Experiments in Unsupervised Entropy-Based Corpus Segmentation

André Kempe
Xerox Research Centre Europe – Grenoble Laboratory
6 chemin de Maupertuis – 38240 Meylan – France
andre.kempe@xrce.xerox.com
http://www.xrce.xerox.com/research/mltt

Abstract

The paper presents an entropy-based approach to segment a corpus into words, when no additional information about the corpus or the language, and no other resources such as a lexicon or grammar are available. To segment the corpus, the algorithm searches for separators, without knowing a priory by which symbols they are constituted. Good results can be obtained with corpora containing "clearly perceptible" separators such as blank or new-line.

1 Introduction

The paper presents an approach to segment a corpus into words, based on entropy. We assume that the corpus is not annotated with additional information, and that we have no information whatsoever about the corpus or the language, and no linguistic resources such as a lexicon or grammar. Such a situation may occur e.g. if there is a (sufficiently large) corpus of an unknown or unidentified language and alphabet.[1] Based on entropy, we search for separators, without knowing a priory by which symbols or sequences of symbols they are constituted.

Over the last decades, entropy has frequently been used to segment corpora [Wolff, 1977, Alder, 1988, Hutchens and Alder, 1998, among many others]. and it is commonly used with compression techniques. Harris [1955] proposed an approach for segmenting words into morphemes that, although it did not use entropy, was based on an intuitively similar concept: Every symbol of a word is annotated with the count of all possible successor symbols given the substring that ends with the current symbol, and with the count of all possible predecessor symbols given the tail of the word that starts with the current symbol. Maxima in these counts are used to segment the word into morphemes.

All steps of the present approach will be described on the example of a German corpus. In addition, we will give results obtained on modified versions of this corpus, and on an English corpus.

2 The Approach

2.1 The Corpus

We assume that any corpus $\mathcal{C}$ can be described by the expression:

$$\mathcal{C} = \mathcal{S}* \ \mathcal{T} \ [\mathcal{S}+ \ \mathcal{T}]* \ \mathcal{S}* \qquad (1)$$

There must be at least one token $\mathcal{T}$ ("word") which is a string of one or more symbols s :

$$\mathcal{T} = s+ \qquad (2)$$

Different tokens $\mathcal{T}$ must be separated form each other by one or more separators $\mathcal{S}$ which are strings of zero or more symbols s :

$$\mathcal{S} = s* \qquad (3)$$

Separators can consist of blanks, new-line, or "real" symbols. They can also be empty strings.

2.2 Recoding the Corpus

We will describe the approach on the example of a German corpus.

First, all symbols s (actually all character codes) of the corpus are recoded by strings of "visible" ASCII characters. For example:[2]

[1] Such a corpus can be electronically encoded with arbitrarily defined symbol codes.

[2] In this example, \ denotes that the current line is not finished yet but rather continues on the next line.

Für Instandsetzung und Neubau der \
Kanalisation dürften in
den nächsten zehn Jahren Beträge in \
Milliardenhöhe ausgegeben werden.
Allein in den alten Bundesländern müssen bis \
zur Jahrhundertwende die
Kommunen km des insgesamt km langen \
Kanal- und
Leitungsnetzes sanieren.

is recoded as:[3]

F ü r BL I n s t a n d s e t z u n g BL u n d BL N e u b a u BL d e r BL K a n a l i s a t i o n BL d ü r f t e n BL i n NL d e n BL n ä c h s t e n BL z e h n BL J a h r e n BL B e t r ä g e BL i n BL M i l l i a r d e n h ö h e BL a u s g e g e b e n BL w e r d e n . NL A l l e i n BL i n BL d e n BL a l t e n BL B u n d e s l ä n d e r n BL m ü s s e n BL b i s BL z u r BL J a h r h u n d e r t w e n d e BL d i e NL K o m m u n e n BL k m BL d e s BL i n s g e s a m t BL k m BL l a n g e n BL K a n a l - BL u n d NL L e i t u n g s n e t z e s BL s a n i e r e n . BL

If the language and the alphabet are unknown or unidentified, the symbols of the corpus can be encoded by arbitrarily defined ASCII strings.

2.3 Information and Entropy

We estimate probabilities of symbols of the corpus using a 3rd order Markov model based on maximum likelihood. The probability of a symbol s with respect to this model M and to a context c can be estimated by:

$$p(s|M,c) = \frac{f(s,M,c)}{f(M,c)} \tag{4}$$

The *information* of a symbol s with respect to the model M and to a context c is defined by:

$$I(s|M,c) = -log_2\ p(s|M,c) \tag{5}$$

Intuitively, information can be considered as the surprise of the model about the symbol s after having seen the context c. The more the symbol is unexpected from the model's experience, the higher is the value of information [Shannon and Weaver, 1949].

The *entropy* of a context c with respect to this model M expresses the expected value of information, and is defined by:

$$H(M,c) = \sum_{s\in\Sigma} p(s|M,c)\ I(s|M,c) \tag{6}$$

Monitoring entropy and information across a corpus shows that maxima often correspond with word boundaries [Alder, 1988, Hutchens and Alder, 1998, among many others].

More exactly, maxima in left-to-right entropy H_{LR} and information I_{LR} often mark the end of a separator string S, and maxima in right-to-left entropy H_{RL} and information I_{RL} often mark the beginning of a separator string, as can be seen in Figure 1. Here, an information value is assigned to every symbol. This value expresses the information of the symbol in a given left or right context. An entropy value is assigned between every two symbols. It expresses the model's uncertainty after having seen the left or right context, but not yet the symbol.

When going from left to right, an end of a separator, is often marked by a maximum in entropy because the next word to the right can start with almost any symbol, and the model has no "idea" what it will be. There is also a maximum in information because the first symbol of the word is (more or less) unexpected; the model has no particular expectation.

Similarly, when going from right to left, a beginning of a separator is often marked by a maximum in entropy because the word next to the left can end with almost any symbol. There is also a maximum in information because the last symbol of the word is (more or less) unexpected; the model has no particular expectation.

Usually, there is no maximum at a beginning of a separator, when going from left to right, and no maximum at a separator ending, when going from right to left, because words often have "typical" beginnings or endings, e.g. prefixes or suffixes.

This means, when we come from inside a word to the beginning or end of this word then the model will anticipate a separator, and since the number of alternative separators is usually small, the model will not be "surprised" to see a particular one. On the other side, when we come from inside a separator to the beginning or end of this separator, although the model will expect a word, it will be "surprised" about any particular word because the number of alternative beginnings or endings of words is large.

It also may be observed that the maxima in one direction are bigger then the maxima in the other direction due to the fact that a particular language may have e.g. stronger constraints on endings than on beginnings of words: A language may employ suffixes with most words in a corpus, which limits the number of endings, but rarely use prefixes, which allows a word to start with almost any symbol.

[3]Note that blanks become "BL" and new-lines become "NL".

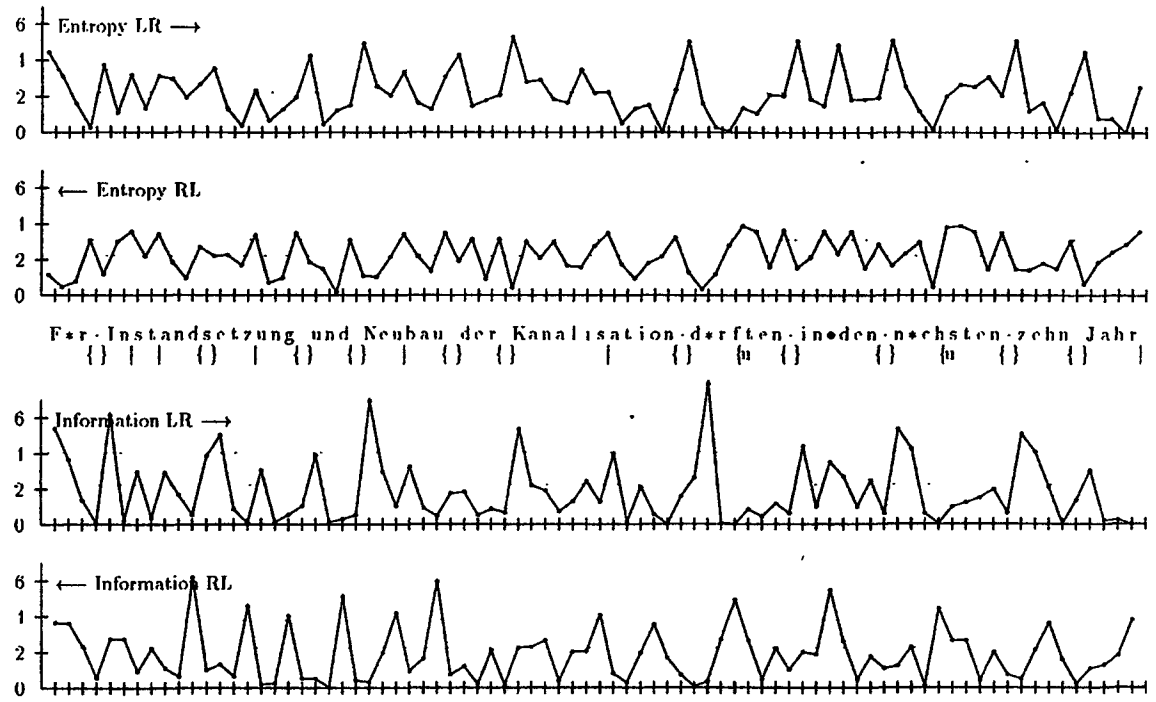

Figure 1: Entropy and information across a section of a German corpus

2.4 Thresholds

Not all maxima correspond with word boundaries. Hutchens and Alder [1998] apply a threshold of $0.5\ log_2\ ||\Sigma||$ to select among all maxima, those that represent boundaries.

The present approach uses two thresholds that are based on the corpus data and contain no other factors: The first threshold τ_{all} is the average of all values of the particular function, H_{LR}, H_{RL}, I_{LR}, or I_{RL}, across the corpus. The second threshold τ_{max} is the average of all maxima of the particular function. All graphs of Figure 1 contain both thresholds (as dotted lines).

To decide whether a value v of H_{LR}, H_{RL}, I_{LR}, or I_{RL} should be considered as a boundary, we use the four functions:

$$b_0(v) : v \geq \tau_{max} \quad (7)$$

$$b_1(v) : v \geq \tau_{all} \quad (8)$$

$$b_2(v) : ismax(v) \quad (9)$$

$$b_3(v) : \neg ismin(v) \quad (10)$$

2.5 Detection of Separators

To find a separator, we are looking for a *strong boundary* to serve as a beginning or end of the separator. In the current example, we have chosen as a criterion for strong boundaries:

$$\begin{aligned}(b_0(h) \wedge b_2(h)) \wedge (b_1(i) \vee b_3(i)) = \quad (11)\\ ((h \geq \tau_{max}(H)) \wedge ismax(h))\\ \wedge\ ((i \geq \tau_{all}(I)) \vee \neg ismin(I))\end{aligned}$$

Here H and I mean either H_{LR} and I_{LR} if we are looking for the end of a separator, or H_{RL} and I_{RL} if we are looking for the beginning of a separator. The variables h and i denote values of these functions at the considered point.

Once a strong boundary is found, we search for a *weak boundary* to serve as an ending that matches the previously found beginning, or to serve as a beginning that matches the previously found ending. For weak boundaries, we use the criterion:

$$\begin{aligned}(b_1(h) \wedge b_2(h)) \wedge (b_1(i) \vee b_3(i)) = \quad (12)\\ ((h \geq \tau_{all}(H)) \wedge ismax(h))\\ \wedge\ ((i \geq \tau_{all}(I)) \vee \neg ismin(I))\end{aligned}$$

If a matching pair of boundaries, i.e. a beginning and an end of a separator, are found, the separator

is marked. In Figure 1 this is visualized by | for empty and { } for non-empty separators.

The search for a weak boundary that matches a strong one is stopped (without success) either after a certain distance[4] or at a *breakpoint*. For example, if we have the beginning of a separator and search for a matching end then the occurrence of another beginning will stop the search. As a criterion for a breakpoint we have chosen:

$$(b_1(h) \wedge b_2(h)) \vee (b_1(i) \wedge b_2(i)) = \quad (13)$$
$$((h \geq \tau_{all}(H)) \wedge ismax(h))$$
$$\vee \ ((i \geq \tau_{all}(I)) \wedge ismax(i))$$

If the search for a matching point has been stopped for either reason, we need to decide whether the initially found strong boundary should be marked despite the missing match. It will only be marked if it is an *obligatory boundary*. Here we apply the criterion:

$$(b_0(h) \wedge b_2(h)) \wedge (b_0(i) \wedge b_2(i)) = \quad (14)$$
$$((h \geq \tau_{max}(H)) \wedge ismax(h))$$
$$\wedge \ ((i \geq \tau_{max}(I)) \wedge ismax(i))$$

In Figure 1 these unmatched obligatory boundaries are visualized by {u or }u.

Each of the four criteria, for strong boundaries, weak boundaries, break points, and obligatory boundaries, can be built of any of the four functions $b_0()$ to $b_3()$ (eq.s 7 to 10).

2.6 Validation of Separators

All separator strings that have a matching beginning and end marker are collected and counted.

Alias	f_{separ}	$f_{context}$	f_{total}	Separator
l_0	95 103	1 484	115 619	BL
l_1	10 841	850	20 011	NL
l_2	45 475	621	1 024 152	
l_3	3 464	450	11 637	t BL
l_4	697	360	3 096	-
l_5	1 271	281	5 736	. BL
l_6	1 328	241	17 053	e BL
l_7	3 223	199	48 136	s
l_8	6 793	160	138 769	e
l_9	3 306	126	32 049	e r
l_{10}	545	119	4 110	. NL
l_{11}	162	108	1 372	t NL

Table 1: Separators from a German corpus (truncated list)

[4]In the example, the maximal separator length is set to 6. This seems sufficient because we found no separators longer than 3 so far (Tables 1 to 5).

Table 1 shows such separators collected from the German example corpus. Column 5 contains the strings that constitute the separators, column 2 shows the count of these strings as separators, colunm 3 says in how may different contexts[5] the separators occurred, colunm 4 shows the total count of the strings in the corpus, and column 1 contains aliases furtheron used to denote the separators. In Table 1 all separators are sorted with respect to column 3. From these separators we retain those that are above a defined threshold relative to the number of different contexts of the top-most separator. In all examples throughout this article, we are using a relative threshold of 0.5, which means in this case (Table 1) that the top-most two separators, "BL" and "NL" that occur in 1484 and 850 different contexts respectively, are retained.[6]

In the corpus, all separators that have been retained (Table 1) and that have at least one detected boundary (Fig. 1), are validated and marked. For the above corpus section this leads to:

F ü r b_0 I n s t a n d s e t z u n g b_0 u n d b_0 N e u
b a u b_0 d e r b_0 K a n a l i s a t i o n b_0 d ü r f t e n
b_0 i n NL d e n b_0 n ä c h s t e n b_0 z e h n b_0 J a h r
e n b_0 B e t r ä g e b_0 i n b_0 M i l l i a r d e n h ö h
e b_0 a u s g e g e b e n b_0 w e r d e n . l_1 A l l e i n
b_0 i n b_0 d e n b_0 a l t e n b_0 B u n d e s l ä n d e r n
b_0 m ü s s e n b_0 b i s b_0 z u r b_0 J a h r h u n d e r t .
w e n d e b_0 d i e NL K o m m u n e n b_0 k m b_0 d e
s b_0 i n s g e s a m t BL k m b_0 l a n g e n b_0 K a n
a l - BL u n d l_1 L e i t u n g s n e t z e s BL s a n i
e r e n . BL

2.7 Recall of Separators

For the above corpus we measured a recall of 86.0 % for both blank (BL) and new-line (NL) together (Table 2).

Alias	Separator	Recall	f_{found}	f_{total}
l_0	BL	88.6 %	102 412	115 619
l_1	NL	71.2 %	14 254	20 011
All		86.0 %	116 666	135 630

Table 2: Recall of separators from a German corpus

Due to the approach, the precision for BL and NL is 100 %. A string which is different from BL and NL cannot be marked as a separator in the above example. If empty string separators were admitted, the precision would decrease.

[5]As context of a separator, we consider the preceding and the following symbol.

[6]In the Tables 1 to 5 the retained separator strings are separated by a horizontal line form the others.

3 More Examples

We applied the approach to modified versions of the above mentioned German corpus and to an English corpus.

3.1 German with Empty String Separators

For this experiment, we remove all original separators, "BL" and "NL", from the above German corpus:

F ü r I n s t a n d s e t z u n g u n d N e u b a u d e r K a n a l i s a t i o n d ü r f t e n i n d e n n ä c h s t e n z e h n J a h r e n B e t r ä g e i n M i l l i a r d e n h ö h e a u s g e g e b e n w e r d e n . A l l e i n i n d e n a l t e n B u n d e s l ä n d e r n m ü s s e n b i s z u r J a h r h u n d e r t w e n d e d i e K o m m u n e n k m d e s i n s g e s a m t k m l a n g e n K a n a l - u n d L e i t u n g s n e t z e s s a n i e r e n .

From this corpus, we collected the separators in Table 3

Alias	f_{separ}	$f_{context}$	f_{total}	Separator
l_0	110 969	1 257	888 522	
l_1	6 355	580	33 335	e n
l_2	5 872	466	54 278	t
l_3	7 975	407	138 769	e
l_4	5 661	374	32 158	e r
l_5	916	345	11 178	.
l_6	3 063	306	48 136	s
l_7	505	297	3 096	-
l_8	399	206	4 566	t e n
l_9	621	189	15 836	t e

Table 3: Separators from a German corpus without blanks and new-line (truncated list)

and obtained the result:

F ü r l_2 I n l_0 s t l_0 a n d s e t z l_0 u l_0 n g l_0 u n d l_0 N e u l_0 b a u l_0 d e r l_0 K a n a l i s l_0 a t i o n d ü r f l_0 t e n l_0 i n d e n l_0 n ä c h l_0 s t e n l_0 z e h n J a h r e n l_0 B e t r ä g l_0 e l_0 i n l_0 M i l l i a r d l_0 e n l_0 h ö h l_0 e a u s l_0 g e l_0 g e b e n l_0 w e r d e n l_0 . A l l l_0 e l_0 i n l_0 i n d l_0 e n l_0 a l t e n l_0 B u n l_0 d l_0 e s l_0 l ä n d l_0 e r l_0 n m ü s s e n l_0 b i s z u r l_0 J a h r h l_0 u n d l_0 e r l_0 t w e n d e l_0 d i e l_0 K o m m u n e n l_0 k m d e l_0 s l_0 i n s l_0 g e s a m l_0 t l_0 k m l_0 l a n g e n l_0 K a n a l - u n d l_0 L e i l_0 t l_0 u n g s l_0 n e t z e s s a n l_0 i e r e n l_0 .

3.2 German with Modified Separators

For the next experiment, we changed all original separators, "BL" and "NL", in the above German corpus into a string from the list { " ", "-", "- -", "#", "# #", "- - -", "# #", "- -" } :[7]

F ü r - - - I n s t a n d s e t z u n g # # u n d - - N e u b a u d e r - K a n a l i s a t i o n - - d ü r f t e n # i n # # d e n - - - n ä c h s t e n # # z e h n - - J a h r e n B e t r ä g e - i n - - M i l l i a r d e n h ö h e # a u s g e g e b e n # # w e r d e n . - - - A l l e i n # # i n - - d e n a l t e n - B u n d e s l ä n d e r n - - m ü s s e n # b i s # # z u r - - - J a h r h u n d e r t w e n d e # # d i e - - K o m m u n e n k m - d e s - - i n s g e s a m t # k m # # l a n g e n - - - K a n a l - # # u n d - - L e i t u n g s n e t z e s s a n i e r e n . -

From this corpus, we collected the separators in Table 4

Alias	f_{separ}	$f_{context}$	f_{total}	Separator
l_0	127 490	844	1 108 920	
l_1	4 966	618	33 907	# #
l_2	3 875	591	17 247	- - -
l_3	3 516	532	68 198	- -
l_4	6 876	292	138 769	e
l_5	4 494	268	32 059	e r
l_6	3 699	265	48 136	s
l_7	5 161	227	54 278	t
l_8	378	189	2 437	. # #
l_9	335	179	3 732	. - -
l_{10}	1 588	170	32 909	e n
l_{11}	1 555	140	41 035	a

Table 4: Separators from a German corpus with modified separators (truncated list)

and obtained the result:

F ü r l_2 I n l_0 s t l_0 a n d l_0 s l_0 e t z l_0 u n g l_0 # # l_0 u n d l_3 N e u l_0 b a u l_0 d e r - K l_0 a l_0 n l_0 a l l_0 i s a t i o n l_0 - - d ü r f l_0 t e n # i n l_1 d e n l_2 l_0 n ä c h l_0 s t e n l_1 z e h n l_3 l_0 J a h r e n B e t r ä g e - i n l_3 l_0 M i l l i a r d l_0 e n h ö h l_0 e # l_0 a u s l_0 g e l_0 g e b l_0 e n l_0 # # l_0 w e r d e n l_0 . l_2 A l l e i n l_0 # # l_0 i n l_3 d l_0 e n a l l_0 t e n - B u n d e s l ä n d e r n l_3 l_0 m ü s s l_0 e n # b i s l_1 l_0 z u r l_2 J a h r h u n d l_0 e r l_0 t w e n d e l_0 # # d i e l_3 K o m m u n l_0 e n l_0 k m - d e s l_0 - - l_0 i n s l_0 g e s a m t # k m # # l_0 l a n g e n l_2 l_0 K a n a l l_0 - # # u n d l_3 L l_0 e i t l_0 u n g s l_0 n e t z l_0 e l_0 s s a n l_0 i e r e n . -

[7] The replacement of every blank and new-line was done by rotation through the list.

3.3 English Corpus

On an English corpus where all original separators have been preserved:[8]

In the days when the spinning-wheels \
hummed busily in the
farmhouses and even great ladies clothed \
in silk and
thread lace had their toy spinning-wheels \
of polished oak there might be seen in \
districts far away among the lanes
or deep in the bosom of the hills certain \
pallid undersized
men who by the side of the brawny \
country-folk looked
like the remnants of a disinherited race.

we measured the information and entropy shown in Figure 2, collected the separators in Table 5,

Alias	f_{separ}	$f_{context}$	f_{total}	Separator
l_0	164 744	1 082	181 518	BL
l_1	7 700	602	20 000	NL
l_2	19 983	374	1 096 301	
l_3	2 689	323	5 895	. BL
l_4	4 009	223	32 576	e BL
l_5	734	216	1 253	. NL .
l_6	6 039	171	105 185	e
l_7	357	167	647	. ' NL .
l_8	646	154	2 876	. NL
l_9	489	99	17 789	t BL
l_{10}	1 932	96	71 247	a
l_{11}	180	96	1 742	-

Table 5: Separators from an English corpus (truncated list)

and obtained the result:

I n l_0 t h e l_0 d a y s l_0 w h e n l_0 t h e l_0 s p i n n
i n g - w h e e l s l_0 h u m m e d l_0 b u s i l y l_0 i n
l_0 t h e l_1 f a r m h o u s e s l_0 a n d l_0 e v e n l_0 g
r e a t l_0 l a d i e s l_0 c l o t h e d l_0 i n l_0 s i l k l_0
a n d NL t h r e a d l_0 l a c e l_0 h a d l_0 t h e i r l_0
t o y l_0 s p i n n i n g - w h e e l s BL o f l_0 p o l i s
h e d l_0 o a k l_0 t h e r e l_0 m i g h t l_0 b e l_0 s e e
n l_0 i n l_0 d i s t r i c t s l_0 f a r l_0 a w a y BL a m
o n g l_0 t h e l_0 l a n e s NL o r l_0 d e e p l_0 i n l_0 t
h e l_0 b o s o m l_0 o f l_0 t h e l_0 h i l l s l_0 c e r t a
i n l_0 p a l l i d l_0 u n d e r s i z e d l_1 m e n l_0 w h
o l_0 b y l_0 t h e l_0 s i d e BL o f l_0 t h e l_0 b r a w
n y l_0 c o u n t r y - f o l k l_0 l o o k e d l_1 l i k e l_0
t h e l_0 r e m n a n t s BL o f l_0 a l_0 d i s i n h e r i
t e d l_0 r a c e .

[8] In this example, \ denotes that the current line is not finished yet but rather continues on the next line.

4 Conclusion and Future Investigations

The paper attempted to show that entropy and information can be used to segment a corpus into words, when no additional knowledge about the corpus or the language, and no other resources such as a lexicon or grammar are available.

To segment the corpus, the algorithm searches for separators, without knowing a priory by which symbols or sequences of symbols they are constituted.

Good results were obtained with a German and an English corpus with "clearly perceptible" separators (blank and new-line). Precision and recall decrease if the original separators of these corpora are removed or changed into a set of different co-occurring separators.

So far, only separators and their frequencies have been taken into account. Future investigations may include:

- the use of frequencies of tokens and their different alternative contexts, to validate these tokens and the adjacent separators. and
- a search for criteria (based on the corpus itself and on the obtained result) to evaluate the "quality" of segmentation, thus enabling a self-optimizing approach.

Acknowledgements

Many thanks to the anonymous reviewers of my article and to my colleagues.

References

[Alder, 1988] Mike Alder. Stochastic grammatical inference. Master's thesis, University of Western Australia, 1988.

[Harris, 1955] Zellig S. Harris. From phoneme to morpheme. *Language*, 31(2):190–222, 1955.

[Hutchens and Alder, 1998] Jason L. Hutchens and Michael D. Alder. Finding structure via compression. In David M. W. Powers, editor, *NeMLaP3/CoNLL98: Joint Conference on New Methods in Natural Language Processing and Computational Natural Language Processing Learning*, pages 79–82, Sydney, Australia, 1998. Association for Computational Linguistics.

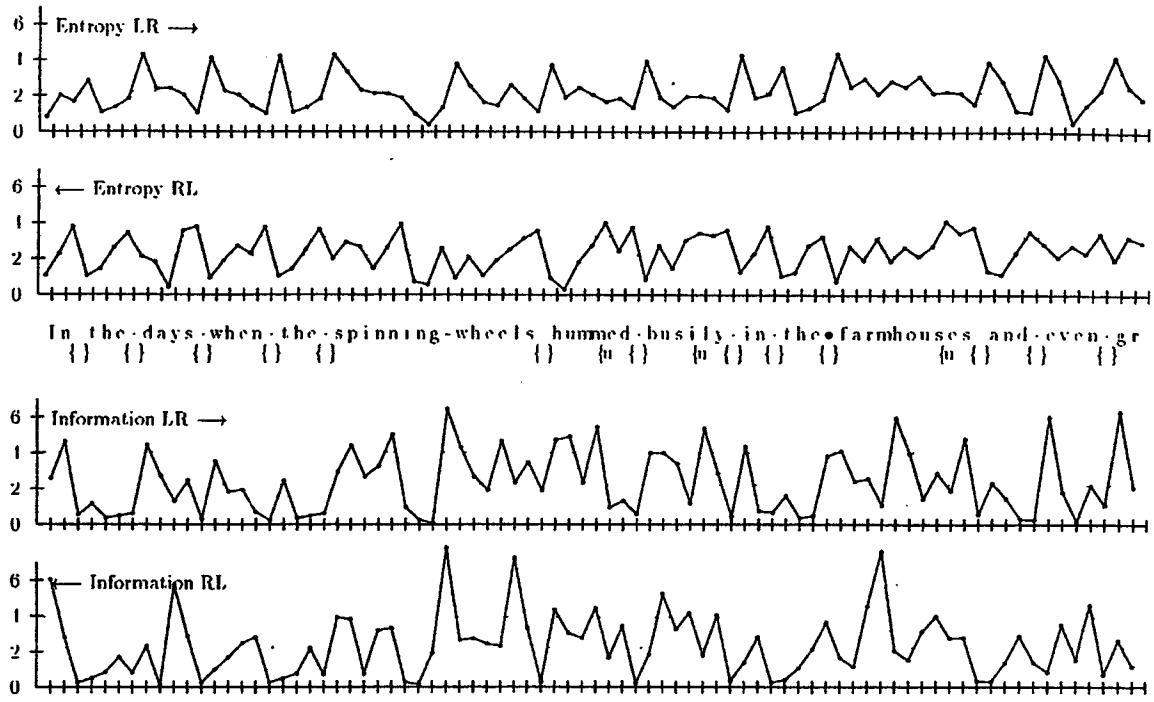

Figure 2: Entropy and information across a section of an English corpus

[Shannon and Weaver, 1949] Claude E. Shannon and Warren Weaver. *The Mathematical Theory of Communication.* University of Illinois Press, 1949.

[Wolff, 1977] J. G. Wolff. The discovery of segments in natural language. *British Journal of Psychology*, 68:97–106, 1977.