Summarizing Jewish Law Articles Using Genetic Algorithms

Yaakov HaCohen-Kerner, Eylon Malin, Itschack Chasson
Department of Computer Sciences, Jerusalem College of Technology (Machon Lev)
21 Havaad Haleumi St., P.O.B. 16031, 91160 Jerusalem, Israel

Abstract: People often need to make decisions based on different kinds of information. However, explosion of information is hard to handle. Summaries allow people to decide whether to read the whole text or not. In addition, they can serve as brief substitutes of full documents. This paper describes the first summarization model for texts in Hebrew. The summarization is done by extraction of the most relevant sentences. First, we have formulated nine known summarization methods and two unique Hebrew summarization methods. Then, we combined them into a hybrid method that achieves better results. Three machine learning methods have been tried: perceptron learning, Naive Bayesian learning, and genetic algorithm. The best results have been achieved by the genetic algorithm. To the best of our knowledge, our model is also the first to use successfully genetic algorithm for sentence extraction.

Key-Words: Genetic Algorithms, Hebrew, Hybrid Method, Jewish Law Articles, Machine Learning, Sentence Extraction, Text Summarization

1. Introduction

People often need to make decisions based on different kinds of information, but the explosion of information is hard to handle and reading everything may be very time consuming. Various kinds of summaries (e.g.: titles, abstracts, keywords, outlines, previews, reviews, biographies and bulletins) help reduce this problem. The introduction of summaries offers the readers the option whether or not to read the entire text. In addition, summaries can serve as brief substitutes of full documents.

Humans have the incredible ability to condense huge amounts of information. In general, they are known as excellent summarizers (Bartlett, 1983). However, the creation of summaries by people requires time, effort and money. Therefore, there has been an increase in the demand for research and development of automatic text summarization. Automatic text summaries can be produced with two main approaches: Natural Language Processing (NLP) and information extraction (IE).

Our model belongs to the sentence extraction approach. That is, it selects the most important sentences from the article and proposes them as a summary. In contrast to many summarization models that were designed and checked mostly for English articles taken from magazines and newspapers, our model deals with articles referring to Jewish law written in Hebrew.

This paper is arranged as follows: Section 2 gives background concerning text summarization, sentence extraction, machine learning in summarization systems and the Hebrew language. Section 3 describes our summarization model based on a hybrid method. Section 4 presents experiments that have been carried out, followed by various results. Section 5 presents the different machine learning methods we have applied and their results. Section 6 summarizes the research and offers a few proposals for future research.

2. Text summarization

Summarization is the process where an information object is reduced to a smaller size, and to its most important points [Alterman, 1992; Mani and Maybury, 1999]. Summaries can be produced with two main approaches: Natural Language Processing (NLP) and information extraction (IE). NLP is a field in artificial intelligence which attempts to use computers to either process information contained in an ordinary language such as English. Text summarization using NLP has been applied in several systems, e.g.: Aone et al. [97], McKeown and Radev [95], and Radev [99].

The extraction approach is simpler. It extracts parts of the original document (e.g.: keywords, sentences and paragraphs) and outputs the results as summaries. The sentence extraction method is the most popular. This method ranks sentences from the original text according to their salience or their likelihood of being a part of a summary. Text summarization using extraction of sentences has been applied in many systems, e.g.: Luhn [58], Edmundson [69], Kupiec et al. [95], Barzilay and Elhadad [97], and Hovy and Lin [99]. Text summarization using extraction of passages has been applied by Zechner [95] and using extraction of keywords has been applied by HaCohen-Kerner [03].

2.1. Text summarization based on sentence extraction

A study by Kupiec et al. [95] shows that 79% of the sentences in man-made abstracts in their corpus are extremely similar to sentences from the original article. In fact, some of the sentences were even extracted verbatim from the original article. Therefore, sentences extracted directly from the original text without being revised or rephrased can make quite an appropriate abstract. Summarization systems that work on the basis of sentence extraction usually rate sentences according to various features. Such features are discussed below.

2.2. Baseline methods for selecting the most important sentences

1) TF (Term Frequency): This method rates a sentence according to the number of terms (key words) that appear in the sentence. First, in order to distinguish between significant terms and other terms, the system will pass through the text, scoring each term according to the number of occurrences in the text. Words and terms that have a grammatical role for the language (e.g.: I, am, of, the) will be excluded from the key words list according to a ready-made stop list. Once the system has a list of key words and the number of their occurrences, the score of each sentence is calculated by the frequency of the key words that occur in it: $TF(s) = \sum_{\{t\} \in s} f(t)$ where $\{t\} \in s$ is the set of terms in a certain sentence s, and $f(t)$ refers to the frequency of t (i.e., the number of occurrences of the term t) throughout the whole text [Luhn, 58; Edmunson 69].

1) Cue words: This method rates a sentence according to the appearance of words and terms that indicate the importance of the sentence, e.g.: “the meaning of this is”, “for conclusion”, and “results”. The more cue words occur in the sentence, the higher score the sentence will be given: $CW(s) = \frac{CW_s}{CW_{\max}}$ where CW_s refers to the number of cue words in a certain sentence s, and $CW_{\max}$ refers to the number of cue words appearing in the sentence that contains the maximal number of cue words [Edmunson 69].

2) Sentence length: It is most probable that sentences that are very short are not included in a summary [Zechner, 96]. This method rates each sentence by dividing the number of its words by the number of the words in the longest sentence (in order to normalize the score): $SL(s) = \frac{length(s)}{length(s_{\max})}$ where s is the current sentence, $s_{\max}$ is the longest sentence [Lin 99].

3) Negative score: Some of the phrases indicate clearly on the sentences, in which they occur at that they do not belong to the summary. These phrases are defined as negative phrases, and will grant the sentences in which they appear a negative score. Examples for such phrases could be: “for example” or “it could be that”. The negative score is calculated as follows: $N(s) = -\frac{N_s}{N_{\max}}$ where N_s refers to the number of negative words in a certain sentence s, and $N_{\max}$ refers to the number of negative words appearing in the sentence that contains the maximal number of negative words [Myaeng, 99].

4) Sentence position: This method rates a sentence by its position relative to its paragraph, and according to the relative position of its paragraph within the article. The sentence position is calculated as follows: $sp(s) = val(pos, par)$ where pos is the position of the sentence in the paragraph, par is the paragraph number in the article, and val is a function that returns the score taking into consideration these two parameters. Return values of val are determined by statistical results [Edmunson, 69; Mani, 98; Lin, 97].

5) Centrality: It is assumed that a sentence that summarizes a few sentences has a big probability of being part of a summary. Taking this idea into consideration, the sentence is rated by the number of sentences it summarizes divided by the number of sentences in the article. The centrality score is calculated as follows: $C(s_i) = \frac{\sum_{s_j \in \{S - s_i\}} res(s_i, s_j)}{|S| - 1}$ where $res(s_i, s_j)$ is a function that checks the resemblance between the sentences s_i and s_j according to various parameters and $|S| - 1$ represents the number of the sentences in the whole article excluding the discussed sentence s_i [Neto, 02].

6) Resemblance to title: This method rates a sentence according to its resemblance to the title. Sentences that resemble the title will be granted a higher score. The resemblance to title score is calculated as follows: $TR(s) = res(s,t)$ where $res(s,t)$ is a function that ranks the resemblance between a

sentence s and the title t [Edmunson, 69; Mani and Bloedorn, 98; Neto, 02].

7) Resemblance to section title: This method rates a sentence according to its resemblance to the title of its section. Sentences that resemble the title of their section will be granted a higher score. The resemblance to section title score is calculated as follows: $STR(s) = res(s, st(s))$ where $res(s, t(s))$ is a function that checks resemblance between a sentence s and the title of its section $st(s)$ [Mani and Bloedorn, 98].

8) TF-ISF (term frequency - inverse sentence frequency): Key words occurring in fewer sentences are more probable to belong to the summary. This method extends the TF method and takes into consideration the ISF property that is calculated as follows: $ISF(t) = \frac{1}{\left|\{s \subset t\}\right|}$ where $\left|\{s \subset t\}\right|$ is the number of sentences containing the term t. The TF-ISF method gives a higher score to keywords appearing in fewer sentences. Since this feature is a weaker indicator than the term frequency, the keyword is multiplied by $\log_2(ISF)$ and not by the ISF score itself. The TF-ISF score is finally calculated as follows: $TF(s) = \sum_{\{t\}\in s} f(t) * \log_2 ISF(t)$ [Neto, 02].

2.3. Machine learning in summarization systems based on sentence extraction

Kupiec et al. [95] develop a summarization system based on the Naive Bayesian machine learning method. They investigate seven different features. The best results have been achieved with three of the features: position of the sentence in the text and in the paragraph (*paragraph*), occurrence of cue words (*fixed phrase*), and an indication of whether or not the sentence length was below a pre-specified number (*sentence-length cutoff feature*). They achieve an accuracy rate of 42% on the test set when their system produces an equal number of sentences similar to corresponding manual summaries.

Mani and Bloedorn [98] tested several machine learning techniques: C4.5, SCDF and AQ15c in order to discover features indicating the importance of a sentence. Their features were divided into three groups: location, thematic and cohesion features. The similarity between the query (the summary provided by the author) and each sentence is computed, and the n sentences most similar to the query are selected for the summary producing fixed-length summaries (typically 10% or 20% of the total number of sentences). Using 10-fold cross-validation the best result for generic summaries was obtained by C4.5, which achieved an accuracy rate of 69%.

Teufel and Moens [99] propose a technique that selects for inclusion in the summary a subset of sentences that preserves some information about the general rhetoric structure of the text. Examples of their technique include *indicator quality*, indicating the occurrence of meta-comments in the text; *indicator rhetorics*, modeling the rhetorical contribution of the phrases; and *header type*, specifying in which part of the text the sentence is included - e.g. “Introduction”, “Conclusion”, etc. In Experiments with Naive Bayesian performing cross-validation, the best result - achieved using all features but *indicator rhetorics* - was 66% of accuracy on the test set.

Neto et al. [00] propose a trainable system that automatically summarizes news and obtains an approximate argumentative structure of their text. They tested C4.5 and Naive Bayesian as machine learning methods. When producing summaries with 20% of sentences of the source documents, their system achieves an accuracy rate of 50.6% using C4.5.

2.4. The Hebrew language

Most of the models that were designed for text summarization were developed for the English language. However, there is no summarization system for Hebrew texts. Hebrew is a Semitic language. It uses the Hebrew alphabet and it is written from right to left. In this sub-section we would like to point out a few properties of the Hebrew language, which make the implementation of the model harder:

1) Tenses – most verbs in the English language differ from the base form only by one or more letters added at the end of the word. This makes words much easier to compare. Truncating all characters after the fifth [HaCohen-Kerner, 03] or sixth [Zechner, 96] character of the word would do the trick. In Hebrew, however, such a simple process may not be so helpful since the various forms change the basic form of the word in various ways. In some cases the same base form can have over 7000 (!) forms for different tenses and bodies. This feature of the Hebrew language makes it nearly impossible to compare two words without making a morphological analysis. For example, the two Hebrew words: (1) msvkm[1,2] (מסוכם , *mesukam*,

[1] The Hebrew Transliteration Table, which has been used in this paper, is taken from the web-site of the Princeton university library (http://infoshare1.princeton.edu/katmandu/hebrew/trheb.html).

summarized-passive), and (2) skmty (סיכמתי, *sikamty*, I summarized) are based on the same root skm (סכם , *sakem* , I summarized).

2) Word suffices – there are 5 letters in Hebrew, which are written differently when they appear at the end of the word. This feature of the Hebrew language also making it harder to compare two words. In the previous example, the Hebrew letter m (ם , *mem sofit* , final m) in the Hebrew word מסוכם, and the Hebrew letter m (מ , *mem* , m) from the Hebrew word סיכמתי are both derived from the same Hebrew letter m (מ , *mem* , m) in the Hebrew root skm (defined above). Although, in the first word it is written by another character since it is positioned at the end of the word.

3) Preposition letters – Unlike English that has unique words dedicated to express relations between objects (such as: in, at, and, from, since), Hebrew has 8 letters concatenated at the beginning of the word where each letter expresses another relation. For example, the Hebrew letter h (ה , ha, the) expresses the determiner ‘the’, and the letter m (מ , *mem* , from) expresses the preposition ‘from’.

4) Pronoun letters – English has unique words dedicated to ownership (such as: her, his, ours). Whereas Hebrew has letters concatenated to the end of the word to express such ownership. For example the Hebrew word m’mr (מאמר , *maamar*, article) means ‘article’, whereas the Hebrew word m’mry (מאמרי , *maamari*, my article) means ‘my article’.

5) Many Hebrew words can be written either in spelling when vowelization is added, e.g.: ks’ (כסא , *kise*, chair) or in spelling with letters denoting vowel sounds, e.g.: kis’ (כיסא , *kise*, chair).

6) Initials – initials are much more frequent in Hebrew than in English. Due to their frequency, ambiguous initials are frequent. For example, the initials ‘”‘(א"א , *alef alef*, a”a) have more than 100 different interpretations.

3. Our summarization model

In our previous work [HaCohen-Kerner et al., 03] we have formulized a basic summarization model that produces conclusive summaries for Jewish law articles written in Hebrew. This model does not have any machine learning capability. The best summarization method found in this research was a hybrid method composed of five different methods: TF-ISF, position, cue words, section title and domain in a linear combination. When producing summaries with 10% of sentences of the source documents, this basic system achieved recall/precision results of 42%/21%.

Our current model includes several significant extensions. Firstly, we investigate additional known summarization methods. Secondly, we develop several special summarization methods for our domain. Finally, we incorporate a machine learning component in order to improve our summarization capability. This component has been applied in three different machine learning methods (details in Section 5).

All nine of the methods mentioned in Section 2.1 have been implemented. Implementation of most of the methods for articles written in Hebrew was quite complex. The difficulties arise mostly in methods that are based on words comparison (e.g.: TF, centrality, title resemblance and section title resemblance) since it is hard to identify two words that have different forms on the one hand, but based on the same root on the other hand.

Many terms in the Hebrew corpus jargon are written by initials. The pronoun and preposition letters concatenated to words in Hebrew cause numerous problems as well. Comparison between terms is far more complicated under these circumstances. Even more so, such problems occur when implementing the methods based on sentences similarity. For the implementation of these methods, there was a need to cope with tenses and forms as well.

During the experiment phase we check the results of each method individually, and the results of various combinations of the nine methods mentioned in Section 2.1. Two methods, TF and centrality turned out to be so ineffective that we decide not to use them. TF that is included in TF-ISF was less effective than TF-ISF. Centrality was very ineffective and ran very slow comparing to other methods.

In addition, we have also developed and tested two new methods. The first method is the "domain method". This method is based on associative words classified according to various domains. At first, the system finds the text domain by seeking the most frequent key words, and then determines which domain they belong to (we have built a word list for each domain for that purpose). Once the domain is determined, key words belonging to this domain are rated accordingly. For example, under the domain ‘constitution and government’ keywords such as: democracy, liberality, and president, will be given higher scores than other keywords.

Another method that we have developed is psyk’ (פסיקה , *pesika*, ruling) cue words (we

[2] Each Hebrew word is presented in four forms: 1) transliteration of the Hebrew letters, (2) the Hebrew letters, (3) transliteration of the word as it actually pronounced, and (4) its translation into English.

call it ruling cue). This method is based on the nature of those articles to have rulings at their conclusion. Due to this nature, words like: forbidden, prohibited, from the outset, etc. will grant a higher score to the sentence that includes them.

In order to measure the success of our summarization methods mentioned above, we use the idea of Mani and Bloedorn [98]. They propose an automatic procedure for generation of reference summaries for articles with author-provided summaries. The main idea of their procedure is to choose the sentences having the closest resemblance (according to the cosine measure) to the sentences in the author-provided summary, in order to present them as summaries. It is quite obvious that one of the most significant components of such a procedure is the sentence comparison function. These reference summaries include only sentences taken from the articles and not from their author-provided summaries. Since our system is not allowed to extract sentences from the author-provided summaries, these reference summaries are much more convenient for comparison with the summaries generated by our system.

In order to measure the success of the summarization methods mentioned above we use the most popular measures which are the precision and recall measures. Usually, these measures are calculated by comparison to the ideal summaries (sentence extraction summaries made by human beings). In our model, they are calculated by comparison to the reference summaries.

Precision is defined by the number of sentences that appear both in the system's summary and in the reference summary divided by the number of sentences in the system's summary. Recall is defined by the number of sentences that appear both in the system's summary and in the reference summary divided by the number of sentences in the reference summary.

Our goal is to raise the recall rate as high as possible. The reason we have decided to focus mostly on this measure is that the main purpose of our system is to help one to get a rather short summary still including as many sentences as possible that appeared in the reference summary. We assume that a user would prefer to have most of the relevant sentences with a bit of unnecessary ones rather than having a pure significant text with many important sentences lacking. We have also taken into consideration the fact that the meaningless sentences can easily be filtered by a human subject.

4. Experimental results

Our corpus contains 60 articles referring to Jewish law written in Hebrew. Each one of the articles has its own conclusive author-provided summary. In the experiments, the summaries generated by our system have a length of about 10% of the original articles. We compare between them and the reference summaries we have produced. The results were awfully low. The highest recall result was 25%. However, as we read both summaries, we have found that the summaries we made by hand were much more indicative than the reference summaries.

The reason for this result seems to be that the cosine measure (which was the basis for the comparison between article sentences and the author-provided summary sentences) does not take into consideration some very significant factors. That is, the reference summaries have not been as indicative as we expected. The two main problems of this process of measuring are: (1) Not taking into consideration partial matches between pairs of similar words (e.g., write and written) (2) Not taking into consideration the importance of words to the text and its domain.

Therefore, we develop a new method for checking the resemblance between sentences. This method takes into consideration the four following factors:

1. The cosine measure.
2. Words that belong to the text issue will be given a higher matching rating. This factor will be calculated this way: $\dfrac{\#\text{of words appeared both in s1 and s2}}{\dfrac{|s1|+|s2|}{2}}$ where s1 and s2 are the compared sentences.
3. We define special Jewish Rabbinical conclusive cue words, as words that indicate conclusions, e.g.: must, forbidden, prohibited. Conclusive key words that belong to the text issue will be given a higher matching rating. This factor will be calculated this way: $\dfrac{\#\text{of conclusive cue words appeared both in s1 and s2}}{\dfrac{|s1|+|s2|}{2}}$ where s1 and s2 are the compared sentences.
4. Regular cue words that indicate the importance of the sentence (e.g.: for conclusion, to sum up) will be also given a higher matching rating. This factor will be calculated this way: $\dfrac{\#\text{of regular cue words appeared both in s1 and s2}}{\dfrac{|s1|+|s2|}{2}}$ where s1 and s2 are the compared sentences.

Each of these factors was multiplied by a coefficient as follows: $\alpha I + \beta C + \gamma R + \delta D$ where I is the issue words factor, C is the Jewish Rabbinical conclusive cue word factor, R is the regular cue words factor, D is the cosine measure method, $\alpha + \beta + \gamma + \delta = 1$ and $0 \leq \alpha, \beta, \gamma, \delta \leq 1$.

This comparison function yields not only much higher similarity between the summaries of our system and the reference summaries; it also yields even more indicative summaries for the latter as well.

The weight of each method was set by its recall value. The following formula defines a set of initial weights: $w_{recall}(F_i) = \frac{recall(F_i)}{\sum_{j=1}^{n} recall(F_j)}$

where n is the number of features and $recall(F_k)$ is the recall success rate of feature F_k. Experiments on our corpora yield the following results for our nine features:

Table 1. Weights (recall measures) of the features

# of feature	Feature	Weights
1	TF-ISF	0.19
2	Position	0.12
3	Ruling cue	0.10
4	Section title	0.11
5	Domain	0.23
6	Negative	0.03
7	Title	0.03
8	Length	0.11
9	Cue	0.08

We have taken the 9 features and their weights from Table 1, and combined them into a linear combination defined in Fig. 1 as our hybrid summarization method.

$$score(s) = \alpha * TF\_ISF(s) + \beta * POS(s) + \gamma * CUE(s) + \delta * ST(s) +$$
$$\varepsilon * LEN(s) + \phi * N(s) + \varphi * T(s) + \gamma * D(s) + \eta * RULING\_CUE(s)$$

Fig. 1. Our hybrid summarization method

Where TF_ISF refers to the TF-ISF method, POS(s) refers to the position method, *CUE*(s) refers to the regular cue words, ST(s) refers to the section title method, Length(s) refers to the length method, *N*(s) refers to the negative method, *T*(s) refers to the title method, *D*(s) refers to the domain method, and *RULING_CUE*(s) refers to the ruling cue words. Note that $\alpha + \beta + \gamma + \delta + e + \phi + \varphi + \gamma + \eta = 1$, and $0 \leq \alpha, \beta, \gamma, \delta, e, \phi, \varphi, \gamma, \eta \leq 1$.

As a result our hybrid method yields a recall rate of 39%. Although 61% of the sentences included in the system's summaries do not appear in the reference summaries, many of them are significant for understanding the main points and sources of the articles.

5. Machine learning

To improve the recall results of our hybrid method (39%) we have applied three common machine learning methods: perceptron learning, Naive Bayesian learning, and genetic algorithm. As mentioned in Section 2.3, the first two machine learning methods have been applied to sentence extraction systems. However, to the best of our knowledge, there is no sentence extraction system that uses genetic algorithm.

In order to test these machine learning methods we use the 10-fold cross-validation on the same corpus using the same comparison technique between the system's generated summaries and reference summaries as mentioned in Section 4.

5.1 The perceptron method

The perceptron training rule (Mitchell, 1997) presented in Fig. 2 is a simple machine learning method. The left w_i is the new weight of feature # i after the learning, the right w_i is the old weight before the learning, ε represents a small constant, (e.g.: 0.1, 0.01, in order to proceed in small and stable changes), t represents the training value, o represents the output value, and x_i represents the actual value of feature # i.

$$w_i = w_i + \varepsilon \cdot (t - o) \cdot x_i$$

Fig. 2. The perceptron training rule

In our system, the initial value of each feature's coefficient was set by dividing the rate of success of that feature (when performed by itself) by the sum of the rates of success of the other features (when performed by themselves). After the initial weights were set (Table 1), they were updated through the training process. The update is done in small and stable steps. Each sentence that had occurred in the reference summary and not in the summary that was generated by our hybrid method updated its own features by the following formula: $w(F_i) = w(F_i) + 0.005 * F_i(s)$

where $F_i(s)$ is a score of a sentence *s* by the feature i. Note that *s* is a sentence that was found in the reference summary that was made by using the author-made summary and not in the summary that was generated by our hybrid method. The value 0.005 was determined after many experiments. The machine learning process adjust the weights to the following values:

Table 2. Weights of features achieved by the perceptron training rule

# of feature	Feature	Weights
1	TF-ISF	0.20
2	Position	0.15
3	Ruling cue	0.08
4	Section title	0.08
5	Domain	0.15
6	Negative	0.07
7	Title	0.07
8	Length	0.16
9	Cue	0.05

Using the weights presented in Table 2, the recall rate of our hybrid method was raised by 0.01 (from 0.39 to 0.4).

5.2 The Naive Bayesian learning

Another well known machine learning method is the Naive Bayesian method. In sentence extraction, for each sentence *s* we compute the probability that it will be included in a summary S given its k features: $F_1, F_2, ...F_k$. This probability can be estimated by using Bayes' rules and assuming statistical independence of the features as follows:

$$p(s \in S \mid F_1, F_2, ...F_k) \propto p(s \in S) \prod_{j=1}^{k} p(F_j \mid s \in S)$$

Classifiers using this kind of estimation are called Naive-Bayes classifiers. More details can be obtained in [Yang and Webb 03].

$P(F_i \in s \mid s \in S)$ is measured by counting the number of sentences appearing in the author-provided summaries from the learning corpora that have the feature F_i divided by the number of all the summary sentences in the corpora. $P(s \in S)$ is calculated as the ratio between the summary sentences and the number of all the sentences in the learning corpora.

One limitation is that the feature independence assumption may be violated in our model. However, Domingos [97] suggests that this limitation has less impact than might be expected and the classification accuracy can remain high even while the probability estimation is poor. Another limitation is that the Naive-Bayes classifier demands discrete values for its features while all the features in our model have continuous values. Thus, we use the Equal Width Interval Binning Discretization method [Dougherty, 95] in order to discrete these values. The values of each feature have been divided into 3 intervals of equal width.

After calculating the probabilities mentioned above, the system presented the sentences that have the highest probability as a summary. The recall rate was lowered by 0.08 (from 0.39 to 0.31). The reasons for this decrease can be the limitations mentioned in the previous paragraph.

5.3 Genetic algorithm

Genetic algorithms (GAs) are search algorithms based on the mechanics of natural selection and natural genetics [Goldberg, 89]. They combine the principle of survival of the fittest among string structures within a structured yet randomized information exchange (crossover and mutation) to form a search algorithm with some of the innovative flair of human search. In every generation, a new set of artificial creatures (strings) is created using bits and pieces of the fitter individuals of the previous generation. While randomized, genetic algorithms are no simple random walk. They efficiently exploit historical information to speculate on new search points with expected improved performance. The main steps of the general GA [Mitchell 97] are:

```
1 Initialize population
2 Evaluate members of the population
3 WHILE "stop criteria" not satisfied DO
  3.1 Selection
  3.2 Crossover
  3.3 Mutation
  3.4 produce a new population
  3.5 Evaluate members of the population
```

In our experiments, we create 5 populations; each contains 200 subjects. The replacement rate of each generation is 70%, and the mutation rate is 1%. At each generation 5 subjects migrated from one population to another. The crossover stage generates new offspring for the next generation by taking some bits from one subject, and some from a second subject (both subjects from the previous generation). The mutation stage changes part of the bits of a certain subject.

We use a library of genetic algorithms [GAlib] as an implementation tool for running our algorithm. After 300 generations the weights of the features achieved the values, presented in Table 3. This Table yields that at least one feature, the length feature is not relevant. That is, using genetic algorithm for our learning corpus discovers that the length of the sentence is not an important feature to determine whether to choose it as a sentence for a summary or not. On the other hand, the domain feature (i.e., the special keywords that classify the domain of the article) appeared to be the most important feature. Using a hybrid method based on the features and their weights from Table 3, raised the recall rate by 0.07 (from 0.39 to 0.46).

Table 3. Weights of features achieved by the genetic algorithm

Feature #	Feature	Weights
1	TF-ISF	0.17

2	Position	0.19
3	Ruling cue	0.15
4	Section title	0.01
5	Domain	0.32
6	Negative	0.03
7	Title	0.10
8	Length	0.00
9	Cue	0.08

Table 4 presents the recall measures of the tested machine learning methods compared to the basic hybrid function. Obviously, the GA achieves the best results (46%). That is, 46% of sentences that appeared in the reference summary have been found by our system. These results appear to be unimpressive. However, we claim that these are rather good results because of the following reasons: (1) This rate is rather reasonable compared to the rate achieved by other summarization systems supplying an equivalent length-rate of summary, e.g.: 42% [Neto et al., 00], and 38% [Neto et al., 00]. (2) These are the results of the first summarization Hebrew system in general and the results of the first summarization system for Jewish law articles in particular. (3) The fact that we have not found the rest of the 54% does not mean that the system chose bad sentences. Rather, sentences proposed by us could be appropriate for summarization although they do not appear in the reference summary. (4) There is evidence that the optimal summary created by extraction is not unique (Rate et al., 61; Chen and Withgott, 92). That is, the reference summaries we compared to are not optimal.

Table 4. The recall measures of the tested machine learning methods

Learning method	**Recall measure**
No learning	0.39
Perceptron	0.4
Naive Bayesian	0.31
genetic algorithm	0.46

6. Summary and future research

In this paper, we have presented several novelties: (1) The first summarization model for Hebrew texts. (2) A special hybrid method for conclusive summarization using sentence extraction. (3) The first sentence extraction model to use successfully genetic algorithm.

General research proposals are: (1) Elaborating the model for summarizing other kinds of Hebrew articles. (2) Investigating other features and other machine learning techniques that might improve summarization. More specific directions for research are: (1) Some rabbinical authorities are taken more seriously by all authors than others. We suggest giving higher rates to sentences where those rabbinical authorities are cited. (2) It is known that certain authors take into consideration some rabbinical authorities rather than others. Therefore the importance of different rabbinical authorities should be computed relatively to the discussed author.

Acknowledgements

Thanks to Ari Cirota and Lavey Y. Freedman for valuable comments on earlier versions of this paper.

References

1. Alterman, R., Text Summarization. In: S. C. Shapiro, (ed.): Encyclopedia of Artificial Intelligence. John Wiley & Sons, New York 1579-1587(1992)
2. Aone C., Okurowski M. E., Gorlinsky J., and Larsen B. A Scalable Summarization System Using Robust NLP. In Proceedings of the ACL Work shop on Intelligent Scalable Text Summarization, pp. 66-73. (1997)
3. Bartlett J., Collection of Familiar Quotations, 15th edition, Citadel Press, 1983.
4. Barzilay R., Elhadad M. Using Lexical Chains for Text Summarization. In Proceedings of the Intelligent Scalable Text Summarization Workshop (ISTS'97), ACL, Madrid, 1997.
5. Chen, F.R. and M.M. Withgott, The use of emphasis to automatically summarize a spoken discourse. In *Proceedings of the IEEE Intl. Conference on Acoustics, Speech, and Signal Processing*, volume 1, pp. 229-232 (1992).
6. Domingos, P., and Pazzani, M. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning 29*, 103–130. (1997)
7. Dougherty, J., Kohavi, R., and Sahami, M. Supervised and unsupervised discretization of continuous features. In *Proceedings of the Twelfth International Conference on Machine Learning*, pp. 194–202. (1995)
8. Edmundson, H.P.: New Methods in Automatic Extraction. Journal of the ACM 16(2): 264-285. (1969)
9. GAlib: A C++ Library of Genetic Algorithm Components, http://lancet.mit.edu/ga.
10. Goldberg, D.E.: Genetic Algorithms in Search Optimization & Machine Learning, Addison-Wesley, (1989).
11. HaCohen-Kerner, Y.: Automatic Extraction of Keywords from Abstracts. In V. Palade, R. J. Howlett, L. C. Jain (Eds.), Proceedings of the Seventh International Conference on Knowledge-Based Intelligent Information & Engineering Systems, Vol. 1, Lecture Notes in Artificial Intelligence 2773, pp. 843-849, Berlin: Springer-Verlag, 2003.
12. HaCohen-Kerner Y., Malin E. and Chasson I., Summarization of Jewish Law Articles in Hebrew. Proceedings of the 16th International

Conference on Computer Applications in Industry and Engineering, Las Vegas, Nevada USA, pp. 172-177, Cary, NC: International Society for Computers and Their Applications (ISCA), 2003.

13. Hovy E.H. and Lin, C-Y.: Automated Text Summarization in SUMMARIST. In Mani and Maybury. *Advances in Automatic Text Summarization*. MIT Press, Cambridge, Massachusetts. (1999)

14. Kupiec J., Pederson J., and Chen F.: A trainable document summarizer. In *Proceedings of the 18th Annual International ACM SIGIR*, pp. 68–73, 1995.

15. Lin, C-Y.: Training a Selection Function for Extraction. In *the 8th International Conference on Information and Knowledge Management* (CIKM 99), Kansa City, Missouri, November 2-6. (1999)

16. Lin, C-Y. and Hovy E.H.: Identifying Topics by Position. In the Proceedings of the Applied Natural Language Processing Conference (ANLP-97), 283-290. Washington. (1997)

17. Luhn, H. P.: The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2):159-165. (1958)

18. Mani, I. and Bloedorn, E.: *Machine Learning of Generic and User-Focused Summarization*. Proceedings of AAAI-98, pp. 821-826. (1998)

19. Mani, I., and Maybury, M. T.: Introduction, Advances in automatic text summarization, Cambridge, MA: MIT Press., ix-xv. (1999)

20. McKeown, K. and Radev, D. R. *Generating summaries of multiple news articles*, pp 74-82, Proceedings of the 18 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, 1995.

21. Mitchell T. M.: Machine Learning, McGraw-Hill, New-York, Second Edition, 1997.

22. Myaeng, S. and Jang, D.: Development and evaluation of a statistically based document summarization system. In Mani and Maybury. *Advances in Automatic Text Summarization*. MIT Press, Cambridge, Massachusetts. (1999)

23. Neto J. L., Santos AD, Kaestner CAA, Freitas AA, and Nievola JC.: A Trainable Algorithm for Summarizing News Stories. In H Zaragoza, P Gallinari, and M Rajman, editors, *Proc. PKDD'2000 Workshop on Machine Learning and Textual Information Access*, Lyon, France, 2000.

24. Neto, J. L., Freitas, A. A. and Kaestner, C. A. A.: Automatic Text Summarization Using a Machine Learning Approach. SBIA: 205-215 (2002)

25. Radev, D. R.: Language Reuse and Regeneration: Generating Natural Language Summaries from Multiple On-Line Sources. PhD thesis, Department of Computer Science, Columbia University, New York (1999).

26. Rath, C.J.,. Resnick, and T.R. Savage, The formation of abstracts by the selection of sentences. American Documentation, 12(2):139-143,(1961)

27. Teufel, S., Moens, M.: Argumentative classification of extracted sentences as a first step towards flexible abstracting. In: I. Mani, M. Maybury (eds.), Advances in automatic Text Summarization, MIT Press, pp. 155-171 (1999)

28. Turney, P.: Learning Algorithms for Keyphrase Extraction, Information Retrieval Journal, Vol.2, No.4, 303-336 (2000)

29. Yang, Y., Webb G. I.: Weighted Proportional k-Interval Discretization for Naive-Bayes Classifiers. In Proceedings of the 7th Pacific-Asia Conference on Knowledge Discovery and Data Mining 501-512 (2003)

30. Zechner, K. A.: Automatic Text Abstracting by Selecting Relevant Passages. M.Sc. dissertation, University of Edinburgh, UK (1995)

31. Zechner, K. A.: Fast Generation of Abstracts from General Domain Text Corpora by Extracting Relevant Sentences. Proceedings of the 16th international Conference on Computational Linguistics, pp. 986-989 (1996)