Largest chunks as short text segmentation strategy: a cross linguistic study

Copyright © 2017
Online Proceedings of UK-CLA Meetings
http://uk-cla.org.uk/proceedings
Vol 4: 273 – 292

LÁSZLÓ DRIENKÓ

adadad@freemail.hu

Abstract

Based on findings on short text segmentation our work aims to draw attention to a distributional dimension of speech segmentation. Using the algorithm of Drienkó (2016), we segment CHILDES texts in four languages: English, Hungarian, Mandarin, and Spanish. The algorithm looks for subsequent largest chunks that occur at least twice in the text. Then adjacent fragments below an arbitrary length bound k are merged. By assigning various values to k, we get a picture of how precision values change as chunks grow longer. Our results suggest that looking for largest recurring chunks may be a powerful cognitive strategy cross-linguistically as well.

Key words: segmentation, computational modelling, cognitive strategy

1. Introduction

The problem of how to segment continuous speech into components dates back at least to Harris (1955). Harris used "successor frequencies", i.e. statistics, to predict boundaries between linguistic units. Saffran et al. (1996), using syllable-based artificial languages, demonstrated that statistical information is indeed available for infants acquiring language. Results in language acquisition research indicate that speech segmentation is affected by various lexical and sub-lexical linguistic cues (see e.g. Mattys et al. 2005). Computational models of speech segmentation typically seek to identify the computational mechanisms underlying children's capacity to segment continuous speech (see Brent, 1999 for a review). Peters (1983) outlines an integrated theory of language acquisition where the learner uses various cognitive heuristics to extract large chunks from the speech stream and the 'ultimate' units of language are formed by segmenting and fusioning the relevant chunks. The philosophy behind our boundary inference algorithm is, broadly speaking, similar in that we first identify "large" utterance fragments in unsegmented texts, i.e. character sequences, and then apply 'fusion' – 'merging', in our terminology – to see how precision changes.

Drienkó (2016) proposed an algorithm for inferring boundaries of utterance fragments in relatively small unsegmented texts. The algorithm looks for subsequent largest chunks that occur at least twice in the text. Then adjacent fragments below an arbitrary length bound are merged. The author experimented with three types of English text: mother-child utterances from the CHILDES database, excerpts from Gulliver's travels by Jonathan Swift, and Now We Are Six, a children’s poem by A. A. Milne. The results were interpreted in terms of four precision metrics: Inference Precision, Alignment Precision, Redundancy, and Boundary Variability. It was found that Inference Precision grows with merge-length, whereas Alignment Precision decreases – i.e. the longer a segment is the more probable that its two boundaries are correct. Furthermore, Redundancy and Boundary Variability also decrease with the merge-length bound – i.e. the fewer boundaries we insert, the closer they are to the ideal boundaries.

The present study aims at checking the validity of the above findings in a cross-linguistic context. The largest-chunk segmentation algorithm will be applied to CHILDES utterances in four languages: English, Hungarian, Mandarin, and Spanish. After a short description of the algorithm in Section 2, we present our results in Section 3. This will be followed by a discussion and some conclusions in Sections 4 and 5, respectively.

2. Description of the algorithm

The basic, CHUNKER, module of our algorithm looks for largest character sequences that occur more than once in the text. Starting from the first character, it concatenates the subsequent characters and if a resultant string s_i only occurs in the text once, a boundary is inserted before its last character in the original text since the previous string, s_{i-1}, is the largest reoccurring one of the i strings. Thus the first boundary corresponds to s_{i-1}, our first tentative speech fragment. The search for the next fragment continues from the position after the last character of s_{i-1}, and so on.

The MERGE component of the algorithm concatenates fragments f_i and f_{i+1} if f_{i+1} consists of less than k characters. In other words, the boundary between f_i and f_{i+1} is deleted if f_{i+1} is shorter than k, an arbitrary merge-length bound. In our experiments we had $1 \leq k \leq 11$.

The EVALUATE module computes four precision metrics: Inference Precision, Alignment Precision, Redundancy, and Boundary Variability.

Inference Precision (IP) represents the proportion of correctly inferred boundaries (cib) to all inferred boundaries (aib), i.e. IP=cib/aib. The maximum value of IP is 1, even if more boundaries are inferred than all the correct (original) boundaries (acb).

Redundancy (R) is computed as the proportion of all the inferred boundaries to all the correct (original) boundaries, i.e. R = aib / acb. R is 1 if as many boundaries are inferred as there are boundaries in the original text, i.e. aib=acb, R is less than 1 if fewer boundaries are inferred than acb, and R is greater than 1 if more boundaries are inferred than optimal.

Alignment Precision (AP) is specified as the proportion of correctly inferred boundaries to all the original boundaries, i.e. AP = cib / acb. Naturally, the maximum value for AP is 1.

Boundary Variability (BV) designates the average distance (in characters) of an inferred boundary from the nearest correct boundary, i.e. $BV = (\Sigma df_i)/aib$. The above measures are not totally independent, since Inference Precision × Redundancy = Alignment Precision, but emphasise different aspects of the segmentation mechanism. Obviously, IP=AP for R=1.

The largest-chunk segmentation algorithm is outlined under (1).

(1)

The Largest-Chunk Segmentation Algorithm

```
1. CHUNKER
        input: segmented text T, unsegmented sequence UST of linguistic symbols (characters) of text T
          For each utterance position p in UST
                {fragment_candidate=""
                        while the occurrence of fragment_candidate in UST is > 1
                        {fragment_candidate = fragment_candidate + character_at_p}
                fragment_candidate → FRAGMENTS
                p → ALL INFERRED BOUNDARIES: p=p+1 }
2. MERGE
        input $k$
          For all fragments $f$ in FRAGMENTS
                {if length $f_{i+1} < k$ then concatenate $f_i$ and $f_{i+1}$ , $f_i = f_i + f_{i+1}$
                        modify ALL INFERRED BOUNDARIES accordingly}
                aib= the number of all the inferred boundaries
3. EVALUATE
        For all words $w$ in $T$
          {boundary_position of $w$ →ALL CORRECT BOUNDARIES }
          acb= the number of all correct boundaries

        For all boundaries in ALL INFERRED BOUNDARIES and ALL
```

For some immediate insight, (2) illustrates the action of the Largest Chunk algorithm on some simple character sequences. Spaces correspond to inferred boundaries. In (2b), for instance, the algorithm starts from the first ***a*** element, detects that ***a*** occurs twice in the sequence ***abcab***, so takes the next element, ***b***, detects that the corresponding segment, ***ab***, occurs twice, proceeds to consider segment ***abc***, detects that it occurs only once and infers a boundary after ***ab***, the first largest “chunk”. Segmentation then continues from element ***c***. Since ***c*** only has a single occurrence, a boundary should be inferred before it. However, this boundary has already been detected, so nothing happens and segmentation continues from the next position, ***a***. Again, dual occurrence of ***a*** is detected, so segment ***ab*** is considered. As ***ab*** occurs twice, the algorithm should step to the next element. However, since ***b*** is the last element of the sequence, a boundary is inferred after it. Note, that the inference of the last boundary is actually independent of the number of occurrences of the last segment.

(2)

a)	abcabc → abc abc
b)	abcab → ab c ab
c)	abc → a b c

To see how precision values are calculated consider the mini-corpus of utterances *{**baby is, baby it, baby bye**}*. Cf. (3). The corresponding *‘**baby is baby it baby bye**’* text contains six boundaries[1], thus acb=6. The Largest Chunk algorithm infers seven boundaries corresponding to fragments ***babyi, s, babyi, t, baby, by,*** and ***e***, which entails that aib=7. Four of the seven inferred boundaries are correct, cib=4, resulting in Inference Precision IP=cib/aib=4/7=0.57 and Alignment Precision AP=cib/acb=4/6=0.67. Redundancy is aib/acb=7/6=1.17. The second, fourth, fifth and seventh boundaries are correct, so their distance from the respective correct boundaries is zero, i.e. $df_2=df_4=df_5=df_7=0$. If we shift the first inferred boundary 1 character to the left, we reach the first correct boundary, following ***baby***. If we shift the first boundary 1 character to the right, we reach the second correct boundary, following ***is***. Clearly, then, $df_1=1$. Similarly, $df_3=1$, as well, since by shifting the third inferred boundary 1 character either to the left or to the right, we reach the third or the fourth correct boundary, respectively. When the distance of an inferred boundary is different for the left-side correct boundary and the right-side correct boundary, the shorter distance is chosen. Thus $df_6=1$, since the sixth inferred boundary, dividing ***bye*** into two, is 2 characters away from the first correct left-side boundary, which precedes ***bye***, and 1 character away from the first correct right-side boundary, actually the last one, so the right-side distance is chosen. We compute Boundary Variability as $BV=(df_1+df_2+df_3+df_4+df_5+df_6+df_7)/aib = 3/7=0.43$.

[1] Note that we choose to take the starting point of the text for granted, so ’position 0’ is not counted as a boundary. In contrast, the last boundary is considered to require inference. Statistically, this distinction is immaterial, however, it may be more sensible to say that AP approaches 1/n than 2/n or 0/n. Cf. Section 4. More importantly, if we do not regard the last boundary as inferred, IP drops to 0/0 in the limit, instead of 1/1 suggested by the data.

(3)

baby is baby it baby bye – 6 boundaries, acb = 6

babyisbabyitbabybye →babyi s babyi t baby by e

4 correct of 7 inferred boundaries: cib=4, aib=7,

IP=cib/aib=4/7=0.57

4 correctly identified boundaries: AP = cib /acb =4/6=0.67

R=aib/acb=7/6=1.17

BV=(1+0+1+0+0+1+0)/7=0.43

The Merge-mechanism is illustrated under (4). The k=1 case is actually the initial case, i.e. it involves no merging. When k=2, all fragments shorter than two characters, i.e. all one-character fragments, are added to the previous fragments. In our example, ***s***, ***t***, and ***e*** are added to ***babyi, babyi,*** and ***by,*** respectively. The resultant segmentation is ***babyis babyit baby bye,*** where all the four inferred boundaries are correct, IP=1, amounting to four correctly identified boundaries of the original six, AP=4/6=0.67. Since there are four inferred boundaries, Redundancy is 4/6=0.67, too. Boundary Variability is 0 because all boundaries are correct. When k=3, fragments at most two characters long are merged into preceding fragments: first **s** is added on ***babyi,*** then ***t*** is added on the second ***babyi,*** then ***by*** is added on ***baby,*** and finally **e** is added on the resultant ***babyby***. The output of the merging process is ***babyis babyit babybye*** with precision values IP=3/3=1, AP=3/6=0.5, R=3/6=0.5, and BV=0. The situation does not change for k=4 owing to a lack of any three-character-long fragment in the initial segmentation. When k=5, only one inferred medial boundary survives since the four-character-long ***baby*** merges with the preceding ***babyit*** forming ***babyitbaby,*** onto which ***by*** and then ***e*** are added to produce ***babyitbabybye***. The maximum value for k in the current example is 6 due to the fact that there is no fragment longer than five characters in the initial segmentation.

(4)

k=1	babyisbabyitbabybye →babyi s babyi t baby by e
	acb=6, cib=4, aib=7
	IP=4/7=0.57, AP=4/6=0.67, R =7/6=1.17, BV=0.43
k=2	babyi s babyi t baby by e →babyis babyit baby bye
	acb=6, cib=4, aib=4,
	IP=4/4=1, AP=4/6=0.67, R=4/6=0.67, BV=0

k=3 babyi s babyi t baby by e →babyis babyit babybye
acb=6, cib=3, aib=3,
IP=3/3=1, AP=3/6=0.5, R=3/6=0.5, BV=0
k=4 (same as k=3)
babyi s babyi t baby by e →babyis babyit babybye
acb=6, cib=3, aib=3,
IP=3/3=1, AP=3/6=0.5, R=3/6=0.5, BV=0
k=5 babyi s babyi t baby by e →babyis babyitbabybye
acb=6,cib=2, aib=2,
IP=2/2=1, AP=2/6=0.33, R=2/6=0.33, BV=0
k=6 babyi s babyi t baby by e →babyisbabyitbabybye
acb=6, cib=1, aib=1,
IP=1/1=1, AP=1/6=0.17, R=1/6=0.17, BV=0

3. The experiments

In our experiments we used data from the CHILDES database (MacWhinney, 2000). All files were converted to simple text format, annotations were removed together with punctuation symbols and spaces. Mother and child utterances were not separated, so the dataset for each language constituted an unsegmented (written) stream of 'mother-child language' represented as a single sequence of characters. The length range of the unsegmented texts was 3756 to 31843 characters.

3.1 Experiment 1 – English

In this experiment the first Anne file, anne01a.xml, of the Manchester corpus, (Theakston et al., 2001) was analysed. The original text consisted of 1815 word tokens and the average word length was 3.75 characters. The unsegmented version of the text consisted of 6801 characters. Initially, k=1, the CHUNKER module of our algorithm inserted 1129 boundaries, i.e. 1129 segments were identified with average segment length 6.02 characters. This means that the inferred fragments were, on average, 2.27 characters longer than the average word length for the original text. The precision values were as follows: Inference Precision = 0.66, Redundancy = 0.62, Alignment Precision = 0.41, Boundary Variability = 0.53. In the second part of the experiment we let the merge-length bound k change from 2 to 11. For instance, k = 3 means that, given the segmentation as provided by the CHUNKER module (the k = 1 case, with no merging), fragment f_{i+1} is glued to the end of f_i if f_{i+1} consists of less than 3 characters, i.e. if f_{i+1} is one- or two-character-long. That is, the maximum merge-length is 2 for k=3. Figure 1 shows how the number of inferred boundaries changes with the maximum merge-length. Figure 2 plots how the precision values change. For k=11, for instance, the values were IP=0.78, R=0.07, AP=0.05, BV= 0.27, and 121 boundaries were inserted amounting to 56.2 as average segment length.

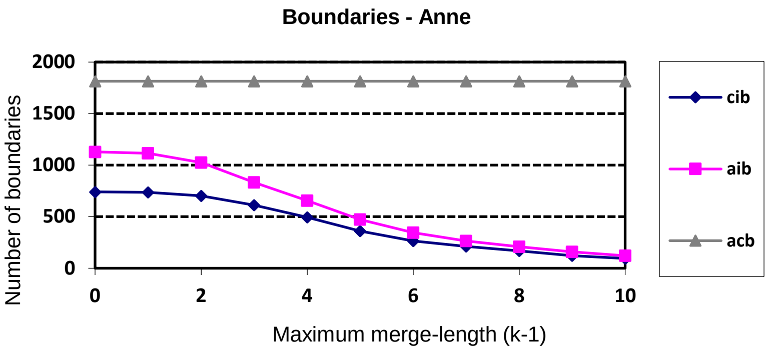


Figure 1. Number of boundaries as function of maximum merge-length – Anne (cib: correctly inferred boundaries, aib: all inferred boundaries, acb: all correct boundaries. See Appendix I for explicit numerical values).

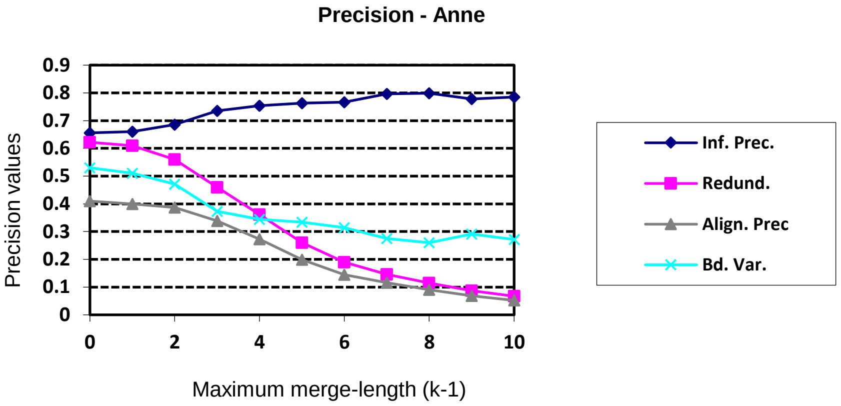


Figure 2. Precision values changing with maximum merge-length – Anne. (See Appendix I for explicit numerical values.)

3.2 Experiment 2 – Hungarian

The Hungarian data used in this experiment correspond with the miki01.xml file of the Réger corpus (Réger 1986, Babarczy 2006). The original text consisted of 1541 word tokens and the average word length was 4.41 characters. The unsegmented version of the text consisted of 6796 characters. The CHUNKER module inserted 1271 boundaries. The average segment length was 5.35 characters, which is about 1 character longer than the 4.41 average word length for the original text. The precision values were the following: IP = 0.53, R = 0.82, AP = 0.44, BV = 0.85. Figure 3 shows how the number of inferred boundaries changes with the maximum merge-length. Figure 4 plots how the precision values change. For k=11, the values were IP=0.78, R=0.08, AP=0.06, BV= 0.28, and 117 boundaries were inserted resulting in a 58.1 value for average segment length.

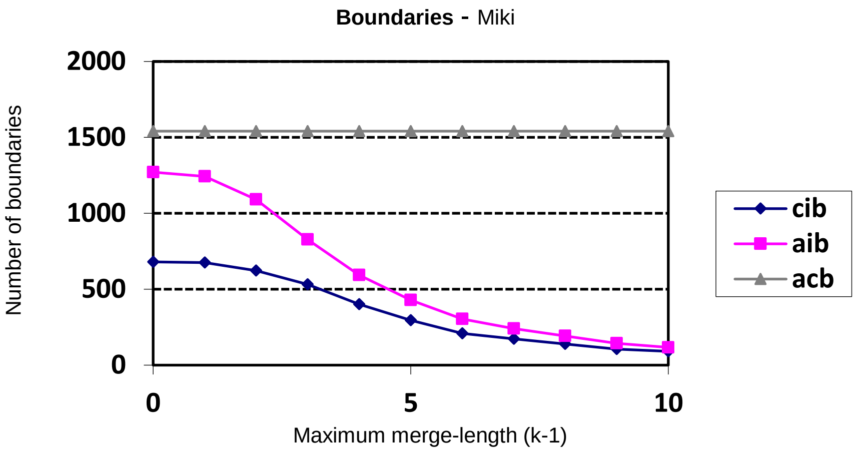


Figure 3. Number of boundaries as function of maximum merge-length – Miki. (cib: correctly inferred boundaries, aib: all inferred boundaries, acb: all correct boundaries. See Appendix I for explicit numerical values.)

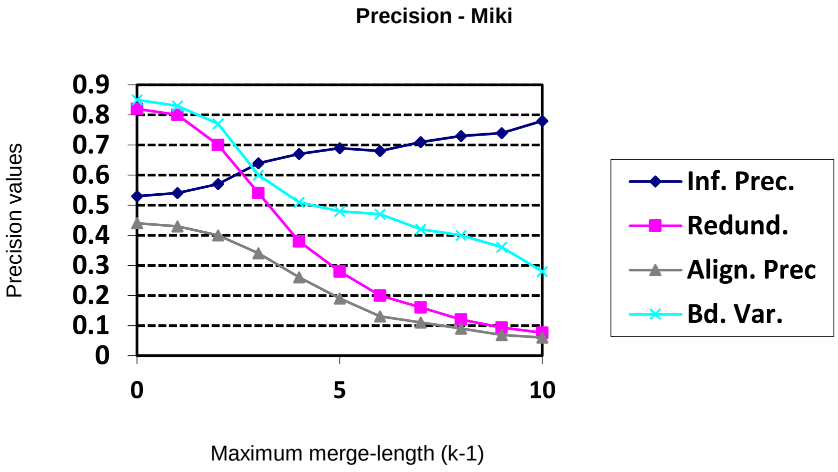


Figure 4. Precision values changing with maximum merge-length – *Miki*. (See *Appendix I* for explicit numerical values)

3.3 Experiment 3 – Mandarin Chinese

In this experiment we segmented Mandarin Chinese text included as bb1.xml in the Beijing corpus (Tardif 1993, 1996). The file contains the pinyin transcription of the utterances. The original text consisted of 7065 word tokens and the average word length was 4.51 characters. The unsegmented text consisted of 31843 characters. The CHUNKER module inserted 4359 boundaries. The average segment length was 7.03 characters which is 2.52

characters longer than the average word length for the original text. The precision values were the following: IP=0.6, R = 0.62, AP = 0.37, BV = 0.65. Figure 5 shows how the number of inferred boundaries changes with the merge-length bound. Figure 6 plots how the precision values change. For k=11, the values were IP=0.72, R=0.1, AP=0.08, BV= 0.42, and 750 boundaries were inserted yielding 42.5 as average segment length.

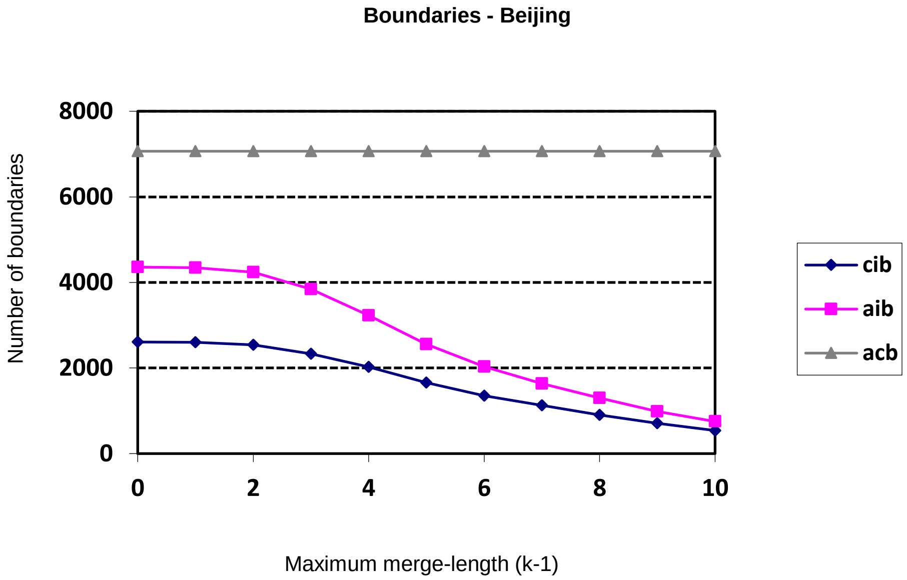


Figure 5. Number of boundaries as function of maximum merge-length – Beijing. (cib: correctly inferred boundaries, aib: all inferred boundaries, acb: all correct boundaries. See Appendix I for explicit numerical values).

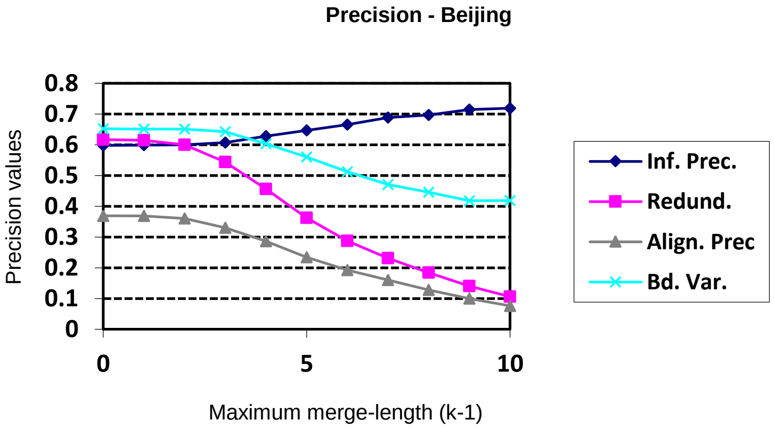


Figure 6. Precision values changing with maximum merge-length – Beijing. (See Appendix I for explicit numerical values.)

3.4 Experiment 4 – Spanish

The Spanish data for this segmentation experiment came from the Koki material contained in the 01jul80.cha file of the Montes corpus (Montes 1987, 1992). The original text contained 957 word tokens and the average word length was 3.92 characters. The unsegmented text consisted of 3756 characters. The CHUNKER module inserted 624 boundaries. The average segment length was 6.02 characters, which is 2.1 characters longer than the average word length for the original text. The precision values were the following: IP = 0.64, R = 0.65, AP=0.42, BV = 0.52. Figure 7 shows how the number of inferred boundaries changes with the maximum merge-length. Figure 8 plots how the precision values change. For k=11, the values were IP=0.79, R=0.07, AP=0.05, BV=0.27, and 63 boundaries were inserted, producing an average segment length of 59.6.

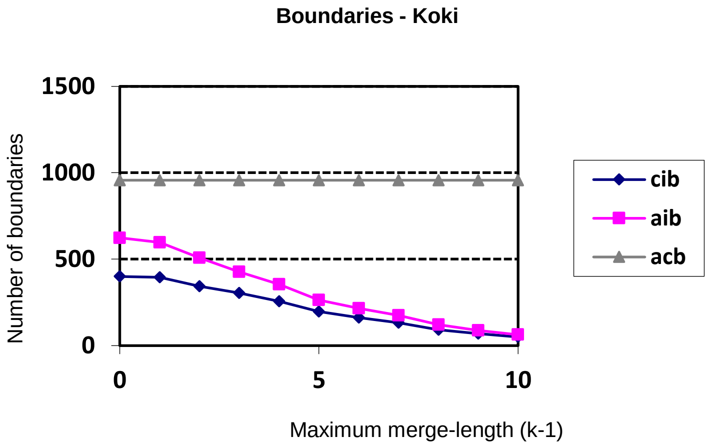
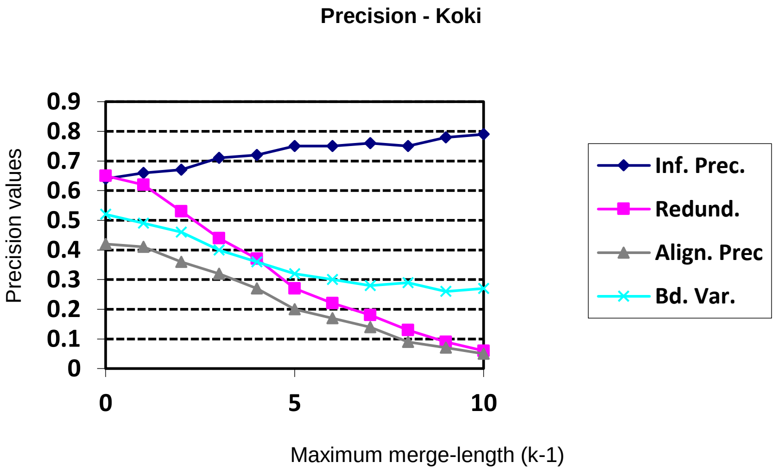


Figure 7. Number of boundaries as function of maximum merge-length – Koki. (cib: correctly inferred boundaries, aib: all inferred boundaries, acb: all correct boundaries. See Appendix I for explicit numerical values).



8. Precision values changing with maximum merge-length – Koki. (See Appendix I for explicit numerical values.)

4. Discussion

For all the texts that we looked at, the following pattern could be observed:

– Inference Precision (the proportion of correctly inferred boundaries of all inferred boundaries) grows (53-66% to 72-79%) with maximum merge-length (0 to 10), whereas Alignment Precision (the proportion of correctly identified boundaries of all the original, correct boundaries) decreases: i.e. the longer a segment is the more probable that its two boundaries are correct.

– Redundancy (the proportion of all the inferred boundaries to all the correct boundaries) and Boundary Variability (the average distance from the closest correct boundary) also decrease with the merge-length bound: i.e. the fewer boundaries we insert, the closer they are to the ideal boundaries.

Our data suggest that, as the merge-length bound grows, Inference Precision approaches 1, Boundary Variability 0, Redundancy and Alignment Precision 1/n, where n is the number of word tokens in the original text.

The k-merge behaviour of our precision metrics is similar to the previous findings for English data indicating that merging longer segments, and thus inserting fewer boundaries, yields higher Inference Precision. This fact supports our claim that looking for largest recurring chunks and then creating larger segments by merging may be a powerful cognitive strategy cross-linguistically as well. Furthermore, we have found that all BV values stay below 1 for all conditions, which means that, for a given Redundancy value, a learner could obtain an optimal segmentation – i.e. where all inferred boundaries are correct – by shifting the inferred boundaries less than 1 character, on average, to the right or to the left. In other words, language learning might be based on memorizing tentative chunks that could be “finalised” later, as cognitive development progresses.

The algorithm we use presupposes that linguistic data are presented in a “batch-like” fashion, i.e. as a single collection of information, and at a certain point in time. For large texts, this fact might render our computational mechanism not very plausible from a language acquisition point of view since language learners build linguistic representations not at a single time-point in their development but over longer periods of time. Recall, however, that we concentrate on relatively “short” texts, where the term “short text” could actually mean a single utterance. Assuming that building linguistic representations is not a single segmentation process but rather it is a series of segmentation processes, we can hypothesise further that small fragments of speech are constantly memorised and the largest-chunk seeking mechanism is applied to these small speech fragments stored in memory. The formation of such small speech fragments may be facilitated by the way language is presented to the learner: child-directed speech, poems, and songs can be of particular help. How the language presented to the child is divided into fragments or units can be shaped by discourse elements as well. For instance, the "variation sets" of partially overlapping subsequent utterances (Küntay & Slobin, 1996), or "topical discourse sequences", i.e. groups of adjacent utterances that centre around a shared topic (Rohde & Frank, 2014) may serve as a basis for organising the teacher's speech. Furthermore, the precision of the segmentation mechanism may be similar in the case of shorter speech fragments: Drienkó (2016) reports a 45% Inference Precision value for the children's poem Now We Are Six written by A. A. Milne consisting of 60 words, 186 characters. This is comparable with the 53-66% IP range we report in the present study.

We emphasise that the basic focus of our research was directed to the investigation of how the distributional texture of natural language may accord with a cognitive segmentation strategy – namely, looking for largest chunks – in the absence of any clues other than the linguistic information encoded in the distributional structure of the text. Regarding the initial Inference Precision values, as output by the CHUNKER module of the segmentation algorithm, as a measure of how well the largest-chunk strategy can be useful we may claim that it was able to find at least 53% (Hungarian data) of the correct boundaries for all the four languages examined. Most probably, it would not be too unrealistic to expect that the segmentation results reported in this study could be significantly enhanced by exploiting further sources of text-related information. As an immediate next step, one might think of taking utterance boundaries – corresponding to larger pauses in speech – for granted which could naturally result in a higher proportion of correctly detected boundaries.

The approach presented in this work is, broadly speaking, compatible with that of Peters (1983) in that we first identify "large" utterance fragments in unsegmented texts and then apply "fusion" – 'merging', in our terminology – to enhance precision via reducing Redundancy, which, in turn, may reduce processing efforts. By "fusioning" chunks, i.e. by merging initially established segments, we naturally lose information about the segmentation process. On the other hand, the data show that Inference Precision increases with k, the merge-length bound. These facts may echo the "less is more" notion in Newport (1990): less detail of utterance structure may facilitate higher precision of boundary inference. Storing tentative chunks in memory might agree with Bannard & Matthews (2008) claiming that children tend to store word sequences during language acquisition.

The utterance fragments that our algorithm can approximate are not necessarily individual words or grammatical syntactic phrases. Computationally, the possible strengths of our method lie, on the one hand, in its potential to provide empirical insights into the statistical structure of natural language i) on the basis of small texts ii) without previous training corpora or iii) explicit probability values. On the other hand, the utterance fragments detected by our algorithm can serve as input for subsequent segmenting mechanisms to break down text into ultimate components, practically, into words.

The data also suggest that the precision metrics we used in our analysis may be better suited for the largest-chunk segmentation process than the metrics of more traditional assessment methods in binary classification. For some comparison we provide Accuracy and F_1-score values in Appendix II. As can be seen there, Accuracy values are in the range 0.74 to 0.81 for each language and any k, whereas F_1-scores seem to imitate the decrease in AP with k. Thus the insights from our IP, R, and BV measures would be lost.

5. Conclusions and future work

The present paper investigated whether looking for largest chunks may be a useful segmentation strategy for four different languages. It was found that a relatively large proportion of speech segment boundaries can be detected by adopting the strategy in question and that the incorrectly inferred boundaries are statistically rather close to the correct ones. It was also shown that higher inference precision values can be obtained for longer segments formed by merging shorter ones. The preliminary speech fragments found by the segmentation algorithm can serve as input for later processing, which may have some relevance for language acquisition.

Our research could be broadened along several dimensions. Letters in a written text do not faithfully represent spoken language. There are phonemes that are denoted by two or three letters such as th, ph, ee, eau in the English words ‘there’, ‘phoneme’, ‘tree’, or ‘beauty’. The segmentation algorithm may actually choose to insert boundaries between the characters denoting such phonological units. Thus, segmenting phonologically transcribed texts would yield a clearer picture of the organisation of linguistic elements. In the case of Chinese we worked with pinyin transcripts where tone markers are represented as numbers 1 to 4 glued to the end of pinyin morphemes. Arguably, any inferred boundary immediately preceding a number would be erroneous. More precise results could be obtained by analysing texts written in Chinese characters. Recall that we did not make a distinction between morpheme-, word-, phrase-, or sentence boundaries. The question of how our results may be affected by establishing various boundary categories could also be an issue for future research. Further results might be obtained by directing investigation to a quantitative comparison of the largest chunk method with other possible segmentation strategies, or by analysing the segmentation properties of non-linguistic symbolic sequences via the approach advocated here.

References

Babarczy, A. (2006). The development of negation in Hungarian child language. Lingua 116, pp. 377-392.

Bannard, C., Matthews, D. (2008). Stored Word Sequences in Language. Learning. Psychological Science, 19(3), 241-248.

Brent, M. R. (1999). Speech segmentation and word discovery: a computational perspective. Trends Cognitive Sciences, 3(8), 294-301.

Drienkó, L. (2016). Discovering utterance fragment boundaries in small unsegmented texts. In Takács, A., Varga, V., and Vincze, V. (eds.) XII. Magyar Számítógépes Nyelvészeti Konferencia. (12th Hungarian Computational Linguistics Conference) pp. 273-281
ISBN: 978-963-306-450-4. http://rgai.inf.u-szeged.hu/mszny2016/

Harris, Z. S. (1955). From phoneme to morpheme. Language, 31, 190-222.

Küntay, A. & Slobin, D. I. (1996). Listening to a Turkish mother: Some puzzles for acquisition. In Slobin, D. I., Gerhardt, J., Kyratzis, A., Guo, J. (Ed.) Social interaction, social context, and language: Essays in honor of Susan Ervin-Tripp. (pp.265-286) Hillsdale, NJ: Lawrence Erlbaum Associates.

MacWhinney, B. (2000). The CHILDES Project: Tools for analyzing talk. 3rd Edition. Vol. 2: The Database. Mahwah, NJ: Lawrence Erlbaum Associates.

Mattys, S. L, White, L., Melhorn, J.F. (2005). Integration of multiple speech segmentation cues: a hierarchical framework. Journal of Experimental Psychology: General. 134(4), 477-500.

Montes, R. (1987). Secuencias de clarificación en conversaciones con ninos (Morphe 3-4): Universidad Autónoma de Puebla.

Montes, R. G. (1992). Achieving understanding: Repair mechanisms in mother–child conversations. Unpublished doctoral dissertation, Georgetown University.

Newport, E. L. (1990). Maturational constraints on language learning. Cognitive Science, 14, 11-28.

Peters, A. (1983). The units of language acquisition. Cambridge, Cambridge University Press.

Réger, Z. (1986). The functions of imitation in child language. Applied Psycholinguistics 7. 323–352.

Rohde H, Frank M. C. (2014). Markers of topical discourse in child-directed speech. Cogn Sci. Nov-Dec; 38(8):1634-61. doi: 10.1111/cogs.12121. Epub 2014 Apr 15.

Saffran, J. R., Aslin, R. N., Newport, E.L. (1996). Statistical learning by 8-month-old infants. Science. 274(5294), 1926-8.

Tardif, T. (1993). Adult-to-child speech and language acquisition in Mandarin Chinese. Unpublished doctoral dissertation, Yale University.

Tardif, T. (1996). Nouns are not always learned before verbs: Evidence from Mandarin speakers' early vocabularies. Developmental Psychology, 32, 492-504.

Theakston, A. L., Lieven, E. V., Pine, J. M., Rowland, C. F. (2001). The role of performance limitations in the acquisition of verb-argument structure: an alternative account. J. Child Lang. 28(1):127-52.

Appendix I: Numerical values for the graphs in the figures

(IP: Inference Precision, R: Redundancy, AP: Alignment Precision, BV: Boundary Variability, MML: Maximum Merge Length, cib: correctly inferred boundaries, aib: all inferred boundaries, acb: all correct boundaries).

1. English

Figure 1. Number of boundaries as function of maximum merge-length – Anne.

MML:	0	1	2	3	4	5	6	7	8	9	10
cib:	741	736	703	613	495	361	264	211	167	123	95
aib:	1129	1114	1024	834	656	473	344	265	209	158	121
acb:	1814	1814	1814	1814	1814	1814	1814	1814	1814	1814	1814

Figure 2. Precision values changing with maximum merge-length – Anne.

MML:	0	1	2	3	4	5	6	7	8	9	10
IP:	0.656	0.66	0.686	0.735	0.754	0.763	0.767	0.796	0.799	0.778	0.785
R:	0.622	0.61	0.56	0.46	0.362	0.26	0.19	0.146	0.115	0.087	0.067
AP:	0.41	0.4	0.387	0.338	0.273	0.199	0.145	0.116	0.091	0.068	0.052
BV:	0.53	0.51	0.471	0.373	0.344	0.334	0.314	0.275	0.26	0.291	0.272

2. Hungarian

Figure 3. Number of boundaries as function of maximum merge-length – Miki.

MML:	0	1	2	3	4	5	6	7	8	9	10
cib:	680	675	623	532	400	296	208	172	140	106	91
aib:	1271	1242	1090	827	593	428	304	241	191	144	117
acb:	1541	1541	1541	1541	1541	1541	1541	1541	1541	1541	1541

Figure 4. Precision values changing with maximum merge-length – Miki.

MML:	0	1	2	3	4	5	6	7	8	9	10
IP:	0.53	0.54	0.57	0.64	0.67	0.69	0.68	0.71	0.73	0.74	0.78
R:	0.82	0.8	0.7	0.54	0.38	0.28	0.2	0.16	0.12	0.093	0.076
AP:	0.44	0.43	0.4	0.34	0.26	0.19	0.13	0.11	0.09	0.069	0.06
BV:	0.85	0.83	0.77	0.6	0.51	0.48	0.47	0.42	0.4	0.36	0.28

3. Mandarin

Figure 5. Number of boundaries as function of maximum merge-length – Beijing.

MML:	0	1	2	3	4	5	6	7	8	9	10
cib:	2608	2603	2543	2333	2025	1656	1354	1127	905	707	539
aib:	4359	4346	4239	3846	3226	2558	2031	1636	1299	988	750
acb:	7066	7066	7066	7066	7066	7066	7066	7066	7066	7066	7066

Figure 6. Precision values changing with maximum merge-length – Beijing.

MML:	0	1	2	3	4	5	6	7	8	9	10
IP:	0.598	0.599	0.6	0.607	0.628	0.647	0.666	0.688	0.697	0.715	0.719
R:	0.617	0.615	0.6	0.544	0.456	0.362	0.287	0.231	0.184	0.14	0.106
AP:	0.369	0.368	0.36	0.33	0.286	0.234	0.192	0.16	0.128	0.1	0.076
BV:	0.652	0.651	0.651	0.643	0.603	0.56	0.512	0.471	0.446	0.418	0.419

4. Spanish

Figure 7. Number of boundaries as function of maximum merge-length – Koki.

MML:	0	1	2	3	4	5	6	7	8	9	10
cib:	400	395	343	305	257	197	162	133	91	69	50
aib:	624	598	509	426	354	264	216	175	121	88	63
acb:	957	957	957	957	957	957	957	957	957	957	957

Figure 8. Precision values changing with maximum merge-length – Koki.

MML:	0	1	2	3	4	5	6	7	8	9	10
IP:	0.64	0.66	0.67	0.71	0.72	0.75	0.75	0.76	0.75	0.78	0.79
R:	0.65	0.62	0.53	0.44	0.37	0.27	0.22	0.18	0.13	0.09	0.06
AP:	0.42	0.41	0.36	0.32	0.27	0.2	0.17	0.14	0.09	0.07	0.05
BV:	0.52	0.49	0.46	0.4	0.36	0.32	0.3	0.28	0.29	0.26	0.27

Appendix II: Accuracy and F$_1$-score

1. Accuracy

In binary classification Accuracy is calculated as (AII1), which translates into (AII2), given the parameters used in our analysis. Recall that cib denotes 'correctly inferred boundaries', aib stands for 'all inferred boundaries', and acb means 'all correct boundaries'. 'Character positions' equals the total number of characters for each text, i.e. it denotes 'all possible boundaries'. Cf. also Figure AII1. Table AII1 lists the values calculated.

(AII1)

$$\text{Accuracy} = \frac{\text{true positives} + \text{true negatives}}{\text{all exemplars}}$$

(AII2)

$$\text{Accuracy} = \frac{\text{cib} + [(\text{character positions} — \text{acb}) — (\text{aib} — \text{cib})\ ]}{\text{character positions}}$$

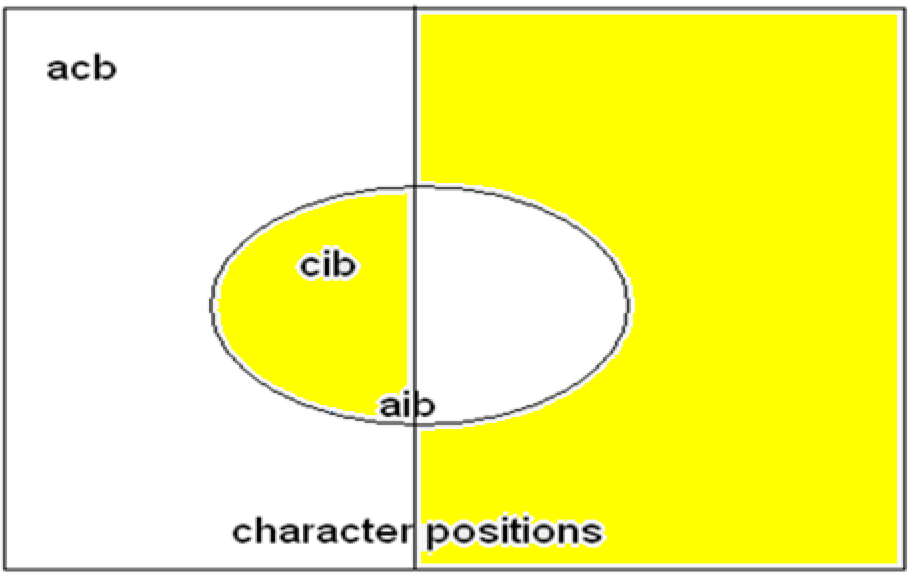


Figure AII1.Visualising classification measures. Accuracy equals the area of the shaded region divided by the area of the whole rectangle.

	English	Hungarian	Mandarin	Spanish
k	Accuracy	Accuracy	Accuracy	Accuracy
1	0.785	0.786	0.8050	0.792
2	0.786	0.789	0.8051	0.796
3	0.789	0.796	0.8047	0.792
4	0.791	0.808	0.8038	0.794
5	0.782	0.804	0.804	0.788
6	0.77	0.797	0.8018	0.778
7	0.76	0.79	0.7994	0.774
8	0.756	0.788	0.797	0.769
9	0.752	0.786	0.7941	0.761
10	0.746	0.7832	0.791	0.758
11	0.743	0.7828	0.788	0.755

Table AII1. Accuracy values for all experimental conditions.

2. F_1-score

F_1-score is the harmonic mean of Precision and Recall as given in (AII3). In our terminology Precision= cib/aib=IP, Recall=cib/acb=AP, which rewrites (AII3) as (AII4). The F_1-score values from the experiments are listed in Table AII2 and displayed graphically in Figure AII2.

(AII3)

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

(AII4)

$$F_1 = 2 \times \frac{\text{IP} \times \text{AP}}{\text{IP} + \text{AP}}$$

	English	Hungarian	Mandarin	Spanish
k	F_1 score	F_1 score	F_1 score	F_1 score
1	0.505	0.481	0.4564	0.507
2	0.498	0.479	0.4559	0.506
3	0.495	0.470	0.45	0.468
4	0.463	0.444	0.427	0.441
5	0.400	0.375	0.393	0.393
6	0.316	0.298	0.344	0.316
7	0.244	0.218	0.298	0.277
8	0.202	0.190	0.260	0.236
9	0.163	0.160	0.216	0.161
10	0.125	0.126	0.175	0.128
11	0.097	0.111	0.137	0.094

Table AII2. F_1-score values for all experimental conditions.

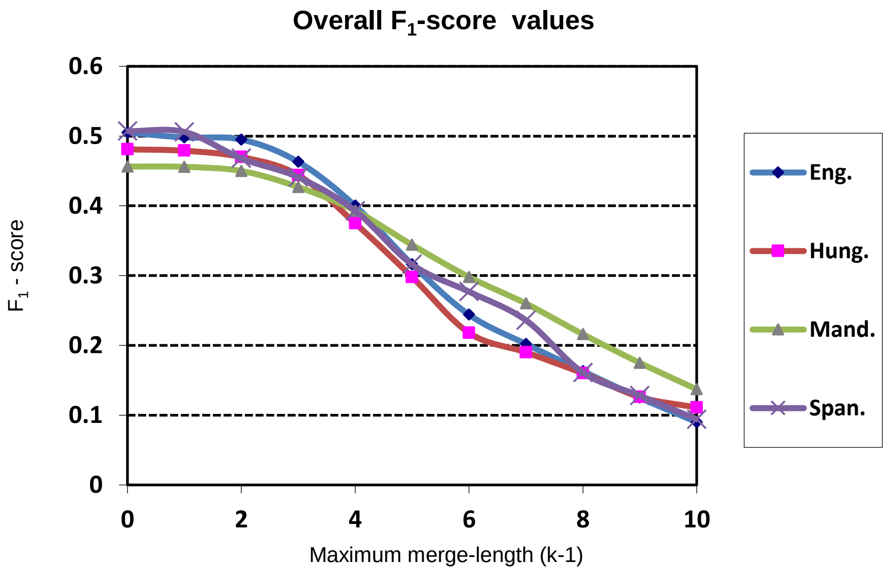


Figure AII2. F_1-score values across the four languages.