DRAFT. Accepted to the *Information Sciences, An International Journal* 104 (1998), 31-47

# Entropy of English text: Experiments with humans and a machine learning system based on rough sets

Hamid Moradi
Jerzy W. Grzymala-Busse
James A. Roberts

Department of Electrical Engineering and Computer Science
The University of Kansas
Lawrence, KS 66045

**Abstract**. The goal of this paper is to show the dependency of measured entropy of English text on subject of the experiment, the type of English text, and the methodology used to estimate the entropy.

## 1. Introduction

Entropy of a language is a statistical parameter which measures, in a certain sense, how much information is produced on the average for each letter of a text in a language. When compressing the text, the letters of the text must be translated into binary digits 0 or 1. Entropy H is the average number of binary digits required per letter of the original language to translate each letter in a most efficient way. In his 1951 paper Shannon [1] proposed a new method of estimating the entropy of English text. This method exploits the fact that anyone speaking the language possesses an enormous knowledge of the statistics of the language, and depends on experimental results. In his experiment, Shannon randomly chose 100 passages 15 characters in length (these characters could only be one of 26 letters in English text or a space) from the book *Jefferson the Virginian* and asked his spouse to guess each character of every passage. The subject continued to make a guess for each character until the correct guess was made and then she was asked to guess the next character with the knowledge of the previous characters that she had already guessed. The number of guesses was recorded for each character. Shannon's results show that the prediction gradually improved, apart from some statistical fluctuation, with increasing knowledge of the past.

Using the number of guesses that it took for each letter, Shannon developed a formula for the upper and lower bound entropy for English text. In another experiment, Shannon randomly selected 100 passages 101 characters in length (from the same book) and asked his subject to guess the next letter in each sequence of text, given the previous 100 letters. He then concluded that the long range statistical effects (up to 100 letters) reduce the entropy of English text to the order of one bit per letter.

Many papers relating to Shannon’s paper have been written. Several papers provide important theoretical material. Maixner [2] helps to clarify the details behind the derivation of Shannon's lower bound to Nth-order ((N-1) known letters) entropy approximations. He demonstrates an explicit proof valid for an ideal prediction, adding a negative proof for general experimental conditions. For Shannon’s lower bound entropy formulas to fail, the subject would have to be exceptionally poor predicting in a quite unusual type of language. Still in principle, it indicates potential sources of distortion, though much less prominent in real applications. Savchuk [3] gives necessary and sufficient conditions on the source distribution for Shannon's bound to hold with equality. Background on entropy estimate limitations can be found in [4]-[8].

Several papers comment on Shannon's empirical results for English text. Grignetti [9] recalculates Shannon's estimate of the average entropy of words in English text. Treisman [12] comments on contextual constraints in language. White [13] uses a dictionary encoding technique to achieve compression for printed English and Jamison [14] and Wanas [15] discuss entropy of other languages.

Shannon's or related estimates are used in many wide ranging applications. Miller [16] discusses the effects of meaningful patterns in English text on the entropy calculations. He concludes that when short range contextual dependencies are preserved in nonsense material, the nonsense is as readily recalled as is meaningful material. From his results it is argued that contextual dependencies extending over five or six words permit positive transfer, and that it is these familiar dependencies, rather than the meaning per se, that facilitate learning. In [17], Blackman discusses interference in communication links and its effect on the information that is lost. In [18], Cover estimates the entropy of English text by altering Shannon's technique by having the subjects place sequential bets on the next symbol of text. The amount of money that each subject place on a bet depends on the underlying probability distribution for the process. By using the subject's capital after n bets, Cover estimates that the English text has an entropy of approximately 1.3 bits/symbol, which aggress well with Shannon’s estimate. An important reference work on the subject is the book by Yaglom [19], which contains an extensive bibliography.

Since Shannon’s paper was published in 1951, many have tried to improve on his experiments to more accurately and definitely calculate the entropy of English text. Burton and Licklider [10] took Shannon’s experiment a step further. As the source of test material they used ten novels instead of a single volume (Shannon’s only source was *Jefferson the Virginian*). The ten novels were of about the same level of reading difficulty and abstraction (as measured by Flesch’s scales [30]). From each source they randomly selected 10 passages of each of 10 length (number of letters known to the subjects): 0, 1, 2, 4, 8, 16, 32, 64, 128 and approximately 10000 letters. A different subject was used for each novel. Subjects were given the passages with different lengths and in each instance were asked to guess the next letter. The number of guesses were recorded and Shannon’s formulas were used to calculate the upper and lower bound entropy. Based on the results of their experiment, Burton and Licklider concluded that written English does not become more and more redundant as longer and longer sequences are taken into account. In fact, their results showed that knowledge of more than 32 characters does not necessarily make for better prediction of the next letter. This results seem to dispute Shannon’s conclusion that knowledge of more letters (up to 100 characters) could reduce the entropy of English text.

In this paper we took Shannon’s experiment even a step further than Burton and Licklider suggested. One hundred passages 64 characters in length were chosen from two different books (total of two hundred sequences) with different style of writings (Shannon used one book only and Burton used ten books all with same style of writing). The first book was a technically written book called *Digital Signal Processing* by William D. Stanley and the other book was a novel called

*Scruples II* by Judith Krantz. The only acceptable characters in our sample sequences were the 26 letter English alphabet and a space. All two hundred sequences were chosen randomly. All punctuation marks were ignored. If a sequence contained characters that could not be ignored without the text losing its meanings (for example numerals), then that sequence was thrown away and another one was randomly selected. We chose two subjects (one for each book) for this experiment that were intelligent, educated, and had a good understanding of the English language and its constraints. As in Shannon's first experiment, each subject was asked to guess every letter in each sequence of the text, given the knowledge of previously guessed letters. This is different than Burton and Licklider's experiment were the subjects were given the first 64 letters (in the case of passages that were 64 letters long) and asked to guess the next letter only. The experiment is also different than Shannon's experiment not only because we used two different types of books, but also because Shannon's passages were only 15 letters in length and ours were 64 letters long. Shannon also did an experiment were his passages were one hundred letters long but he did not ask his subject to guess every letter of each sequence, just the last one. The number of guesses made for each letter was then recorded and used in Shannon's formulas to calculate upper bound entropy for each letter of the sequence. Our results for both books showed that the entropy of English text decreases as up to about 31 letters are known to the subjects. Knowledge of more letters did not reduce the entropy in our experiment for either books. This result agrees well with Burton and Licklider's results to further dispute Shannon's conclusion that knowledge of more than 32 letters could reduce the entropy of English text.

In 1964, Paisley [11] published a paper in which he had studied the English text's entropy variation due to authorship, topic, structure, and time of composition. Paisley's allowed character set included the 26 letters of English text, space, and a period. All punctuation marks were coded as periods. Numerals, parenthesis, and special characters were ignored. Paisley's test material included 39 texts of 2528 character samples (totaling 98542 characters) from English translation of 9 Greek texts. These texts covered 3 time periods, 18 authors, and 9 topics. Shannon's guessing scheme was not used in this experiment. Paisley used frequencies of occurrences of all possible characters and character pairs to estimate the single letter probability and conditional probabilities of the letter combinations for every text. With redundancy patterns attributed to variation in four source factors, it was necessary to design a testing scheme in such a way that only one factor is free to vary in each comparison. For example, when testing for the authorship, Paisley used two verse translations of the same book (*Iliad)* (keeping topic and structure constant) by two different authors (Rees and Lattimore) as his test material. The translations were done about the same time period (constant time of composition). Paisley then concluded that letter redundancy covaries with all four factors.

In this paper , Shannon's guessing scheme (not letter probabilities as in Paisley's experiment) was used to study the effects of topics and subjects of the experiment on the entropy of English text. Eight different subjects were used (Shannon only used his wife). Once again, we tried to used educated, intelligent subjects that had a good understanding of English literature and its constraints. In our first experiment we concluded that using Shannon's guessing scheme, entropy of any English text approaches its limits when it is calculated using the number of guesses made for the 32nd character of the sequence with the knowledge of the previous 31 letters (Burton and Licklider [10] also agreed with this conclusion). For this experiment, we took advantage of that conclusion. One hundred sequences of text 32 letters in length were chosen from four different books (total of 400 sequences). The books were selected to have entirely different topics. These books were as follows:

*Scruples II* by Judith Krantz (novel),

*Digital Signal Processing* by William D. Stanley (technically written book),

*101 Dalmatians* by Walt Disney corporation (children book), and

United States Federal Aviation Handbook and Manuals (government document).

As before, the only acceptable characters in our sample sequences were the 26 English alphabet letters and a space. Each subject was given the first 31 characters of every sequence and was asked to guess the 32nd. The subjects were told to continue to guess until the correct guess was made. The number of guesses were recorded for each sequence and used in Shannon's formulas to calculate upper bound entropy for all four books (a total of 32 entropies were calculated, 8 subjects x 4 books). Entropy for the technically written book (*Digital Signal Processing* ) was calculated to be the lowest and for the children book (*101 Dalmatians* ) to be the second lowest in 7 out of 8 cases (88%) (in each case the subject of the experiment is the same). These results indicated a definite dependency of the entropy of English text on the type of text used. The results from the other books were not very conclusive. In 5 out of 8 cases (63%) the entropy was calculated to be the 3rd lowest for the children book and in 4 out of 8 cases (50%) the government document resulted in the largest entropy. To study the effects of the subjects of the experiment on the entropy estimates, it is only necessary to compare the entropy calculated for any particular book (keeping the topic constant) for all 8 subjects. The lowest entropy calculated for the *Digital Signal Processing* book was 1.62 bits per letter. The highest entropy calculated for this book was 2.28 bits per letter. This is a discrepancy of about 28%. It is very apparent that Shannon's experiment (using a human subject to guess the next letter in a sequence) is definitely subject dependent. The knowledge base along with the mind set of the subject determine how he makes his guesses which in turn determine the entropy calculations. From this experiment we concluded that indeed the entropy of English text covaries with both subjects of Shannon's experiment and the type of text used.

Shannon also envisioned a communication system that will use an identical twin pair to achieve text compression. He suggested that text compression can be achieved by having identical twins serve as the coder and decoder at both ends of the communication channel. The first twin will guess the letters of the text to be transmitted, one by one. The number of guesses that it takes for the first twin to guess each letter correctly is then coded and transmitted instead of the letter itself. Ideally the twin would guess the letters in the order of decreasing conditional probabilities to insure the best possible guess for each letter. Each letter is guessed correctly with a minimum of one guess and maximum of 27 guesses (27 letter alphabet). Thus reducing the coding process to mapping of letters into numbers from 1 to 27. This mapping has been such that the symbol 1 now has extremely high frequency (most letters would be guessed correctly on the first try, if conditional probabilities are used). The symbols 2, 3, 4,..., 27 have successively smaller frequencies. It could be said that the probabilities of various symbols have collapsed to a small group and therefore reducing the number of bits per letter required to translate each letter into binary digits. These new symbols, 1 to 27, are called the reduced text. At the receiver, the identical twin receives the reduced text and will be able to correctly decode the letter because he knows what letter to guess. The logic at the receiver is simple. Suppose the number that the twin at the receiver receives is 4. Given the known text up to the unknown letter, the twin will simply guess the next letter in decreasing conditional probability 4 times. The fourth guess will be the letter that was originally transmitted. However, this system is practically impossible to build using human beings as the coder and decoder since there are no two human beings that think exactly alike.

Based on our research, nobody has ever build Shannon's communication system using an AI system as the subject of Shannon's experiment. In this paper the idea is explored and experimented with, using LERS learning system. In [21], Grzymala-Busse explains the LERS learning system in detail. This system uses the rough set theory that is explained in detail in [24] and [25]. LERS can be trained by the use of some examples, to make educated guesses on the next letter in the sequence of letters. Obviously, the more examples the system can see, the better trained it will be and the better guesses it will make. Papers [22] and [23] are on uncertainty in expert systems. Shannon's proposed communication system can practically be built to achieve text compression using LERS as the identical twins at both ends of the communication channel.

The first two experiments in my research were done for theoretical reasons. We were able to change Shannon's experiment somewhat to test the long range constraints of the English text (first experiment). We were also able to study the effects of the type of text and the subjects used in Shannon's experiments on the entropy calculations of the English text (second experiment). The next two experiments were designed to use LERS as the subject of Shannon's experiment.

In our first experiment, we suggested (Burton and Licklider also did) that if 31 characters are known to the predictor, the number of guesses for the 32nd character (using Shannon's guessing scheme) could be used in Shannon's formulas to calculate an accurate estimate of the entropy of English text. This means that if identical twins could be found for the coder and the decoder and they had complete knowledge of the conditional probabilities, then we could achieve maximum compression of the English text using Shannon's communication system. To use LERS as the subject of Shannon's experiment means that we would have to train the system to be able to guess the 32nd character of a sequence, given the first 31 characters. there are roughly $27^{32}$ ($3^{96}$) possible ways of arranging 32 characters in a sequence. For LERS to make perfect guesses, about $27^{32}$ examples of sequences 32 characters in length have to be used for training purposes. Obviously this is impossible to do. To achieve good text compression, it is not necessary for LERS to make perfect guesses every time. Good compression rate can be achieved if LERS makes good guesses on the next letter in a sequence. In this experiment, we used 10592 examples of sequences 32 characters in length (from the novel *Scruples II* by Judith Krantz) to train LERS learning system. This is a very small fraction of all possible sequences with length of 32 characters. Randomly selected sequences of text (other than those used for training purposes), 32 characters in length, were selected from the same book for testing purposes. LERS was not able to make good guesses in any of test cases. This was not very surprising since we did not use a large percentage of all possible sequences with 32 character lengths to train the system.

Next, we trained LERS with 86,004 examples of 4 characters each from the novel *Scruples II* by Judith Krantz. LERS will use the first three characters of each sequence as known letters and the fourth letter as the decision letter to learn from. One hundred sequences of text, four characters long each, were randomly selected (other than the examples used to train the system with) from the same novel for testing purposes. The number of guesses made by LERS for the fourth character of each sequence were recorded and used in Shannon's formulas to calculate the estimated entropy of the novel. This entropy was calculated to be 2.46 bits/letter. This is better than the entropy calculated from the data obtained with a human as subject of Shannon's experiment (2.62 bits/letter) for the same experiment.

Our research indicates that Shannon's communication system could be built using LERS as the prediction tool to achieve good compression for English text. This can be achieved because LERS can be used both as the coder and the decoder, thus eliminating the identical twin problem. Given more data to learn from, LERS could make even better guesses and therefore achieve even better compression rates. As computer's processing speed improves, it becomes more realistic to train LERS with enough examples to achieve maximum compression rate.

## Section 2

Entropy is a measure of information. Entropy of English test is the average number of bits per letter of the text that will be required to translate the language into binary bits. From the entropy, the redundancy of English text can be calculated. The lower the entropy of the English text is, the more redundant it is. "The redundancy measures the amount of constraint imposed on the English text due to its statistical structure" [1]. For example, space is the most probable character in English text and Q is always followed by a U. In the following section of this section, the notations and the formulas used in calculating the entropy of English text from the statistics of the language is explained. Section II covers a description of an ideal predictor as introduced by Shannon [1]. In section III, Shannon's entropy bounds calculation for an ideal predictor is summarized.

### *I. ENTROPY OF ENGLISH TEXT FROM THE STATISTICS OF ENGLISH*

Calculating the entropy of English text from the statistics of English can be done by a series of approximations $F_0$, $F_1$, $F_2$, ..., $F_N$. $F_N$ is called the N-gram entropy and is a "measure of amount of information due to statistics extending over N adjacent letters of text" [1]. As N approaches infinity, $F_N$ approaches H, the entropy of English text. $F_N$ is given by [1]

$$F_n = -\sum_{i,j} p(b_i, j)\log_2(p_{b_i}(j)) \qquad (1)$$

$$= -\sum_{i,j} p(b_i, j)\log_2 p(b_i, j) + \sum_i p(b_i)\log p(b_i)$$

where $b_i$ is a block of N-1 letters (N-1) gram; j is an arbitrary letter following $b_i$; $p(b_i ,j)$ is the probability of the N-gram $b_i$ ; j, $p_{b_i}(j)$ is the conditional probability of letter j after the block $b_i$ and is given by $p(b_i,j)/p(b_i)$.

As N is increased, $F_N$ includes longer and longer statistics and approaches H, the entropy of English text, for N infinite.

$$H = \mathrm{LIM}_{N \to \infty} F_N. \qquad (2)$$

$F_N$ can be calculated for small values of N (1, 2, 3) from the standard tables of letters, digrams, and trigrams [29]. For a 27 letter alphabet (26 letters and a space), $F_0$ is, by definition,

$$F_0 = \log_2 27 = 4.76 \text{ bits per letter}$$

$F_1$ can be calculated by using letter probabilities as follows :

$$F_1 = -\sum_{i=1}^{27} p(i)\log_2 p(i) = 4.03 \qquad (3)$$

bits per letter.

The digram approximation F2 is given in [1] by

$$F_2 = -\sum_{i,j} p(i,j)\log_2 p_i(j) \qquad (4)$$

$$= -\sum_{i,j} p(i,j)\log_2 p(i,j) + \sum_i p(i)\log_2 p(i) = 3.32$$

bits per letter. The trigram entropy F3 is given by [1]

$$F_3 = -\sum_{i,j,k} p(i,j,k)\log_2 p_{i,j}(k) \qquad (5)$$

$$= -\sum_{i,j,k} p(i,j,k)\log_2 p(i,j,k) + \sum_{i,j} \log_2 p(i,j) = 3.1$$

bits per letter. Higher order approximations ($F_4$, $F_5$, ..., $F_N$) are not calculated this way since tables of N-gram frequencies are not available. However, word frequencies have been tabulated [20] and can be used for further approximation. Shannon demonstrates this approximation in [1] and calculates an entropy of 11.82 bits per word for English text. This is also equal to 11.82 / 5.5 = 2.14 bits per letter since an average word has 5.5 letters (includes a space). However, this is not the same as $F_5$ or $F_6$. $F_5$ and $F_6$ correspond to a combination of any 5 or 6 letters. They do not have to form a word. Shannon also carried out similar calculations for a 26 letter alphabet (no space). The results for both calculations are summarized bellow:

| | $F_0$ | $F_1$ | $F_2$ | $F_3$ | $F_{word}$ |
|---|---|---|---|---|---|
| 27 letter --------> | 4.76 | 4.03 | 3.32 | 3.1 | 2.14 |
| 26 letter --------> | 4.70 | 4.14 | 3.56 | 3.3 | 2.62 |

"Since the space symbol is almost completely redundant when sequences of one or more words are involved, the value of $F_N$ in a 27-letter case will be 4.5/5.5 or 0.818 of $F_N$ for the 26 letter alphabet when N is reasonably large" [1].

## *II. IDEAL PREDICTOR*

lets define the parameter $p_{i1,i2,...,i(N-1)}(j)$ as the conditional probability of letter j following the $i1,i2,...,i(N-1)$ block of letters. The best guess for the next letter given that block of N-1 letters are the preceding letters, would be the letter with the highest conditional probability. The second best guess would be the letter with the second highest conditional probability. An ideal predictor would guess the letters in the order of decreasing conditional probabilities to insure the best possible guess for each letter. Each letter is guessed correctly with a minimum of one guess and maximum of 27 guesses (27 letter alphabet). Thus using an ideal predictor will reduce the coding process to mapping of letters into numbers from 1 to 27. This is done by mapping the most probable letter (based on conditional probability) into 1, the second most probable letter into 2, etc. The frequency of 1's in the reduce text will then be given by [1]

$$q_1^N = \sum p(i_1, i_2, \ldots i_{N-1}, j) \tag{6}$$

"where the sum is taken over all (N-1) grams $i_1, i_2, \ldots, i_{N-1}$ and j is the one which maximizes p for that particular (N-1) gram. Similarly, the frequency of 2's, $q_1^N$, is given by the same formula with j chosen to be that letter having the second highest probability, p" [1]. For N preceding letters known, $q_1^{N+1}$, $q_2^{N+1}$, ..., $q_{27}^{N+1}$ are the probabilities of the new symbols 1, 2, ..., 27.
In [1], Shannon proves the inequality

$$\sum_{i=1}^{s} q_i^{N+1} \geq \sum_{i=1}^{s} q_i^{N} S = 1,2... \tag{7}$$

"which means that probability of being right in the first S guesses when the preceding N letters are known is greater than or equal to that when only (N-1) letters are known" [1].

It can be said that the ideal predictor is a transducer which translates the language into a sequence of numbers running from 1 to 27. It could also be said that ideal prediction collapses the probabilities of various symbols to a small group and therefore reduces the number of bits per letter required to translate each letter into binary digits. These new symbols, 1 to 27, are called the reduced text.

The reduced text contains the same information as the original text. When transmitting this information over a communication channel, only the reduced text is transmitted, reducing the number of bits required to transmit the information. This communication system is shown bellow in Figure 1 [1].

**Figure 1. Communication System Using Reduced Text.**

This is how it works. Suppose letter 'A' is to be transmitted. The preceding letters to this letter are sent to the predictor and the predictor will make a guess on the letter. If 'A' is the first letter, then no preceding letters are sent to the predictor. The predictors guess and the actual text (letter 'A') are sent to the comparator. If the guess was incorrect, then the predictor is asked to guess again. This is continued until the predictor guesses the letter correctly. Suppose the predictor guessed the letter 'A' on the third try. The reduced text is now number 3. Number 3 is transmitted over the channel to the receiver. The first stage on the receiver side is a comparator. The receiver must also employ the exact same predictor as the transmitter. The predictor at the receiver will also make guesses on the letter with the knowledge of the preceding letters. The predictor will continue guessing the letter until it is told to guess the next letter. At the receiver, the comparators function is to count the number of guesses the predictor makes. The third guess of the predictor will be letter 'A', the original text.

## *III. SHANNON'S ENTROPY BOUNDS*

An upper and a lower bound entropy can be calculated with the knowledge of the symbol frequencies from the ideal predictor. It is possible to set both these bounds to the N-gram entropy, $F_N$, of the original text. In [1], Shannon defines these bounds as follows :

$$\sum_{i=1}^{27} i(q_i^N - q_i^N)\log i \leq F_N \leq -\sum_{i=1}^{27} q_i^N \log q_i^N \qquad (8)$$

the proof for the lower bound is offered in [1] and is not repeated here. The upper bound follows from the fact that the maximum possible entropy in a language with letter frequencies $q_i^N$ is

$$-\sum q_i^N \log q_i^N$$

This means that the entropy of the language can not be greater than this. " The N-gram entropy of the reduced text is equal to that for the original language, as may be seen by inspection from (1) of $F_N$. The sums involved will contain precisely the same terms although, perhaps, in a different order" [1]. The upper bound entropy is the entropy calculated for all the experiments that were done in this paper .

## Section 3

In a 1950 paper [1], Shannon proposed a new method in estimating the entropy of English text. This method takes advantages of the fact that anyone who speaks the language has a clear understanding of the statistics of the language. This person will be familiar with the grammar, letter probabilities, tendencies of some letters following others, and words in the language. This enables him to make educated guesses on the missing letter of a sequence of text with the knowledge of preceding letters.

One of the experiments that Shannon completed was to randomly select 100 sentences from the book " *Jefferson the Virginian*" by Dumas Malone, each fifteen characters in length, and ask a subject to guess all the characters of each sentence one at a time with the preceding letters known to him.

The first experiment carried out in this paper was an extension of Shannon's experiment. This experiment is explained in the following section. In section II of this section, the results of this experiment are analyzed. Section III covers the same experiment with passages from a different type of text and its results.

### *I. EXPERIMENT (I_A)*

In this experiment, one hundred samples of text, each 64 character long, were selected from the book Scruples II by Judith Krantz. These samples were all unfamiliar to the subjects. The subjects were asked to guess the letters of each sample starting with the first letter. If he guesses correctly, then he is asked to guess the next letter in the sequence with the knowledge of the previous letters. If the guess is incorrect, he is told so and asked to guess again. This is continued until all 64 letters of each sample are guessed correctly. Meanwhile, the number of guesses for each letter is kept in a data base and updated with each new sample. Thus one hundred samples were obtained for which the subjects had 0,1,2,3,4, ..., 63 preceding letters. These results are tabulated in table I_a in the appendix and are partially shown in table I on the next page.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 32 | 64 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 18.2 | 29.2 | 38.0 | 56.0 | 43.0 | 56.0 | 61.0 | 68.0 | 61.0 |
| 2 | 10.7 | 14.8 | 16.0 | 11.0 | 11.0 | 11.0 | 11.0 | 9.0 | 10.0 |
| 3 | 8.6 | 10.0 | 9.0 | 2.0 | 7.0 | 9.0 | 5.0 | 2.0 | 4.0 |
| 4 | 6.7 | 8.6 | 8.0 | 4.0 | 5.0 | 3.0 | 4.0 | 3.0 | 4.0 |
| 5 | 6.5 | 7.1 | 9.0 | 2.0 | 4.0 | 1.0 | 4.0 | 3.0 | 2.0 |
| 6 | 5.8 | 5.5 | 3.0 | 3.0 | 0 | 1.0 | 1.0 | 2.0 | 2.0 |
| 7 | 5.6 | 4.5 | 3.0 | 3.0 | 2.0 | 4.0 | 2.0 | 1.0 | 2.0 |
| 8 | 5.2 | 3.6 | 1.0 | 3.0 | 0 | 2.0 | 2.0 | 1.0 | 3.0 |
| 9 | 5.0 | 3.0 | 0 | 1.0 | 9.0 | 1.0 | 2.0 | 1.0 | 1.0 |
| 10 | 4.3 | 2.6 | 3.0 | 5.0 | 2.0 | 0 | 2.0 | 0 | 0 |
| 11 | 3.1 | 2.2 | 0 | 1.0 | 2.0 | 0 | 0 | 1.0 | 3.0 |
| 12 | 2.8 | 1.9 | 0 | 2.0 | 2.0 | 2.0 | 1.0 | 2.0 | 2.0 |
| 13 | 2.4 | 1.5 | 1.0 | 0 | 1.0 | 0 | 0 | 0 | 0 |
| 14 | 2.3 | 1.2 | 2.0 | 2.0 | 0 | 1.0 | 1.0 | 1.0 | 1.0 |
| 15 | 2.1 | 1.0 | 1.0 | 2.0 | 1.0 | 0 | 1.0 | 1.0 | 0 |
| 16 | 2.0 | 0.9 | 1.0 | 0 | 2.0 | 2.0 | 0 | 1.0 | 1.0 |
| 17 | 1.6 | 0.7 | 1.0 | 0 | 1.0 | 0 | 0 | 0 | 0 |
| 18 | 1.6 | 0.5 | 1.0 | 0 | 3.0 | 2.0 | 1.0 | 1.0 | 0 |
| 19 | 1.6 | 0.4 | 0 | 2.0 | 1.0 | 0 | 1.0 | 0 | 1.0 |
| 20 | 1.3 | 0.3 | 0 | 0 | 1.0 | 1.0 | 1.0 | 2.0 | 0 |
| 21 | 1.2 | 0.2 | 0 | 0 | 0 | 1.0 | 0 | 0 | 1.0 |
| 22 | 0.8 | 0.1 | 1.0 | 0 | 1.0 | 0 | 0 | 0 | 1.0 |
| 23 | 0.3 | 0.1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 24 | 0.1 | 0.0 | 1.0 | 1.0 | 0 | 0 | 0 | 0 | 0 |
| 25 | 0.1 | 0.0 | 0 | 0 | 1.0 | 0 | 0 | 0 | 0 |
| 26 | 0.1 | 0.0 | 0 | 0 | 1.0 | 1.0 | 0 | 0 | 0 |
| 27 | 0.1 | 0.0 | 1.0 | 0 | 0 | 2.0 | 0 | 1.0 | 1.0 |

Table I

The column corresponds to the number of preceding letters known to the subject plus one; the row is the number of guesses. "The entry in column N at row S is the number of times the subject guessed the right letter at the Sth guess when (N-1) letters were known" [1]. For example, the entry 11 in column 7, row 2, means that with six letters known the correct letter was guessed on the second try eleven times out of a hundred.

" The first two columns of this table were not obtained by the experimental results but were calculated directly from the known letter and digram frequencies. Thus with no letters known, the most probable symbol is the space (18.2 times out of a hundred); the next guess, if this is wrong, should be E (10.7 times out of hundred) " [1].

The 27 letter symbols are mapped into a new set of symbols, numbers 1 to 27. The entries in each column of table I represent the conditional probabilities for the new symbols. For example, number 56 in row 1 column 4 (1,4) represents the number of times the fourth letter (given 3 preceding letters) was the most probable choice. This results in a probability of 0.56 for the number 1 which corresponds to that letter being guessed correctly on the first try. The probability of the fourth letter being guessed correctly on the second try is therefore 0.11 and so on.

## *II. RESULTS FOR EXPERIMENT (I_A)*

### A) STATISTICAL ERRORS IN THE RESULTS

The data entries in table I are subject to experimental errors. Only one hundred samples were examined. To be statistically correct, hundreds or thousands more are needed which requires extremely long hours to complete the experiment. However, we can manipulate the data to reduce the percentage of error in the entropy calculations. The entries in each column must be in decreasing order since they are the conditional probabilities for the text (i.e. number 1 is mapped into a most probable letter of English text and number 2 is mapped into the second most probable letter and so on). Also the small numbers in the table are more sensitive to variation than the larger numbers. For this experiment, we will consider the numbers equal or bellow 5 as small. Going down in each column, starting with the entries that do not follow the decreasing order or/and are small, we will use a uniformization process to reduce the percentage error. for example, in column

4, entry 2 in the 3rd row is the first small number. Starting with this entry, the remaining entries in this column are summed and divided uniformly (in a decreasing order) between rows 3 to 24 (the last non zero entry in that column). The entire table is rearranged in this order and is shown in the appendix as Table I_b.

## B) UPPER BOUND ENTROPY

From the conditional probabilities for English text given in table I_B in the appendix (each table entry is divided by 100 to obtain the probabilities), the upper and lower bound entropies can be calculated for the Scruples novel using equation (8). Some of these calculated entropies are tabulated bellow.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | … | 31 | 32 | 33 | … | 64 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Upper Bound Entropy | 4.07 | 3.43 | 2.71 | 2.62 | 3.46 | 2.75 | 2.73 | | 2.25 | 2.70 | 1.69 | | 1.72 |
| Lower Bound Entropy | 3.14 | 2.49 | 1.68 | 1.59 | 2.44 | 1.71 | 1.67 | | 1.34 | 0.88 | 0.87 | | 0.94 |

The entropy versus the number of letters known are plotted in figure (2) bellow. Some of these numbers are higher and some are lower than the entropies obtained from the experiment done by Shannon. The fluctuation in these numbers from Shannon's numbers could be blamed on the differences in types of text or subject from those of Shannon's. As shown in figure (2), the entropy of English text levels off at about 32 characters to about 2.05 bits per letter. This observation suggests that written English does not become more and more redundant as longer sequences are taken into effect. This result also agrees with the results found by Burton and Licklider [11]. Burton and Licklider also repeated some of Shannon's experiments with samples of 0, 1, 2, 4, 8, 16, 32, 64, 128, and approximately 1000 letters known to the subjects and asked the subjects to guess the next character in each passages. They concluded that the redundancy of the text does not increase noticeably when the subjects were given 1000 known letters versus 32 known letters.

### *III) EXPERIMENT (I_B) AND THE RESULTS*

In this experiment, a different type of text was used. The passages were taken from the book *Digital Signal Processing* (a technically written book) by William D. Stanley. Also, a different subject, one who was familiar with the terminology used in a technical book, was used. The same exact steps and procedures as experiment I were followed. This data was also manipulated to reduce the percentage of error due to experimental inaccuracy by following the steps explained in the previous experiment. The original data obtained from this experiment and the improved data are respectively shown in table IA and table II_b in the appendix. The upper bound entropy versus the number of letters known is plotted in figure (3) bellow. The upper bound entropy calculated for this experiment also levels out after about 32 characters known. This is consistent with the results found in experiment I_A and by Burton and Licklider. We can safely conclude that regardless of the type of text or of the subject of the experiment, the entropy of English text does not decrease more and more as the sequence gets longer and longer. This is also the conclusion that Burton and Licklider reached in their research [11].

## Section 4

Shannon's experiment relies on the idea that anyone that speaks the language has a very good understanding of the statistics of the language and its grammar. The subject should be able to make the best possible choice for the next letter in the sequence, given the previous letters. It would only make sense that the type of text being used would make a difference in the entropy calculation. We anticipate that if a text follows good grammar skills, has a small vocabulary, and/or uses a collection of very frequently used words, it is more likely to be predictable. Obviously, the more predictable the letters of a text are, the easier it will be for the subjects of the experiment to make the best probable guess for those letters and the entropy will be lower. The next section of this section describes an experiment that was designed to compare the entropy of different types of English text. In section II, the results for this experiment are discussed.

## I) EXPERIMENT (II) ENTROPY FOR DIFFERENT TYPES OF TEXT

The passages used for this experiment were taken from four different types of books. The four books used are as follows:

Digital Signal Processing by William D. Stanley,

Scruples II by Judith Krantz,

101 Dalmatians by Walt Disney corporation, and

United States Federal Aviation Handbook and Manuals.

The books are completely different in contents. The first book listed above is a technically written text book in electrical engineering, the second book is a novel, the third book is a very easy to read kids book, and the fourth book mentioned above is a government document.

In experiments I_a and I_b we showed that the entropy of English text, using Shannon's experiment, levels off after 32 characters. For this experiment, one hundred samples of text, each 32 characters in length were taken from the four books (there were one hundred passages from each book). Eight different subjects (4 were male and 4 were female) were used for this experiment. The subjects were given the first 31 letters of each passage from each book and were asked to guess the 32nd letter.

## *II) RESULTS FOR EXPERIMENT (II)*

The data obtained from each subject for all four books were manipulated (using the procedure explained in section (3) to reduce the percentage error due to inconsistent data. Equation (8) in section (2) (Shannon's equation) was used to calculate the entropy for all books and for all eight subjects. These results are shown in figures (4) and (5) bellow. Figure (4) is the results obtained for the male subjects and figure (5) is the results obtained for the female subjects. Interesting observations can be made.

The data obtained from all four male subjects resulted in the lowest entropy for the Digital Signal Processing book with the entropy ranging from 1.65 to 2.27 and the highest entropy for the government document with entropy ranging from 2.5 to 3.0. The kids book was the book with the second lowest entropy and the novel was the book with the third lowest entropy in all four cases. Based on the data obtained from the male subjects, there is definitely a correlation between the types of the English text and its entropy.

The data obtained from the female subjects were not as conclusive as the data obtained from the male subjects. Entropy for the digital signal processing book was calculated to be the lowest and for the kids book to be the second lowest in three of four cases. This data does not result in the government document having the highest entropy for any of the four female subjects as oppose to the data from the male subjects that resulted in the government document having the highest entropy for all four cases. This somewhat invalidates our previous conclusion. However, the entropies calculated are still very low (1.6 to 2.95). If we could find a prediction method that is not so subject dependent and could make as good prediction as the human subjects, we could be on to something. This leads us into the next section where we use computers as the subjects for Shannon's experiments.

## Section 5

Suppose that a subject was asked to guess the sequence of text that is to be transmitted, character by character. The number of guesses for each character is then coded and transmitted instead of the character itself. To decode this information at the receiver end, an identical twin of the individual who produced the numbers is needed. Given the number of guesses for each character, the twin will be able to guess correctly what was transmitted.

In principal, it is possible to recover the original text at the receiver using an identical twin. In reality, it is impossible to find such identical twin. To be able to encode the text using Shannon's guessing scheme and actually recover the same text at the receiver, a method must be developed to make educated guesses at the transmitter that can be duplicated at the receiver. Such method has not yet been developed for long sequence of English characters. In the following section a machine learning system (LERS) is briefly explained. This system can make educated guesses for the unknown characters given enough input data for training purposes.

### *I) USING LERS- A SYSTEM FOR LEARNING FROM EXAMPLES BASED ON ROUGH SETS*

LERS is a machine learning system that was developed at the University of Kansas. This system can be trained to guess the next character in a sequence of characters. The system handles inconsistency in the input data due to its usage of rough set theory principle [24] [25]. To do so, a set of attributes, decisions, and examples must be defined. Attributes are the known sequence of characters, the decision is the next character of the sequence that is used to train the system, and examples are a combination of known characters and decisions. An example of input data used to train the system is shown in table (2) bellow.

| | | | | |
|---|---|---|---|---|
| <a | a | a | a | d> |
| [Temperature | Hemoglobin | Blood_Pressure | Oxygen_Pressure | Comfort] |
| low | fair | low | fair | low |
| low | fair | normal | poor | low |
| normal | good | low | good | low |
| normal | good | low | good | medium |
| low | good | normal | good | medium |
| low | good | normal | fair | medium |
| normal | fair | normal | good | medium |
| normal | poor | high | good | very_low |
| high | good | high | | |

Table 2

The symbols 'a' and 'd' in the first line of the input file represent attribute and decision. There are four attributes and one decision for this example. The second line contains the names for the attributes and the decision. Lines 3 through 9 of the table represent the examples that are used for training purposes. "In table 1, decision Comfort has three values: low, medium, and very_low. Each such value represents a concept" [21]. Given these examples, the system checks for consistency and induces a set of rules for the concepts. These rules are then used to make educated guesses for the decision given another set of attributes.

For the decision table presented in table 2, the set of all induced rules is presented bellow. Certain rules are [21]

(Oxygen_Saturation, poor) --> (Comfort, low)
(Temperature, low) & (Hemoglobin, fair) --> (Comfort, low)
(Temperature, low) & (Blood_Pressure, low) --> (Comfort, low)
(Hemoglobin, fair) & (Blood_Pressure, low) --> (Comfort, low)
(Hemoglobin, fair) & (Oxygen_Saturation, fair) --> (Comfort, low)
(Blood_Pressure, low) & (Oxygen_Saturation, fair) --> (Comfort, low)
(Temperature, low) & (Hemoglobin, good) --> (Comfort, medium)
(Temperature, normal) & (Hemoglobin, fair) --> (Comfort, medium)
(Temperature, normal) & (Blood_Pressure, normal) --> (Comfort, medium)
(Temperature, low) & (Oxygen_Saturation, good) --> (Comfort, medium)
(Hemoglobin, good) & (Blood_Pressure, normal) --> (Comfort, medium)
(Hemoglobin, fair) & (Oxygen_Saturation, good) --> (Comfort, medium)
(Blood_Pressure, normal)&(Oxygen_Saturation, good) --> (Comfort, medium)
(Blood_Pressure, normal)&(Oxygen_Saturation, fair) --> (Comfort, medium)
(Temperature, high) --> (Comfort, very_low)
(Hemoglobin, poor) --> (Comfort, very_low)
(Blood_Pressure, high) --> (Comfort, very_low)

and possible rules are

(Blood_Pressure, low) --> (Comfort, low)
(Oxygen_Saturation, poor) --> (Comfort, low)

(Temperature, low) & (Hemoglobin, fair) --> (Comfort, low)

(Temperature, normal) & (Hemoglobin, good) --> (Comfort, low)

(Hemoglobin, fair) & (Oxygen_Saturation, fair) --> (Comfort, low)

(Temperature, normal) & (Hemoglobin, good) --> (Comfort, medium)

(Temperature, low) & (Hemoglobin, good) --> (Comfort, medium)

(Temperature, normal) & (Hemoglobin, fair) --> (Comfort, medium)

(Temperature, normal) & (Blood_Pressure, low) - -> (Comfort, medium)

(Temperature, normal) & (Blood_Pressure, normal) --> (Comfort, medium)

(Temperature, low) & (Oxygen_Saturation, good) --> (Comfort, medium)

(Hemoglobin, good) & (Blood_Pressure, low) --> (Comfort, medium)

(Hemoglobin, good) & (Blood_Pressure, normal) --> (Comfort, medium)

(Hemoglobin, good) & (Oxygen_Saturation, good) --> (Comfort, medium)

(Hemoglobin, fair) & (Oxygen_Saturation, good) --> (Comfort, medium)

(Blood_Pressure, low) & (Oxygen_Saturation, good) --> (Comfort, medium)

(Blood_Pressure, normal)&(Oxygen_Saturation, good) --> (Comfort, medium)

(Blood_Pressure, normal)&(Oxygen_Saturation, fair) --> (Comfort, medium)

(Temperature, high) --> (Comfort, very_low)

(Hemoglobin, poor) --> (Comfort, very_low)

(Blood_Pressure, high) --> (Comfort, very_low)

After the system has been trained, new attributes given to the system will be checked for possible matches with the rules. If enough input data has been used for training, the system will find one or more matches with the rules and one or more possible decisions. These decisions are then given a number that represents the strength of that decision.

## *II) EXPERIMENTING WITH LERS*

As was mentioned earlier, using Shannon's experiment, the entropy of English text approaches its limit after the 32nd character has been guessed. Therefore the entropy of English text can be approximated by calculating the entropy of the 32nd character. One of the experiments explained earlier was asking the subjects to guess the 32nd character given the previous 31 characters. The data collected from that experiment was used to calculate an approximation for entropy of English text for different types of books. We tried the same experiment with LERS.

### (II_a) EXPERIMENT (III)

The system was trained with 10592 examples (about 158 pages of the novel SCRUPLES) of 32 characters each. The first 31 characters were used as known letters (attributes) and the 32nd was used as the decision. LERS was run on the DEC 5000/2000 computers and took over forty one hours (real time) to be trained. Once completed, the system returned 5043 rules with average of 3.66 conditions per rule. A sample of these rules is given in the appendix. For testing purposes, strings of 31 characters were randomly selected from the book and inputted as attributes. LERS was not able to find a match with its rules set and therefore no decision was given. There were just not enough input data for training purposes. For all practical purposes, it is impossible to provide enough input data for this system to be trained for reasonable accuracy with 31 attributes. The next step is to train LERS with smaller window size (attributes) and see how accurate it can be in predicting the letters of an English text.

### (II_b) EXPERIMENT (IV)

In this experiment, three characters were used as know letters (number of attributes) and the fourth was used as the decision. The system was trained with 36833 examples of 4 characters each and returned 12624 rules. One hundred character strings of length 4 were randomly selected, from the same novel, for testing purposes. Bellow is a sample of the results obtained from LERS.

### (II_c) RESULTS FOR EXPERIMENT (IV)

Given the sequence "LDER" with "LDE" being known to LERS and "R" the character that it is tying to guess, LERS came up with the following results:

| possible solution | | strength |
|---|---|---|
| R | -------> | 21 |
| N | -------> | 3 |
| D | -------> | 3 |

The first column is the possible solutions that LERS came up with and the second column is a strength assigned to that solution. The character with the highest strength is the first guess, the character with the second highest is the second guess and so on. This was an example where LERS was able to guess the correct character on the first try. Lets do an example where LERS makes multiple guesses before the correct character can be identified.

Given the sequence " BRA" with " BR" being known to the system and "A" the decision character, LERS produced the following output.

| possible solution | strength |
|---|---|
| E | 60 |
| O | 51 |
| I | 36 |
| A | 15 |
| U | 12 |

Character "A" has the fourth highest strength and therefore four guesses have to be made by LERS before it can produce the next letter in the sequence correctly.

There were 5 cases (out of possible 100 examples) where LERS was not able to produce a list of possible characters that included the correct character. Here is an example.

Given the sequence "SHA " with "SHA" being known and " " (space) being the next character in the sequence, LERS produced the following possibilities.

| possible solution | strength |
|---|---|
| R | 15 |
| L | 9 |
| P | 3 |
| K | 3 |
| V | 0 |

None of the possible characters above is the character that we are looking for (space). In this case, the predictor will be instructed to follow straight probabilities to guess the next letter in the sequence. The most probable character in English text is the space " " which happens to be the next character in the sequence.

The results for all one hundred examples were totaled and tabulated in table (3) bellow:

| # Guesses | # of times out of 100 | # guesses | # of times out of 100 | # guesses | # of times out of 100 | # guesses | # of times out of 100 |
|---|---|---|---|---|---|---|---|
| 1 | 51 | 8 | 2 | 15 | 0 | 22 | 0 |
| 2 | 16 | 9 | 0 | 16 | 1 | 23 | 1 |
| 3 | 9 | 10 | 1 | 17 | 1 | 24 | 0 |
| 4 | 4 | 11 | 0 | 18 | 0 | 25 | 1 |
| 5 | 7 | 12 | 0 | 19 | 0 | 26 | 0 |
| 6 | 1 | 13 | 3 | 20 | 0 | 27 | 0 |
| 7 | 2 | 14 | 0 | 21 | 0 | | |

Table 3

Using the data above and Shannon's upper bound entropy formulas given in section 2, the entropy for the novel was calculated to be

$H(x) = 2.46$ bits per letter.

This means that if we use Shannon's communication system and LERS as the predictor, we should be able to code, transmit, and decode the contents of the SCRUPLES novel with an average of 2.46 bits per letter.

Increasing the size of the window (from 3 characters to 4 or more) will result in better guesses by LERS and therefore lower entropy but will also require more data to be used for training purposes and therefore days or perhaps weeks of processing time. Also, the data that was used was noisy. Noisy data means there were characters that were not recognized by the systems as English text characters. Remember, only the 26 English letters and the space are considered in this experiment. Since a huge sample of the English text was needed to train the system, a scanner was used to scan in over 150 pages of the novel Scruples II into a text file. Using an editor, I tried to clean up the text file by getting rid of all illegal characters (all punctuation, all numbers, and so on). There were a few illegal characters that were mistakenly left in. Getting rid of all illegal characters will cause LERS to be trained with better input data and therefore will produce better results and reduce the entropy of the text.

Using LERS will solve the problem of finding an identical twin at the receiver end. LERS can be a good predictor at the transmitter and at the receiver with exact same algorithm used for picking the next character in the sequence at both ends, a requirement for the Shannon's communication system (figure 1). In the next section we will talk about some other experiments that can be performed using LERS in the future.

## REFERENCES

[1] C. E. Shannon, "Prediction and entropy of printed English," Bell Syst. Techn. J., pp. 50-64, Jan. 1951.

[2] V. Maixner, "Some remarks on entropy prediction of natural language texts, " Inform. Stor. Retr., Vol. 7, pp. 293-295, 1971.

[3] A. P. Savchuk, “On the evaluation of the entropy of language using the method of Shannon,” Theory Prob. Appl., vol. 9, no. 1, pp. 154-157, 1964.

[4] T. Nemetz, “On the experimental determination of the entropy,” Kybern. 10, pp. 137-139, 1972.

[5] G. P. Basharin, “On a statistical estimate for the entropy of a sequence of independent random variable,” Theory Prob. Appl., vol. 4, no. 3, pp. 333-336, 1959.

[6] E. Pfaffelhuber, “Error estimation for the determination of entropy and information rate from relative frequency,” Kybern. 8, pp. 50-51, 1971.

[7] C. R. Blyth, “Note on estimating information,” Tech. Rep. 17, Dept. of Statistics, Stanford Univ., Stanford, CA, 1958.

[8] G. A. Barnard, “Statistical calculation of word entropies for four western languages,” IRE Trans. Inform. Theory, no. 1, pp. 49-53, 1955.

[9] M. Grignetti, "A note on the entropy of words in printed English," Inform. Contr., Vol. 7, pp. 304-306, 1964.

[10] W. J. Paisley, "The Effects of authorship, topic structure, and time of composition on letter redundancy in English text," J. Verbal Learn. Behav. 5, pp. 28-34, 1966.

[11] N. G. Burton and J. C. R. Licklider, "Long-range constraints in the statistical structure of printed English," Amer. J. Psych., no. 68, pp. 650-653, 1955.

[12] A. Treisman, “ Verbal responses and contextual constraints in language, “ J. Verbal Learn. Behav. 4, pp. 118-128, 1965.

[13] H. E. White, “Printed English compression by dictionary encoding, ”Proc. IEEE, vol. 55, no. 3, pp. 390-396, Mar. 1967.

[14] D. Jamison and K. Jamison, “A note on the entropy of partially known languages,” Inform. Contr., vol, 12, pp. 164-167, 1968.

[15] M. A. Wanas, A. I. Zayed, M. M. Shaker, and E. H. Taha, “First-second- and third-order entropies of Arabic text, “ IEEE Trans. Inform. Theory, vol. IT-22, no. 1,p. 123, Jan. 1967.

[16] G. A. Miller and J. A. Selfridge, "Verbal context and the recall of meaningful material," Amer. J. Psych., Vol. 63, pp. 176-185, 1950.

[17] N. M. Blackman, "Prevarication versus redundancy," Proc. IRE, pp. 1711-1712, 1962.

[18] T. M. Cover, "A convergent gambling estimate of the entropy of English," IEEE Trans. Info. Theory, Vol. IT-24, no. 4, pp. 413-421, 1978.

[19] A. M. Yaglom and J. M. Yaglom, Probability and Information. 3rd revised ed. Leningrad: Science House, 1973, Chapter IV, Part 3, pp. 236-329.

[20] G. Dewey, "Relative Frequency of English Speech Sounds," Harvard University Press, 1923.

[21] J. W. Grzymala-Busse, “LERS- A System For Learning From Examples Based on Rough Sets, In Intelligent Decision support. Handbook of applications and advances of the rough set theory, by R. Slowinski, Kluwer Academic Publishers, pp. 3-18 (1992).

[22] J. W. Grzymala-Busse, ”Knowledge Acquisition Under Uncertainty- A Rough Set Approach,” Journal of Intelligent & Robotic Systems 1, pp. 3-16 (1988).

[23] J. W. Grzymala-Busse, ”Managing Uncertainty in Expert Systems,” Kluwer Academic Publishers, 1991.

[24] Z. Pawlak, Rough sets. Int. J. Computer and Information Sci., 11, 1982, 341-356.

[25] Z. Pawlak, Rough Classification. Int. J. Man-Machine Studies 20, 1984, 469-483.

[26] E. Plaza, and R. Lopez de Mantaras. A case-based apprentice that learns from fuzzy examples. Proc. of the 5th Int. Symp. on Methodologies for intelligent systems, 1990, 420-427.

[27] William L. Hays and Robert L. Winkler, Statistics, Vol. I, II, 1970.

[28] Stephen Wolfram, Mathematica, second edition

[29] Fletcher Pratt, "Secret and Urgent," Blue Ribbon Books, 1942.