

A software for learning information theory Basics with emphasis on entropy of spanish

Fabio G. Guerrero^{1*} Lucio A. Pérez²

^{a*} *Electrical Engineering School, Universidad del Valle, Cali, Colombia*

^b *IPTotal Software S.A, Cali, Colombia*

^{*} *e-mail: fguerrer@univalle.edu.co*

(Recibido: Febrero 17 de 2008- Aceptado: Julio 4 de 2008)

ABSTRACT

In this paper, an educational software aimed to help learning Information Theory basics in a practical way is reported. The software, called IT-tutor-UV, can produce entropy analysis for both natural and artificially produced Spanish text. In the last case use of a modern existing Spanish corpus for the modelling of the source is done. Source coding, channel coding, and a binary symmetric channel are also included in this educational tool as part of the learning experience.

KEYWORDS: Entropy, Information theory, Natural languages, Source coding, Channel coding, Education.

Software para el aprendizaje de aspectos Básicos de teoría de la información con Énfasis en entropía del español

RESUMEN

En este trabajo se reporta un software educativo dirigido al aprendizaje de conceptos básicos de Teoría de la Información de una forma práctica. El software, llamado IT-tutor-UV, puede producir análisis de entropía tanto para texto de lenguaje español natural como artificialmente producido. En este último caso se hace uso de un corpus moderno existente del español para el modelado de la fuente. Como parte de la experiencia de aprendizaje en esta herramienta educativa, se incluyen también funcionalidades de codificación de fuente, codificación de canal y un canal binario simétrico.

PALABRAS CLAVE: Entropía, Teoría de la Información, Idiomas Naturales, Codificación de Fuente, Codificación de canal, Educación.

1. INTRODUCTION

The teaching of information theory basic concepts is usually based on mathematical derivations. For instance, concepts such as the entropy rate of a language are often presented at theoretical level. This means that students should have a considerable capacity of abstraction in order to understand thoroughly.

From other side, there is an evident lack of tools for Information Theory specifically oriented to education. For example, an exhaustive search done by the authors on April 2008 over the EIEE Transactions on Education, the International Journal of Engineering Education, and the International Journal of Electrical Engineering Education provided no information about tools oriented to the teaching and learning of Information Theory.

A software tool intended to the teaching and learning of information theory basic concepts was designed at the Electrical and Electronic Engineering Department at Universidad del Valle (Cali, Colombia, South America) and has being used at the graduate course Information Theory (class code 710599) since the fall 2006 season. The name given to this educational software was IT-tutor-UV, which stands for Information Theory tutor at University of Valle.

The IT-tutor-UV was developed using Matlab 7.0 under the Windows operating system. This software is able to analyze both natural text and also produce artificial Spanish using a source modelled as a first-order stochastic process.

In section 2 the materials and methods used for the construction of the IT-tutor-UV is presented. In section 3 some of the ways in which the IT-tutor-UV can be used are discussed with some educational examples. Finally in section 4 the main conclusions of this work are summarized.

2. MATERIALS AND METHODS

The IT-tutor-UV has the following logical modules: information source, entropy analysis, source coding, channel coding, and binary symmetric channel as shown in Fig. 1.

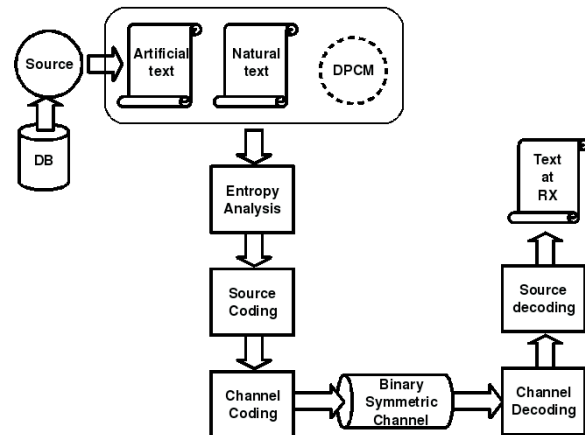


Fig. 1. IT-tutor-UV block diagram.

For the discrete case the source can be either natural text or artificial text. A DPCM module can also be added by the students for the instructional goal of dealing with a continuous source. The discrete source for producing artificial text is modelled as a stochastic process using a first-order word approximation (i.e., statistically independent words and the right frequency in Spanish). The user can specify the number of words for the file of the generated text sequence.

A Spanish word frequency database compiled by Alameda and Cuetos [1] is used for the generation of artificial Spanish. This database contains the frequency of the 81323 most common Spanish words obtained from a corpus of 1950375 words appearing in different representative sources of the Spanish literature such as newspapers, transcriptions, etc.

At the entropy analysis block the observed entropy is calculated using the classical entropy equation (1), where p_i is the probability of the particular type of symbol selected by the user.

$$H = - \sum_i p_i \log p_i \quad (1)$$

Currently the entropy analysis module calculates letter frequency, digram frequency, trigram frequency, or the word, di-word, three-word, and four-word frequency according to the selection made by the user through the graphical user interface. In fact, there would be no limit on the number of words that could be analysed except for the enormous time it would take to analyze large symbols (i.e. symbols formed by a long number of words).

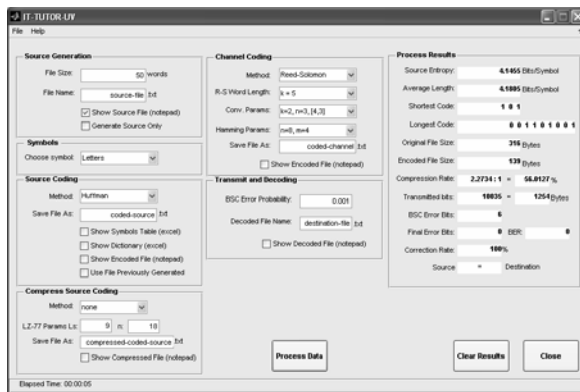


Fig. 2. IT-tutor-UV graphical user interface.

The source module can also generate an Excel file with the frequency of the symbols found in the generated text. It is also possible to display and save the word sequence as a text file for later use. Before this is done, the database file should be loaded using the Windows™ ODBC Data Source Administrator. Fig. 2 shows a snapshot of the IT-tutor-UV graphical user interface.

Due to the long time that it can take to process a long sample when running the discrete source, the source coding, and the channel coding modules at the same time, the IT-tutor-UV has a box switch to use the discrete source module alone. This is useful to quickly observe the letter, digram, or trigram entropy, for instance.

Fig. 3 shows a fifty word text sample produced by the IT-tutor-UV. The text sample in Fig. 3 conveys no meaning. The reason for this is that the text in Fig. 3 was produced using a first-order word model which naturally produce a text which lacks many of the rules given by the grammar and structure of Spanish. The rules of grammar lead to create a statistical dependency between words that are near to each other. Therefore, the more

dependency is considered by the discrete source model the greater the intelligibility of the text produced artificially will be.

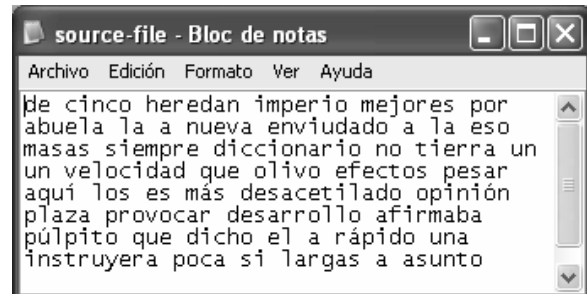


Fig. 3. Fifty word text sample.

The source coding module supports the Huffman coding method [2], the Arithmetic coding method, and a basic implementation of the Lempel-Ziv 77 coding method. The user can select from the graphical user interface the type of symbol that will be used for the coding process as letters, digrams, trigrams, words, di-words, three-words and four-words.

In the channel coding module, the Reed-Solomon (RS) and convolutional coding techniques have been included as representative coding methods from the algebraic and statistical coding fields respectively.

When the RS coding is enabled, users are able to try different values for the RS word length k (e.g., 5, 9, or 13), so they can verify the error correction capacity of the block coding when different amounts of redundancy are used. For convolutional coding, two simple configurations of codes, which are shown in Fig. 4 and Fig. 5, can be chosen by the users.

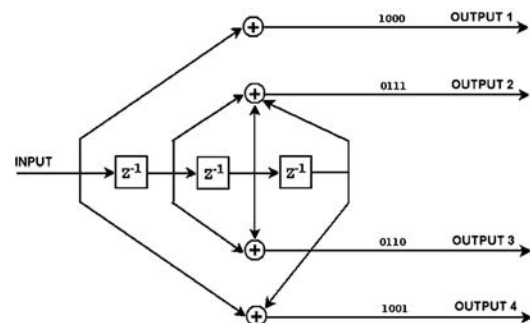


Fig. 4. Convolutional code, Rate= $\frac{1}{4}$, $K=3$.

At the instructional level, students can observe the effect of the redundancy on error correction capacity of convolutional codes and they can make some comparisons with block coding not only on the basis of error correction capacity but also on coding time.

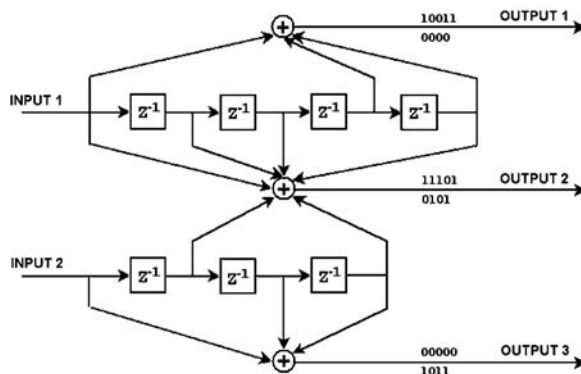


Fig. 5. Convolutional code, Rate = 2/3, $K=[4,3]$

For educational purposes, the channel has been modelled as a binary symmetric channel with error probability p because of its simplicity. Of course, nothing precludes adding modules for other type of channels. The value of p can also be entered using the graphical user interface.

Once the message has been coded and sent through this channel, the IT-tutor-UV can display the bits transmitted, the error bits introduced by the channel, the BER (Bit Error Rate), and the final error bits remaining after the message has been decoded. These values are used to calculate the correction capacity of the codes, which is displayed as a percentage of correction. Finally, a symbol-to-symbol comparison between the original source file and the decoded file is provided through the graphical interface.

3. RESULTS

3.1. Letters, digrams, and trigrams

Table 1 shows some values of the entropy for letters, digrams, and trigrams obtained using the IT-tutor-UV. The time needed to calculate these results depends on the number of letters forming the symbols as the number of combinations

increase: the larger the number of letters, the larger the time it would take the computer to finish the calculations.

For instance, the trigram calculation in Table 1 took 7.25 hours on a laptop with 1 GB of RAM memory and a 64-bit Processor at 2.2 GHz. It should be noticed that the text sample's length (number of words) has to be long enough to get the convergent values of the entropy for the respective type of symbol in Table I.

Table 1. Letter, digram, and entropy of artificial Spanish

Symbol	Sample length (words)	H_{symbol}
Letter	200000	4.1484
Digram	100000	7.3223
Trigram	50000	10.1432

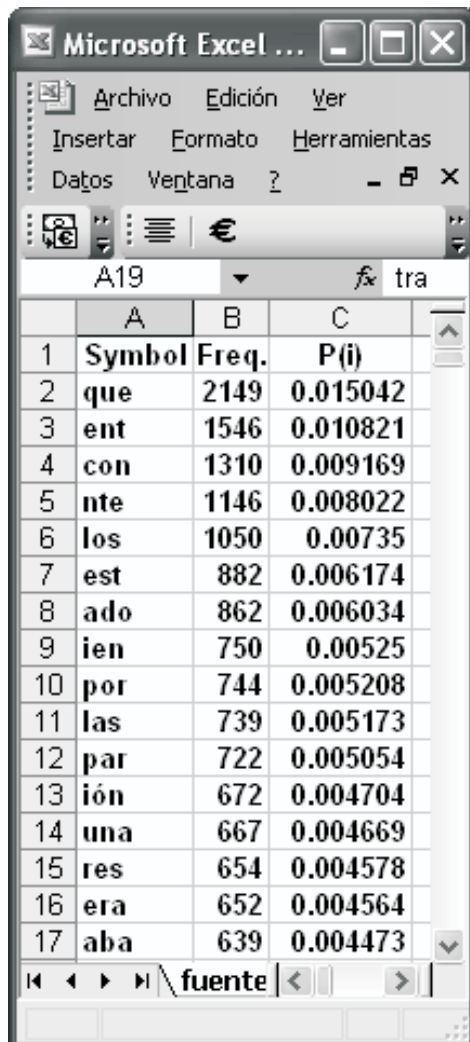
For entropy analysis, the IT-tutor-UV can take into account every symbol encountered on the analyzed text. This is the case, for instance, for the accented vowels (á, é, í, ó, ú). This makes sense because in the Spanish language these vowels can give different meanings to the words. For the same reason the IT-tutor-UV can also differentiate the “ñ” letter from the “n” letter. So if these six additional letters are considered then the total numbers of letters of the alphabet will be 33. This feature of the IT-tutor-UV allows providing much more detailed frequency tables for Spanish than tables which usually consider a limited alphabet of only 26 letters as in [3].

As can be seen on Table 1, the single letter entropy obtained using the IT-tutor-UV was 4.15 bits/letter which is greater than the 4.0 bits/letter entropy obtained when considering only 26 letters, in accordance with the basic properties of entropy.

For digrams, the obtained entropy was 7.3223 bits/symbol. For trigrams, the obtained entropy was 10.1432 bits/symbol, based on 3821 trigrams.

Figure 6 shows the snapshot with the 16 most frequent trigrams of an Excel trigram file produced by the IT-tutor-UV after analyzing a

52507-word literary work. In this case, there were 3833 trigrams, and the sum of the trigram frequencies was 142865.



	A	B	C
	Symbol	Freq.	P(i)
2	que	2149	0.015042
3	ent	1546	0.010821
4	con	1310	0.009169
5	nte	1146	0.008022
6	los	1050	0.00735
7	est	882	0.006174
8	ado	862	0.006034
9	ien	750	0.00525
10	por	744	0.005208
11	las	739	0.005173
12	par	722	0.005054
13	ión	672	0.004704
14	una	667	0.004669
15	res	654	0.004578
16	era	652	0.004564
17	aba	639	0.004473

Fig. 6. Trigram frequency example.

As it was mentioned the same type of entropy analysis can be done by the IT-tutor-UV when using a sample of natural Spanish language such as a literary work as input. This feature would allow the students, for instance to find word entropies and compare them with word entropy values found by indirect methods such as the Barnard's method [4].

3.2 Entropy Rate

In order to introduce the entropy rate concept [5] to students, first the entropy per word of the IT-tutor-UV's source is calculated using (1) as follows

$$\sum_{i=1}^{81323} p_i \log_2 p_i = 10.4281$$

Thus, the entropy per letter is $10.4281/\alpha = 2.2198$ bits/letter. This value is also the entropy rate, H_w , of this source because the symbols it generates (i.e., words) are independent.

Since in practice source coding is done by compressors, at this point the instructor could challenge students to find the compressor with the best compression ratio. For example, from the several compression programs the authors tested on a 10 million word file without spaces created by IT-tutor-UV, the best compression result was produced by PPMonstr, a compression tool written by Dmitry Shkarin [6]. PPMonstr was able to compress to 2.402 bits/letter, a value quite close to the 2.2198 bits/letter entropy limit. Of course, there may be better compressors but that discussion is beyond the scope of this paper.

3.3 Source Coding and Channel Coding Example

When the Huffman source coding option is selected, the IT-tutor-UV provides the Huffman dictionary, the longest codeword, the shortest codeword, the average code length, and the compression rate.

Figure 7 shows a snapshot of a Huffman dictionary in the Excel file generated by the IT-tutor-UV program using words as source symbols.

The following example shows the results obtained for a short test text file:

File size: 10000 words
 Source coding: Huffman method
 Source symbols: words
 Channel Coding: Convolutional (rate = 2/3, K = [4, 3])

BSC Error Probability: 0.005
 Source Entropy: 8.009 bits/symbol
 Average Code Length: 8.022 bits/symbol
 Shortest code: 1011
 Largest code: 00100000001
 Original File Size: 5749 bytes
 Encoded File Size: 1003 bytes
 Compression rate: 82.5535%
 Number of transmitted bits: 12033
 Error Bits: 72
 Error Bits after correction: 1
 BER: 0.00012466
 Correction rate: 98.6111%
 Source file ? Destination File

From these values, it can be seen how Huffman coding closely approaches the source entropy when the coder's dictionary (i.e., the matching between code symbols and source symbols that the receiver needs to recover the text into its original form) is not considered.

	A	B	C	D	E	F	G	H	I	J	K
1	Symbol	Code									
2	de	0	0	0	1						
3	que	0	0	0	0	0					
4	profundo	1	0	1	0	1	0	1	1	0	1
5	solamente	1	0	1	0	1	0	1	1	0	0
6	el	1	1	1	0	0					
7	recuerdo	1	0	1	0	1	0	1	1	1	1
8	le	1	1	1	1	0	1	0			
9	los	1	1	0	1	0	0				
10	sobre	0	0	0	0	1	0	1	1		
11	estaré	1	0	1	0	1	0	1	1	1	0
12	siluetas	1	0	1	0	1	0	1	0	0	1
13	un	1	1	0	0	0	0				
14	utilidad	1	0	1	0	1	0	1	0	0	0
15	sus	1	1	1	0	1	0	1			
16	día	0	0	1	1	1	0	1	1		
17	costado	1	0	1	0	1	0	1	0	1	1

Fig. 7. Example of a Huffman dictionary file generated by the IT-tutor-UV.

Using the IT-tutor-UV, students can verify how close the arithmetic coding is to the entropy limit and what its advantages over Huffman coding are. As it is well known the Arithmetic coding algorithm overcomes Huffman coding limitations by assigning a label to the whole sequence of symbols in the message. This label is generated according to the frequency of each symbol; therefore, there is only a single code for the entire message, and a code dictionary is not necessary, etc.

The IT-tutor-UV's executable code can be downloaded at the software homepage [7] and can be used under a GPL license.

3.4 Some homeworks using the IT-TUTOR-UV

The IT-TUTOR-UV has been used do far used at Universidad del Valle for three main homeworks: the first one is a detailed entropy analysis of a literary work in Spanish, the second one is to extend the analysis capabilities of the IT-TUTOR-UV (for instance, analysis of symbols of higher order), and the third one is adding a module to deal with continuous source (for instance, using DPCM). Extra credit is given to students than provide improvement to the software by themselves.

4. CONCLUSION

The IT-tutor-UV is an educational software aimed to help transforming the experience of learning some of the highly abstract concepts of information theory such as discrete source generation, information entropy, channel capacity, and so on, into a more tangible and enjoyable experience.

The information theory concepts that the IT-tutor-UV can support at instructional level are: entropy calculation, entropy basic properties, Huffman coding, Arithmetic coding, LZ-77 coding, RS-coding, Hamming coding, convolutional coding, and binary symmetric channel.

One of the several possible pedagogical applications of the IT-tutor-UV is obtaining complete and precise frequency lists of letters, digrams, trigrams, words, di-words, three-words,

and four-words of Spanish. Beyond information theory, these lists could be useful for the study of classic decryption systems based on frequency.

As most pieces of software, the IT-tutor-UV can certainly be improved. Therefore, the authors would welcome any suggestions and recommendations for the enhancement of this educational software.

5. ACKNOWLEDGMENT

Thanks to Mr. Fabián Barco, graduate student at Universidad del Valle, for some minor but useful extensions to the IT-TUTOR-UV.

6. BIBLIOGRAPHIC REFERENCES

[1] F. Cuetos, and J. R. Alameda, *Diccionario de las unidades Lingüísticas del Castellano: Volumen I: Orden Alfabético / Volumen II: Orden por Frecuencias*. [Online]. Available: <http://www.uhu.es/jose.alameda>, 1995.

[2] A. D. Huffman, *A Method for the Construction of Minimum-Redundancy Codes*, Proc. IRE, Vol 40, pp 1098-1101, Sept., 1952.

[3] J. Ferrero, *La lengua española y los medios audiovisuales*, Congreso de la Lengua Española. Sevilla, [Online]. Available: http://cvc.cervantes.es/obref/congresos/sevilla/comunicacion/mesaredon_ferrero.htm, 1992

[4] G. A. Barnard, III, *Statistical calculation of word entropies for four western languages*, IEEE Trans. Information Theory, Vol. 1, pp. 49–53, Mar., 1955.

[5] R. B. Wells, *Applied Coding and Information Theory for Engineers*. Upper Saddle River, New Jersey: Prentice Hall, ch. 1, 2002.

[6] D. Shkarin, *Monstrous PPMII compressor based on PPMd var.I*, [Online]. Available: <http://compression.ru/ds/>, April 2002.

[7] F. G. Guerrero and L. A. Perez, *IT-tutor-UV instructional software*, [Online], Available: <http://sistel-uv.univalle.edu.co/it-tutor-uv.html>, December 2006