- *COV_DEG*, *KEY* and *D_COV_J* that are among the best multilingual metrics and have close ratings in both monolingual corpora, have positive weights in all

three vectors. Also, *POS_F*,[12] having high ranks, 8 and 4 in English and Hebrew corpora respectively, has all positive weights, and so do *LUHN_DEG* (ranks 7 and 6) and *ML_TR*[13] (ranks 11 and 15).

Frequency-based metrics have been empirically shown to be mostly detrimental in summarization systems [11,12,13]. Our results for *TFISF, SVD, TF, LUHN* partially confirm this estimation. It is interesting that most metrics have both positive and negative weights in different corpora. Some features, like *TITLE_J*, *TITLE_O*, *POS_B* have different ratings in different languages and their results reflect this behavior, but there are also other features (*TITLE_C, LEN_W, D_COV_C*, etc.), whose weights do not have an intuitive explanation. It is known that Genetic Algorithms can produce sub-optimal results trapped in the local optima. So, sometimes it can explain "strange" counter-intuitive results. However, Genetic Algorithm is known as a good optimization technique to the problems having complex search space landscape, where there is no simple smooth fitness function, due to its ability to cover a wide search space in combination with a random factor (via the mutation operator). Since we did not consider fitness (ROUGE as a summarization quality) as a simple smooth function of the 31 features, we did not expect fully intuitive and understandable results. Due to the lack of previously published results for the optimized weighted linear combinations of various sentence features, we have nothing to compare our results with.

Figures 9 and 11 depict examples of a gold standard summary and automatically generated summaries for the best individual method in each corpora (*COV_DEG* and *D_COV_J* for English and Hebrew respectively) and the optimized combination of methods (MUSE) for two documents: English *AP880228-0097* and Hebrew *doc1*. Summaries for English *AP880217-0100* document (see Figure 4) are also shown in Figure 12.

Table 6. MUSE: Results of 10-fold cross validation. Mean ROUGE-1 Recall.

	ENG	HEB	MULT
Train	0.4483	0.5993	0.5205
Test	0.4461	0.5936	0.5027

Table 7. Summarization performance. Mean ROUGE-1 Recall.

Metric	ENG	HEB	MULT
MUSE	**0.4461**	**0.5921**	**0.4633**
COV_DEG	0.4363	0.5679	0.4588
D_COV_J	0.4251	0.5748	0.4512
POS_F	0.4190	0.5678	0.4440
ML_TR	0.4138	0.5190	0.4288
MS_SUM	0.3097	0.4114	0.3184

Assuming an efficient implementation, most of the metrics have a linear computational complexity relative to the total number of words in the document, $O(n)$ [55], and, as a result, MUSE total computation time given a trained model is linear as well (at factor of

[12] It was empirically shown to be the best individual method in [11,12].
[13] The best state-of-the-art multilingual summarizer.

Table 8. Induced weights for the best linear combination of scoring metrics.

Metric	ENG	HEB	MULT
COV_DEG*	8.490	0.171	0.697
KEY_DEG*	15.774	0.218	-2.108
KEY	4.734	0.471	0.346
COV_PR*	-4.349	0.241	-0.462
COV	10.016	-0.112	0.865
D_COV_C*	-9.499	-0.163	1.112
D_COV_J*	11.337	0.710	2.814
KEY_PR*	0.757	0.029	-0.326
LUHN_DEG*	6.970	0.211	0.113
POS_F	6.875	0.490	0.255
LEN_CH	1.333	-0.002	0.214
LUHN	-2.253	-0.060	0.411
LUHN_PR*	1.878	-0.273	-2.335
LEN_W	-13.204	-0.006	1.596
ML_TR	8.493	0.340	1.549
TITLE_E_J*	-5.551	-0.060	-1.210
TITLE_E_O*	-21.833	0.074	-1.537
D_COV_E_J*	1.629	0.302	0.196
D_COV_O*	5.531	-0.475	0.431
TFISF	-0.333	-0.503	0.232
DEG*	3.584	-0.218	0.059
D_COV_E_O*	8.557	-0.130	-1.071
PR*	5.891	-0.639	1.793
TITLE_J	-7.551	0.071	1.445
TF	0.810	0.202	-0.650
TITLE_O	-11.996	0.179	-0.634
SVD	-0.557	0.137	0.384
TITLE_C	5.536	-0.029	0.933
POS_B	-5.350	0.347	1.074
GRASE*	-2.197	-0.116	-1.655
POS_L	-22.521	-0.408	-3.531
Score	**0.4549**	**0.6019**	**0.526**
Iterations	10	6	7

the number of metrics in a combination). The training time is proportional to the number of GA iterations multiplied by the number of individuals in a population times summarization time. In all our experiments, GA performed between 1 to 17 iterations – selection and reproduction continued, before reaching convergence, for $5-6$ iterations on average. Figure 7 demonstrates a convergence chart for one of the GA runs, with maximal number of iterations – training on English corpus (training set containing $9/10$ of 533 documents), given 25 non-correlated metrics for optimization.

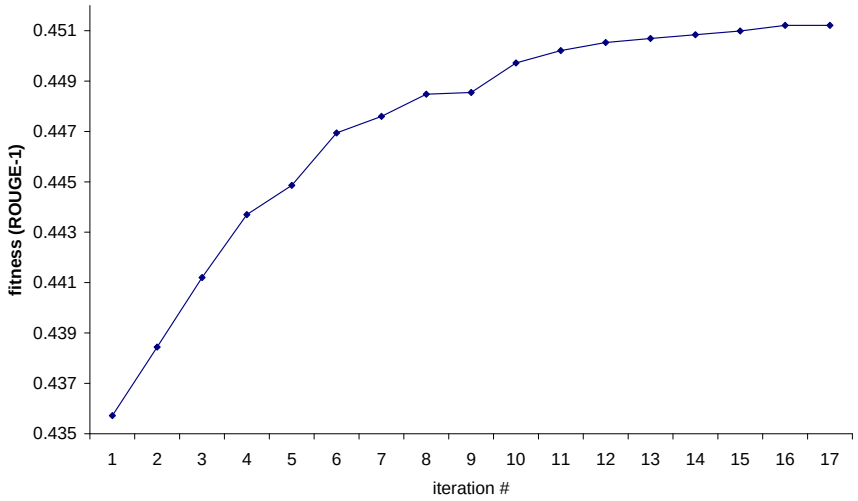

Figure 7. Convergence chart for GA run.

5. Conclusions and Future Work

In this chapter, we described novel methods for language-independent sentence extraction. We use statistical ranking methods which do not embed any language-specific knowledge. Some of our summarization methods are based on a graph text representation that utilizes structural information about a document. We presented MUSE – a new GA-based approach to multilingual extractive summarization aimed at finding an optimal linear combination of multiple sentence scoring metrics. We evaluated the proposed methodology on two different languages: English and Hebrew. The experimental results show that MUSE significantly outperforms TextRank – the best state-of-the-art multilingual approach, in both languages using either monolingual or bilingual corpora. Moreover, our results suggest that the same weighting model may be applicable across two different languages.

In the future, our methods may be evaluated on additional graph representations of documents. All word-based graph representations introduced in [45] can be applied to language-independent summarization. For language-specific summarization, a concept-based representation where the graphs are built from the concepts fused from the texts may be developed.

The evaluations of language-independent keyphrase and sentence ranking metrics can be extended to other languages such as German, Arabic, Greek and Russian. If needed, new benchmark corpora can be generated for any language by native speakers using such tool as TAHA (see Section 4.3).

One can use additional techniques for summary evaluation and study the impact of morphological analysis on the top ranked multilingual scores using part-of-speech (POS) tagging[14], anaphora resolution, named entity recognition, and word sense disambiguation.

[14]Our experiments have shown that syntactic filters, selecting only lexical units of a certain part of speech, do not significantly improve the performance of the evaluated multilingual scoring methods.

1	America's Showing Worst In 52 Years
2	America's Showing Worst In 52 Years Time ran out for the U.S. athletes when the Winter Olympics ended Sunday, with a team headed by Brent Rushlaw a tick away from a U.S. bobsled medal and Dutch speed skater Yvonne van Gennip a triple gold medalist with time to spare.
3	The best America could do was six medals, its worst Winter Games showing in 52 years.
4	Two of the six were gold medals, won by Brian Boitano in figure skating and Bonnie Blair in speed skating.
5	Blair also won a bronze in speed skating, making her America's only double winner.
6	The Olympics wrapped up Sunday evening with a rousing closing ceremony before 60,000 people in McMahon Stadium.
7	The 250 figure skaters who performed included past and present medal winners as well as young skaters from Albertville, France, site of the 1992 Winter Games, and Seoul, South Korea, site of this year's summer Games.
8	Athletes marched in carrying miniature Olympic torches, and banners reading Until We Meet Again" in eight languages were draped along the top of the stands.
9	Earlier Sunday, Rushlaw and his three team members missed winning the first U.S. bobsled medal in 32 years when they was beaten by .02 seconds for the bronze.
10	Van Gennip took more than six seconds off her own world record in the 5,000 meters to win her third gold medal.
11	American Mary Docter was 22.87 seconds behind.
12	East Germans finished 2-3 in the race.
13	Finland won the hockey silver medal, handing the Soviets their first loss of the Games, 2-1.
14	The Soviets clinched the gold medal Friday night, and America finished seventh for the second straight time.
15	In 1936, America won just four medals, but only 51 were available then.
16	This time, 138 medals were handed out.
17	The Soviets finished first in both number of gold medals, with 11, and total medals, with 29, a record for the Winter Olympics.
18	East Germany's athletes won nine golds, 25 overall; while Switzerland's team came in third in both gold medals, five, and overall medals, 15.
19	Four other teams also bested the United States in terms of overall medals: Austria, West Germany, Finland and the Netherlands.

Figure 8. English: AP880228 text document (19 sentences).

Also, the optimization process can be enhanced in order to find the best combination of features for sentence extraction by applying different optimization techniques such as Evolution Strategy, known to perform well in a real-valued search space, and reducing the search for the best summary to the problem of multi-objective optimization, combining several summary quality metrics that need to be maximized (i.e precision, recall, F-measure, etc.) with some “information redundancy"(i.e. content overlap between sentences, etc.) metrics that need to be minimized.

Similar methods can be also applied to the task of multi-document summarization. Graph matching techniques can be used for this task. Also, efficient application of operations on graphs can be useful in update and query-based summarization.

References

[1] Filippova K, Surdeanu M, Ciaramita M, Zaragoza H. Company-oriented extractive summarization of financial news. In: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics; 2009. p. 246–254.

[2] Mani I, Maybury MT. Advances in Automatic Text Summarization. MIT Press, Cambridge, MA,; 1999.

[3] Salton G, Singhal A, Mitra M, Buckley C. Automatic text structuring and summarization. Information Processing and Management. 1997;33(2):193–207.

The Olympics wrapped up Sunday evening with a rousing closing ceremony in Calgary's McMahon Stadium.
The best America could do was six medals, its worst Winter Games showing in 52 years.
In 1936, America won just four medals, but only 51 were available then.
This time, 138 medals were handed out.
America's bobsled team missed a bronze medal by .02 seconds.
The Soviets finished first in both number of gold medals, 11, and in total medals with a record 29.
East Germany's athletes won nine golds, 25 overall; while Switzerland's team came in third in both gold, five, and overall with 15.

(a) summary by human (abstract)

2	America's Showing Worst In 52 Years Time ran out for the U.S. athletes when the Winter Olympics ended Sunday, with a team headed by Brent Rushlaw a tick away from a U.S. bobsled medal and Dutch speed skater Yvonne van Gennip a triple gold medalist with time to spare.
9	Earlier Sunday, Rushlaw and his three team members missed winning the first U.S. bobsled medal in 32 years when they was beaten by .02 seconds for the bronze.
17	The Soviets finished first in both number of gold medals, with 11, and total medals, with 29, a record for the Winter Olympics.

(b) summary by COV_DEG

2	America's Showing Worst In 52 Years Time ran out for the U.S. athletes when the Winter Olympics ended Sunday, with a team headed by Brent Rushlaw a tick away from a U.S. bobsled medal and Dutch speed skater Yvonne van Gennip a triple gold medalist with time to spare.
4	Two of the six were gold medals, won by Brian Boitano in figure skating and Bonnie Blair in speed skating.
9	Earlier Sunday, Rushlaw and his three team members missed winning the first U.S. bobsled medal in 32 years when they was beaten by .02 seconds for the bronze.

(c) summary by MUSE

Figure 9. English: summary examples for text document from Figure 8.

[4] Luhn HP. The automatic creation of literature abstracts. IBM Journal of Research and Development. 1958;2:159–165.

[5] Turney PD. Learning Algorithms for Keyphrase Extraction. Information Retrieval. 2000;2(4):303–336.

[6] Mihalcea R, Tarau P. TextRank – bringing order into texts. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing; 2004. .

[7] ;.

[8] Litvak M, Last M. Graph-based keyword extraction for single-document summarization. In: Proceedings of the Workshop on Multi-source Multilingual Information Extraction and Summarization. Association for Computational Linguistics; 2008. p. 17–24.

[9] Gulli A, Signorini A. The Indexable Web is More than 11.5 Billion Pages; 2005. http://www.cs.uiowa.edu/ asignori/web-size/.

[10] Melby AK, Warner T. The Possibility of Language. Benjamins Translation Library 14; 1995.

[11] Edmundson HP. New Methods in Automatic Extracting. ACM. 1969;16(2).

[12] Kupiec J, Pedersen J, Chen F. A trainable document summarizer. In: Proceedings of the 18th annual international ACM SIGIR conference; 1995. p. 68–73.

[13] Teufel S, Moens M. Sentence Extraction as a Classification Task. In: Proceedings of the Workshop on Intelligent Scalable Summarization, ACL/EACL Conference; 1997. p. 58Ű–65.

1	45 שנות מאסר לגבריאל תורג'מן "הפדופיל החמקן" בית המשפט בבאר שבע גזר עונש מאסר חסר תקדים לעבירות מין על גבריאל תורג'מן, שהורשע באינוס ובביצוע מעשי סדום בחמש ילדות בנות חמש עד תשע.
2	עונש חסר תקדים על עבירות מין נגזר היום (שני) על גבריאל תורג'מן, שירצה 45 שנות מאסר בפועל בגין אונס, מעשים מגונים וביצוע מעשי סדום בחמש ילדות בנות חמש עד תשע.
3	תורג'מן, תושב אשדוד בן 51, שנודע לשמצה בכינוי הפדופיל החמקן", חטף את קורבנותיו ליד ביתן בין השנים 2002-2004.
4	הוא הסיע אותן למקום שומם, שם תקף אותן מינית תוך שימוש בכוח ובאיומים.בנוסף הורשע תורג'מן בהדחה בחקירה.
5	הפדופיל הסדרתי הוא נשוי, אב לארבעה ילדים ובעל עבר פלילי עשיר.
6	תורג'מן ישב בכלא שנים ארוכות בגין עבירות סמים ורכוש.
7	במהלך חקירתו במשטרה ולכל אורך המשפט הכחיש את כל האישומים שיוחסו לו.
8	על פי כתב האישום, תורג'מן נהג לתקוף את קורבנותיו באכזריות רבה.
9	את הילדות שחטף היה מכה וחונק, ונהג להפשיטן ולבצע בהן את זממו באיומים.
10	על אחת מהילדות אף איים בסכין.
11	תורג'מן הקפיד להסיע בחזרה לביתן ארבע מהילדות שתקף ,ואילו החמישית, שהצליחה להימלט ממנו לאחר מעשה, נותרה במקום השומם שבו הותקפה, ולבסוף הצליחה לשוב לביתה בכוחות עצמה.
12	המשטרה הפעילה מאמצים רבים ללכוד את "הפדופיל החמקן".
13	לבסוף, פחות מחודש לאחר מקרה התקיפה האחרון שבו הואשם ,זיהו אותו שוטרי סיור כדומה לקלסתרון שהרכיבו ילדות שהותקפו.
14	דגימות דנ"א שניטלו ממנו הושוו עם דגימות שנלקחו מזירת התקיפה האחרונה, ולאחר שנמצאו מתאימות, נעצר.
15	הסבל של הילדות הפך למנת חלקם של ההוריםכעבור שבועות אחדים התברר כי דגימות הדנ"א שלו מתאימות גם לממצאים בזירת תקיפה שאירעה תשע שנים קודם לכן, בה תקף ילדה בת תשע.
16	תורג'מן הואשם בעבר באותה תקיפה אך בית המשפט מצא אותו זכאי.
17	לאחר מעצרו ביקשה המשטרה להסיר את צו איסור הפרסום משמו ותמונתו, כדי לאפשר לקורבנות אחרים לזהותו, אך סניגורו עתר לבית המשפט המחוזי, ואחר כך לבית המשפט העליון.
18	לבסוף החליט השופט מישאל חשין להתיר את פרסום פרטיו.
19	הרכב של שלושה שופטים בראשות השופטת רויטל יפה-כץ, גזר על תורג'מן 45 שנות מאסר בפועל, שבהן כלול עונש של 15 שנות מאסר על כל אחד מהמקרים שבהם הורשע.
20	עם זאת, את עונשיו ירצה הפדופיל באופן חופף ומצטבר.
21	בגזר הדין כתבה השופטת יפה-כץ כי "הנאשם הותיר אחריו חמישה קורבנות, חמש משפחות הרוסות ,חמש נפשות פצועות ומיוסרות.
22	הכאב והסבל של הילדות הפכו גם למנת חלקם של ההורים".
23	על רקע העונש הכבד שגזרו השופטים, הדגישה כי"אסור לנו לשכוח שמעשי האונס האכזריים הציתו אימה ופניקה בקרב הציבור הרחב.
24	למרות שהארץ רעשה וגעשה לנוכח מעשיו, הדבר לא הרתיע אותו מלשוב ולבצע מעשים נוספים."

Figure 10. Hebrew: doc1 text document titled “45 years in prison for Gabriel Turgeman, “elusive pedophile"", published at 06 April, 2009 (24 sentences).

[14] Radev D, Blair-Goldensohn S, Zhang Z. Experiments in single and multidocument summarization using MEAD. First Document Understanding Conference. 2001;.

[15] Saggion H, Bontcheva K, Cunningham H. Robust generic and query-based summarisation. In: EACL ’03: Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics; 2003. .

[16] Goldstein J, Kantrowitz M, Mittal V, Carbonell J. Summarizing Text Documents: Sentence Selection and Evaluation Metrics. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval; 1999. p. 121–128.

[17] Figuerola CG, Berrocal JLA, Zazo AF, Diaz RG. In: A Simple Approach to the Spanish-English Bilingual Retrieval Task. vol. 2069. Series Lecture Notes in Computer Science, Springer Berlin; 2001. p. 224–229.

[18] Littman M, Dumais ST, Landauer TK. Automatic Cross-Language Information Retrieval using Latent Semantic Indexing. In: Cross-Language Information Retrieval, chapter 5. Kluwer Academic Publishers; 1998. p. 51–62.

[19] Gong Y, Liu X. Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis. In: Proceedings of the 24th ACM SIGIR conference on Research and development in information retrieval; 2001. p. 19–25.

[20] Steinberger J, Jezek K. Text summarization and singular value decomposition. Lecture Notes in Computer Science. 2004;p. 245–254.

[21] Brown RD. Automated Generalization of Translation Examples. In: Proceedings of the Eighteenth

1	45 שנות מאסר לגבריאל תורג'מן "הפדופיל החמקן" בית המשפט בבאר שבע גזר עונש מאסר חסר תקדים לעבירות מין על גבריאל תורג'מן, שהורשע באינוס ובביצוע מעשי סדום בחמש ילדות בנות חמש עד תשע.
2	עונש חסר תקדים על עבירות מין נגזר היום (שני) על גבריאל תורג'מן, שירצה 45 שנות מאסר בפועל בגין אונס, מעשים מגונים וביצוע מעשי סדום בחמש ילדות בנות חמש עד תשע.
23	על רקע העונש הכבד שגזרו השופטים, הדגישה כי"אסור לנו לשכוח שמעשי האונס האכזריים הציתו אימה ופניקה בקרב הציבור הרחב.
24	למרות שהארץ רעשה וגעשה לנוכח מעשיו, הדבר לא הרתיע אותו מלשוב ולבצע מעשים נוספים."

(a) summary by human

1	45 שנות מאסר לגבריאל תורג'מן "הפדופיל החמקן" בית המשפט בבאר שבע גזר עונש מאסר חסר תקדים לעבירות מין על גבריאל תורג'מן, שהורשע באינוס ובביצוע מעשי סדום בחמש ילדות בנות חמש עד תשע.
2	עונש חסר תקדים על עבירות מין נגזר היום (שני) על גבריאל תורג'מן, שירצה 45 שנות מאסר בפועל בגין אונס, מעשים מגונים וביצוע מעשי סדום בחמש ילדות בנות חמש עד תשע.
19	הרכב של שלושה שופטים בראשות השופטת רויטל יפה-כץ, גזר על תורג'מן 45 שנות מאסר בפועל, שבהן כלול עונש של 15 שנות מאסר על כל אחד מהמקרים שבהם הורשע.

(b) summary by D_COV_J

1	45 שנות מאסר לגבריאל תורג'מן "הפדופיל החמקן" בית המשפט בבאר שבע גזר עונש מאסר חסר תקדים לעבירות מין על גבריאל תורג'מן, שהורשע באינוס ובביצוע מעשי סדום בחמש ילדות בנות חמש עד תשע.
2	עונש חסר תקדים על עבירות מין נגזר היום (שני) על גבריאל תורג'מן, שירצה 45 שנות מאסר בפועל בגין אונס, מעשים מגונים וביצוע מעשי סדום בחמש ילדות בנות חמש עד תשע.
3	תורג'מן, תושב אשדוד בן 51, שנודע לשמצה בכינוי הפדופיל החמקן", חטף את קורבנותיו ליד ביתן בין השנים 2002-2004.
4	הוא הסיע אותן למקום שומם, שם תקף אותן מינית תוך שימוש בכוח ובאיומים.בנוסף הורשע תורג'מן בהדחה בחקירה.

(c) summary by MUSE

Figure 11. Hebrew: summary examples for text document from Figure 10.

International Conference on Computational Linguistics (COLING-2000); 2000. p. 125–131.

[22] Hofmann T. Probabilistic Latent Semantic Analysis. In: In Proc. of Uncertainty in Artificial Intelligence, UAIŠ99; 1999. p. 289–296.

[23] Jiang N. Lexical representation and development in a second language. Applied Linguistics. 2000;21(1):47–77.

[24] Vinokourov A, Shawe-taylor J, Cristianini N. Inferring a Semantic Representation of Text via Cross-Language Correlation Analysis. In: In NIPS, volume 15. MIT Press; 2002. p. 1473–1480.

[25] Chen HH, Lin CJ. A multilingual news summarizer. In: Proceedings of the 18th International Conference on Computational Linguistics; 2000. p. 159–165.

[26] Nie JY, Isabelle P, Plamondon P, Foster G. Using a Probabilistic Translation Model for Cross-Language Information Retrieval. In: 6th Workshop on Very Large Corpora. Morgan Kaufmann Publishers; 1998. p. 18–27.

[27] Ogden W, Cowie J, Davis M, Ludovik E, Molina-salgado H, Shin H. Getting Information from Documents You Cannot Read: An Interactive Cross-Language Text Retrieval and Summarization System. In: Joint ACM DL/SIGIR Workshop on Multilingual Information Discovery and Access; 1999. http://www.clis.umd.

[28] Evans D, McKeown K. Identifying similarities and differences across english and arabic news. In: Proceedings of International Conference on Intelligence Analysis; 2005. p. 23–30.

[29] Siddharthan A, Mckeown K. Improving Multilingual Summarization: Using Redundancy in the Input to Correct MT errors. 2008;.

[30] Erkan G, Radev DR. LexRank: Graph-based lexical centrality as salience in text summarization. Journal of Artificial Intelligence Research. 2004;22:457–479.

The Oscar, created 60 years ago by MGM art director Cedric Gibbons, weighs 8 pounds and stands 13 inches tall.

It is made of Britannia metal and plated with copper, nickel, and gold.

From 1942-44 it was made of plaster.

It is manufactured by Chicago's R.S. Owens Co. 1,816 have been produced so far.

The only engraving mistake was "Dick Tracy" instead of "Spencer Tracy" in 1938.

The Academy of Motion Picture Arts and Sciences reserves the first right to buy an Oscar from anyone.

"All About Eve" is the most-nominated film (14), Ben Hur the most awarded (11), and Walt Disney the most- awarded person (32).

(a) summary by human (abstract)

2	Statistics on Six Decades of Oscar With PM-Oscar Nominations Bjt The motion picture industry's most coveted award, Oscar, was created 60 years ago and 1,816 of the statuettes have been produced so far.
3	Weighing 8 pounds and standing 13 inches tall, Oscar was created by Metro-Goldwyn-Mayer studios art director Cedric Gibbons, who went on to win 11 of the trophies.
6	According to the Academy of Motion Pictures Arts and Sciences, the only engraving mistake was in 1938 when the best actor trophy given to Spencer Tracy for Boy's Town" read: Best Actor: Dick Tracy."

(b) summary by COV_DEG, MUSE (identical)

Figure 12. English: summary examples for text document from Figure 4.

[31] Mihalcea R. Language independent extractive summarization. In: AAAI'05: Proceedings of the 20th national conference on Artificial intelligence; 2005. p. 1688–1689.

[32] Baxendale PB. Machine-made index for technical literaturĚUan experiment. IBM Journal of Research and Development. 1958;2(4):354–361.

[33] Satoshi CN, Satoshi S, Murata M, Uchimoto K, Utiyama M, Isahara H. Sentence Extraction System Assembling Multiple Evidence. In: Proceedings of 2nd NTCIR Workshop; 2001. p. 319–324.

[34] Kallel FJ, Jaoua M, Hadrich LB, Hamadou AB. Summarization at LARIS Laboratory. In: Proceedings of the Document Understanding Conference; 2004. .

[35] Vanderwende L, Suzuki H, Brockett C, Nenkova A. Beyond SumBasic: Task-focused summarization with sentence simplification and lexical expansion. Information processing and management. 2007;43(6):1606–1618.

[36] Neto JL, Santos AD, Kaestner CAA, Freitas AA. Generating text summaries through the relative importance of topics. Lecture Notes in Computer Science. 2000;p. 300–309.

[37] Brin S, Page L. The anatomy of a large-scale hypertextual Web search engine. Computer networks and ISDN systems. 1998;30(1-7):107–117.

[38] Tombros A, Sanderson M, Gray P. Advantages of query based summaries in information retrieval. In: Working notes of the AAAI Spring Symposium on Intelligent Text Summarization; 1998. p. 44–52.

[39] Paice C, Jones P. The Identification of Important Concepts in highly structured Technical Papers. In: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval; 1993. p. 69–78.

[40] Jing HR, Barzilay R, McKeown K, Elhadad M. Summarization evaluation methods: Experiments and analysis. In: Working notes of the AAAI Spring Symposium on Intelligent Text Summarization; 1998. p. 60–68.

[41] Maybury M. Generating Summaries from Event Data. Information Processing and Management. 1995;31(5):735–751.
[42] Lin CY. ROUGE: A Package for Automatic Evaluation of summaries. In: Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004); 2004. p. 25Ű–26.
[43] DUC. Document Understanding Conference; 2001–2007. http://duc.nist.gov.
[44] Schenker A, Bunke H, Last M, Kandel A. Classification of Web Documents Using Graph Matching. International Journal of Pattern Recognition and Artificial Intelligence. 2004;18(3):475–496.
[45] Schenker A, Bunke H, Last M, Kandel A. Graph-theoretic techniques for web content mining; 2005.
[46] Dasgupta S, Papadimitriou CH, Vazirani UV. Algorithms; 2006. http://www.cs.berkeley.edu/ vazirani/algorithms.html.
[47] Nastase V, Szpakowicz S. A study of two graph algorithms in topic-driven summarization. In: Proceedings of the Workshop on Graph-based Algorithms for Natural Language; 2006. .
[48] Goldberg DE. Genetic algorithms in search, optimization and machine learning. Addison-Wesley; 1989.
[49] Litvak M, Friedman M, Last M. A new Approach to Improving Multilingual Summarization using a Genetic Algorithm. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL2010); 2010. p. 927–936.
[50] Vignaux GA, Michalewicz Z. A genetic algorithm for the linear transportation problem. IEEE Transactions on Systems, Man and Cybernetics. 1991;21:445–452.
[51] Lin CY, Hovy E. Automatic evaluation of summaries using N-gram co-occurrence statistics. In: NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology; 2003. p. 71–78.
[52] MS. Microsoft ®Office Word; 2003, 2007.
[53] Hovy E, Lin CY, Zhou L. Evaluating DUC 2005 using Basic Elements. In: Proceedings of DUC-2005; 2005. .
[54] Hovy E, Lin CY, Zhou L, Fukumoto J. Automated Summarization Evaluation with Basic Elements. In: In Proceedings of the Fifth Conference on Language Resources and Evaluation (LREC; 2006. .
[55] Litvak M, Kisilevich S, Keim D, Lipman H, Gur AB, Last M. Towards language-independent summarization: A comparative analysis of sentence extraction methods on English and Hebrew corpora. In: Proceedings of the Fourth International Workshop On Cross Lingual Information Access (CLIA2010) at COLING 2010; 2010. .
[56] Friedman M, Kandel A. Fundamentals of Computer Numerical Analysis. CRC Press; 1994.
[57] Varga RS. Matrix Iterative Methods. Prentice-Hall; 1962.