Natural Language Engineering **2** (4) 355–364. Printed in the United Kingdom

© 1996 Cambridge University Press

Finite state segmentation of discourse into clauses

EVA EJERHED

Department of Linguistics, University of Umeå, S-90187 Umeå Sweden
e-mail: ejerhed@ling.umu.se

(*Received 19 December 1996; Revised 9 February 1997*)

Abstract

The paper presents background and motivation for a processing model that segments discourse into units that are simple, non-nested clauses, prior to the recognition of clause internal phrasal constituents, and experimental results in support of this model. One set of results is derived from a statistical reanalysis of the Swedish empirical data in Strangert, Ejerhed and Huber 1993 concerning the linguistic structure of major prosodic units. The other set of results is derived from experiments in segmenting part of speech annotated Swedish text corpora into clauses, using a new clause segmentation algorithm. The clause segmented corpus data is taken from the Stockholm Umeå Corpus (SUC), 1 M words of Swedish texts from different genres, part of speech annotated by hand, and from the Umeå corpus DAGENS INDUSTRI 1993 (DI93), 5 M words of Swedish financial newspaper text, processed by fully automatic means consisting of tokenizing, lexical analysis, and probabilistic POS tagging. The results of these two experiments show that the proposed clause segmentation algorithm is 96% correct when applied to manually tagged text, and 91% correct when applied to probabilistically tagged text.

1 Introduction

The problem of recognizing simple clauses and segmenting discourse into such units is beginning to emerge as a problem area in its own right as evidenced by Abney 1990; Abney 1991. This is a problem area in which I have been interested since the early eighties Ejerhed 1982; Ejerhed 1988; Ejerhed 1990a; Ejerhed 1990b; Ejerhed 1993; Ejerhed and Church 1983. Simple clauses share with simple noun phrases the property of being nonrecursive Church 1988; Ramshaw and Marcus 1995, but the problems of defining, and recognizing, simple clauses are harder than those of defining and recognizing simple noun phrases.

However, the problem of clause segmentation of written and spoken discourse is worth addressing because of the many applications that successful clause segmentation algorithms could be used for. The following are some important application areas:

- speech synthesis: improved prosody in text to speech systems,
- speech recognition: automatic segmentation of input to speech recognizers,
- text analysis: preprocessing input to parsers for orthographic sentences,

https://doi.org/10.1017/S1351324997001629 Published online by Cambridge University Press

- machine translation: clauses as translation units,
- knowledge acquisition: databases of lexical preferences, SVO triplets, databases of facts, events, etc.

2 Clause segmentation of read speech

2.1 Background

The study presented in Strangert, Ejerhed and Huber 1993 is based on read speech. Four adult Swedish speakers, two women and two men, read the same text, which had a length of 878 words, excluding punctuation. The study combines the following analytic data contributed by the three authors: algorithmically derived information about the location of clause boundaries (C) in the text, based on Ejerhed 1990a; acoustically derived information about the location of intonation unit boundaries (I), and silent intervals (S), based on Huber 1988; perceptually derived information about perceived pauses (P) using two judges, based on Strangert 1990a; Strangert 1990b. The exact definitions of clause, intonation unit, silent interval and perceived pause that were used in analyzing the data of Strangert, Ejerhed and Huber 1993, and the details of the procedures whereby these units were identified are described in the respective references provided above.

Table 1 taken from Strangert, Ejerhed and Huber 1993 is a sample that shows the distribution of C, I, P, S for the four speakers. A look at the data in Table 1 invites a number of questions. In this presentation, I will concentrate on formulating a few questions that I find particularly interesting, and attempt to provide precise answers to them, using basic statistical concepts. First, I will deal with how the three acoustic/perceptual events I, P and S are related to each other, and show that they are not independent events. Second, I compare the relation of either of these three events to clause boundaries as opposed to phrase boundaries, showing that clause boundaries are better predictors of $I \vee P \vee S$ events than phrase boundaries are. Third, the agreement among the fours speakers in the location of $I \vee P \vee S$ events is investigated, and this agreement is found to be much greater than the agreement predicted by chance, despite the great differences between the four speakers in their individual speech styles with respect to prosodic segmentation.

2.2 Distribution of I, P and S

How surprising is the occurrence of the three events I, P, and S in the same position? A first answer to this question can be formulated by consulting the observed distributional data in Table 2, and the probabilities estimated from this data.

Counting the frequency of I events we get $p(I) = I/N = 583/3512 = 0.1660$, and similarly $p(P) = P/N = 617/3512 = 0.1756$ and $p(S) = S/N = 561/3512 = 0.1597$. If P, I, and S were independent, we would expect $p(I \wedge P \wedge S) \approx p(I) \cdot p(P) \cdot p(S) = 0.0046$, while in fact we find $p(I \wedge P \wedge S = 454/3512 = 0.1292$. The discrepancy is so overwhelming that we can be virtually certain that more traditional measures of association, such as chi-square, would show the same effect.

https://doi.org/10.1017/S1351324997001629 Published online by Cambridge University Press

Table 1. *Excerpt of data for all four speakers*

Text	F1	F2	M1	M2
I				
stadens				
fästning,	CIØØ	CIØØ	CIØØ	CØØØ
vars				ØØØS
normandiska				
torn				
ännu				
reser				
sig				
oanfrätt				
över				
den				
låga				
bebyggelsen		ØØPØ		
i				
dalen,	CIPS	CIPS	CIPS	CIPS
befann				
sig				
prins				
Edvard,	CIPS	CIPS	CIPS	CIPØ
kungens				
son	ØIPS	ØIPS	ØIPS	ØIPS
och				
känd				
som				
Långskank.	CIPS	CIPS	CIPS	CIPS
Edvard			ØØPS	ØØPØ
kommenderade	ØIØØ	ØIØØ	ØØPØ	
3.000				
beridna				
riddare			ØØPØ	
och				
soldater.	CIPS	CIPS	CIPS	CIPS
Baronernas				
här,	CIPS	CØØØ	CØPS	CØØØ
som				
leddes				
av				
Simon				
de				
Montfort,	CIPS	CIPS	CIPS	CØPØ
earl				
av				
Leicester,	CIPS	CIPS	CIPS	CIPØ
var				
lägrad				
i				
Fletching,	CIPS	CIPS	CIPS	CØPS

https://doi.org/10.1017/S1351324997001629 Published online by Cambridge University Press

Table 2. *IPS data*

Cases	Variables			Speakers F1	F2	M1	M2	Sums	Means
i	I	P	S	136	125	102	91	454	113.50
ii	I	P	$\overline{S}$	0	7	10	14	31	7.75
iii	I	$\overline{P}$	S	5	0	2	1	8	2
iv	I	$\overline{P}$	$\overline{S}$	28	13	32	17	90	22.50
v	$\overline{I}$	P	S	33	9	12	5	59	14.75
vi	$\overline{I}$	P	$\overline{S}$	22	24	17	10	73	18.25
vii	$\overline{I}$	$\overline{P}$	S	22	7	9	2	40	10
viii	$\overline{I}$	$\overline{P}$	$\overline{S}$	632	693	694	738	2757	689.25
				878	878	878	878	3512	878

Table 3. *Clause and phrase data*

Cases	Speakers F1	F2	M1	M2	Sums	Means
$(I \vee P \vee S)$	246	185	184	140	755	189
$(C \wedge (I \vee P \vee S)$	135	127	130	117	509	127
$(Ph \wedge (I \vee P \vee S)$	193	165	160	129	647	162

2.3 Distribution of $I \vee P \vee S$ in relation to clause boundaries and phrase boundaries

How surprising is the occurrence of either of the three events I, P, or S in the same position as a clause or phrase boundary? The text of length 878 words has 141 clause boundary locations. $141 \cdot 4$ speakers $= 564$. Thus, 564 events out of a total of 3512 events are C events, $p(C) = 564/3512 = 0.1605, p(C \wedge (I \vee P \vee S)) = 0.1449 >> p(C) \cdot p(I \vee P \vee S) = 0.0345$.

The text of length 878 words has 314 phrase boundary locations. $314 \cdot 4$ speakers $= 1256$. Thus, 1256 events out of a total of 3512 events are Ph events. $p(Ph) = 1256/3512 = 0.3576, p(Ph \wedge (I \vee P \vee S)) = 0.1842 >> p(Ph) \cdot p(I \vee P \vee S) = 0.0768$.

Based on the data and probabilities above for clauses and phrases, respectively, and on the definition of independence, we are entitled to the claims that C events and $I \vee P \vee S$ events are not independent, and Ph events and $I \vee P \vee S$ events are not independent.

Comparing clause boundaries and phrase boundaries with respect to their ability to predict $I \vee P \vee S$ events, we find that clause boundaries predict $I \vee P \vee S$ events better than Phrase boundaries do, as evidenced from the data in Table 3 and comparisons of precision and recall figures in Table 4.

https://doi.org/10.1017/S1351324997001629 Published online by Cambridge University Press

Table 4. *Comparison of precision and recall, clauses vs. phrases*

	Speakers				
	F1	F2	M1	M2	Means
Clause boundary					
Precision	96%	90%	92%	83%	90.25%
Recall	55%	69%	71%	84%	69.75%
Phrase boundary					
Precision	61%	53%	51%	41%	51.50%
Recall	78%	89%	87%	42%	86.50%

2.4 Agreement among speakers

How surprising is the occurrence of an $I \vee P \vee S$ event in the same position for all four speakers? Since the probability of an $I \vee P \vee S$ event is $p(I \vee P \vee S) = 755/3512 = 0.2149$, roughly 1 event in 5 is an $I \vee P \vee S$ event. If we assume that speakers produce $I \vee P \vee S$ events independently of each other, then the probability that all speakers produce an $I \vee P \vee S$ event at one and the same location is $0.2149^4 = 0.0021$. However, the observed agreement between the four speakers in the location of $I \vee P \vee S$ events, when examined, was found to be much greater than that predicted by chance: 100% agreement in 452/3512 events ($p = 0.1287$) and 75% agreement in 140/3512 events ($p = 0.0398$), adding up to 75-100% agreement in 592/3512 events ($p = 0.1686$), which effectively rules out the null hypothesis.

3 Clause segmentation of unrestricted text

3.1 Previous work

A recent clause parser that has attracted interest is Abney's parser, described in Abney 1990 as follows:

> 1. The Parser CASS [=Cascaded Analysis of Syntactic Structure] takes as its input the output of Church's POS (Part of Speech) program [Church 1988]. POS tags words with their part of speech, and also marks nonrecursive NP's (i.e. the segments of an NP from the first word to the head noun). CASS consists of three main filters:
>
> 1 The Chunk filter builds chunks. It corrects some common errors made by POS's NP-recognition component.
>
> 2 The Clause filter recognizes clauses. It identifies the beginning and end of simplex clauses, and marks subject and predicate. If it does not find a unique subject and predicate, it attempts error correction.
>
> 3 The Parse filter assembles chunks into complete parse trees. Its primary tasks are dealing with conjunction and attachment.

Constraint based grammar, Karlsson et al 1995, is also a framework in which the recognition of simple clauses plays an important role. In Scandinavia, the work of Paul Diderichsen Diderichsen 1946, and particularly his definitions of main and subordinate clauses in Danish are well known. His definitions can easily be expressed as regular expressions over parts of speech, but the essentially finite state character of his definitions have not, to my knowledge, been much noted in the literature.

https://doi.org/10.1017/S1351324997001629 Published online by Cambridge University Press

3.2 A new clause segmentation algorithm

The new clause segmentation algorithm that I want to propose here is described below in a way to facilitate comparison with Abney's parser. The clause segmenter takes as its input the output of the probabilistic tagger for Swedish by Åström, using the SUC tagset, which consists of 160 morphosyntactic tags. This tagger only tags single word tokens. It does not have a component that recognizes and marks nonrecursive NPs.

1. No identification of phrasal constituents, or correction of incorrectly marked nonrecursive NPs precedes clause segmentation.
2. The clause segmenter identifies the beginning of each simplex clause. This supports an end to end segmentation of a text into clause units, where a clause unit is defined as the sequence of words from the beginning of one simplex clause to the beginning of the next simplex clause. The identification of the end of each simplex clause is postponed to a later stage of processing.
3. A clause parser takes as input the output of the clause segmenter, and clause internal parsing and identification of the ends of simple clauses is followed by the assembly of clauses into complete parse trees for orthographic sentences.

The inspiration for the clause segmentation component of this model came from Salomaa 1981 (Section 6.1, Local and regular languages, pp. 96–97):

> Let Σ be an alphabet, let A and B be subsets of Σ, and let C be a subset of Σ^2. Then, by Theorem 2.7 [= The family of representable languages is closed under Boolean operations], the language
>
> $$L = A\Sigma^* \cap \Sigma^* B \backslash \Sigma^* C \Sigma^* \tag{6.2}$$
>
> is regular. Languages L of the form (6.2), for some Σ, A, B, C are referred to as *local*.
>
> The term "local" originates from the fact that if L is a language of the form (6.2), then it suffices to scan an arbitrary word w locally to find out whether or not w is in L. ...
>
> Local languages are also referred to as "2-testable", reflecting the length of the local scans, i.e. the size of the hole. An analogous definition can be given for k-testable languages. Essentially, we are then considering a hole from which k adjacent squares can be seen.
>
> Local languages are closely related with finite deterministic automata, ..."

The new clause segmentation algorithm is based on the hypothesis that the language of simple clauses is k-testable in the sense defined by Salomaa. A more general and radical version of this idea is that what is k-testable makes a good processing unit in NLP.

Experiments were conducted in Umeå during the summer of 1996 with a clause recognition algorithm for unrestricted Swedish text by the author, with k=4, and with the assistance of Åström and Backman in the implementation (Åström: tokenizing, lexing, tagging Åström 1996, Backman: clause boundary insertion).

The basis for the clause segmentation rules formulated by the author was studies of bigrams and trigrams of parts of speech in the SUC corpus, combined with excerpts of tagged four word sequences from the SUC corpus, in order to test the clause segmentation rules before implementing them. The rules that were implemented will

https://doi.org/10.1017/S1351324997001629 Published online by Cambridge University Press

Table 5. *Descriptive data about SUC and DI93 texts*

	Text 1 – SUC kk09	Text 2 – DI93 DI930320
#tokens, incl. punctuations, block and paragraph	1251	1579
#clause units	219 (210 retrieved)	194 (178 retrieved)
#orthographic sentences	120	87
#finite verbs	192	176
avg. #tokens/sentence	1251/120 = 10	1579/87 = 18.15
avg. #tokens/clause	1251/219 = 5.7	1579/194 = 8.1
avg. #clauses/sentence	219/120 = 1.8	194/87 = 2.2
avg. #finite verbs/sentence	192/120 = 1.6	176/87 2.0

be presented in the fuller version of the paper to appear in the companion volume. We are aware of the fact that there are important cases of simple clauses that are not covered by the rules that were implemented. Those are cases where a finite verb is the only indicator of the beginning of a new clause, and where there are three or more words between this finite verb and the previous finite verb. Before stating rules for these cases, we wanted to collect more information about them. When evaluating the performance of the current implementation of the clause segmentation algorithm, the failure to recognize clause boundaries in all such cases was considered as an error.

4 Evaluation and results

To evaluate the performance of the clause segmentation algorithm and the implementation of it, two tagged texts were clause segmented, and portions of the output of comparable length were manually scored for correctness. The two texts were taken from the SUC corpus, and from the DI93 corpus respectively, and one relevant difference between the two texts is that the first text was manually tagged, whereas the second was automatically tagged. In a separate evaluation based on a test set of 19,608 tokens, the fully automatic tagger for DI93 was found to be 95.45% correct, where correctness was defined as agreement with manual disambiguation (between given alternative analyses) of the same text. Another relevant difference is that the two texts belong to very different genres. The SUC text is an excerpt from a novel (Stig Claesson, *Rosine*, Stockholm, Bonniers, 1991) and the DI93 text is financial newspaper text.

Descriptive data about the two texts are provided below, and those data are based only on the portions of the two texts that were scored. The length of the two texts, measured by their numbers of tokens, is not identical. The reason for this is that to have a roughly comparable number of clause units in the two texts that were scored, we needed to use more of text 2, because of its greater sentence length.

A closer analysis of the nature of the errors revealed that in the case of text 1, all nine errors were due to cases not covered by the current set of rules, and none were due to errors in the manual tagging. Future work will be directed to covering such

https://doi.org/10.1017/S1351324997001629 Published online by Cambridge University Press

Table 6. *Errors*

	Text 1	Text 2
Error rate	9/219 = 4.1%	18/194 = 9.2%
Clauses underrecognized	9	16
Clauses overrecognized	0	2
Wrong place clause boundary	0	0

Table 7. *Precision and recall*

	Text 1	Text 2
Precision	210/210 = 100%	176/178 = 98.8%
Recall	210/219 = 95.8%	176/194 = 90.7%

cases. In the case of text 2, only six of the 16 errors of underrecognition were due to cases not covered by the current rules, eight were due to tagging errors in the input to the clause segmenter, and two were due to a bug in the current implementation of the algorithm. Of the two errors of overrecognition in text 2, one was due to a tagging error, and the other was due to an error in the automatic, typographically driven paragraph segmentation that was used. This shows that improving the performance of the tagger would reduce errors in clause segmentation.

All in all, what is most striking in these results is that the number of cases that we knew would not be covered by the current set of clause rules represent such a small portion of the total number of clause units in actual empirical data, and that the majority of occurring cases are covered by these rules.

5 Conclusions

- Clause boundaries are better predictors than phrase boundaries for the occurrence of the acoustic and perceptual events I, S, P.
- The agreement between the four speakers in the location of $I \vee P \vee S$ events is much greater than chance, despite their widely different individual speech styles.
- Clause boundaries in unrestricted, tagged text can be recognized with great precision prior to any chunking of the text into constituents (nonrecursive NPs, PPs, etc.).
- Automatic clause segmentation at an early stage of processing can provide the basis for an incremental parser that parses a clause at a time and assembles the result into parse trees for complete orthographic sentences.
- Relations between words and phrases within the same clause are qualitatively different from relations between words and phrases in different clauses, and clause segmentation makes it possible to exploit these differences in improving the performance of natural language processors.

https://doi.org/10.1017/S1351324997001629 Published online by Cambridge University Press

- There are many open questions, even for a single language, concerning the definition of the clause units to have as targets for clause segmentation. The following are three important choices that come to mind. The approach to clause segmentation presented in this paper selects in each case the first option in the definition of the targeted clause units.
  - Only finite clauses or both finite and non-finite (infinitival and participial) clauses?
  - At most one finite verb per clause, or exactly one finite verb per clause?
  - Selective or indiscriminate use of punctuation?
- Clause definitions and clause segmentation rules, such as the ones presented in this paper, are highly language specific and new rules need to be written for each language and tested on large amounts of empirical corpus data.

References

Åström, M. 1996. A probabilistic tagger for Swedish using the SUC tagset. In Helmut Feldweg & Erhard W. Hinrichs (eds.), *Lexikon und Text*, pp. 245-256. Lexicographica Series Maior 73. Niemeyer. Tuebingen.

Abney, S.P. 1990. Rapid incremental parsing with repair. *Proceedings of the 6th New OED Conference* pp. 1-9. University of Waterloo, Ontario.

Abney, S.P. 1990. Parsing by chunks. In: S. Abney, R. Berwick & C. Tenny (eds.), *Principle-Based Parsing* pp. 257-278. Kluwer Academic Publishers. Dordrecht.

Church, K. 1988. A stochastic parts program and noun phrase parser for unrestricted text. *Proceedings of the 2nd Conference on Applied Natural Language Processing* pp. 136-143. ACL. Austin, Texas.

Diderichsen, P. 1946. *Elementaer Dansk Grammatik*. Gyldendal. Copenhagen.

Ejerhed, E. 1982. The processing of unbounded dependencies in Swedish. In E. Engdahl and E. Ejerhed (eds.), *Readings on Unbounded Dependencies in Scandinavian Languages*, pp. 99-149. Almqvist & Wiksell Intl. Stockholm.

Ejerhed, E. 1988. Finding clauses in unrestricted text by finitary and stochastic methods. *Proceedings of the 2nd Conference on Applied Natural Language Processing* pp. 219-227. ACL. Austin, Texas.

Ejerhed, E. 1990. A Swedish clause grammar and its implementation. In Rögnvaldsson & J. Pind. (eds.), *Papers from the Seventh Scandinavian Conference of Computational Linguistics*, pp. 14-29. Linguistic Institute, University of Iceland. Reykjavik.

Ejerhed, E. 1990. On corpora and lexica. In Jan-Ola Östman and Kristiina Jokinen (eds.), *SKY 1990 (The Linguistic Association of Finland, 1990 Yearbook)*, pp. 77-96. Helsinki.

Ejerhed, E. 1993. Nouveaux courants en analyse syntaxique. *t.a.l.(Traitement Automatique des Langues) 1993*, volume 1, pp. 61-82. ATALA, Paris.

Ejerhed, E. and Church, K. 1983. Finite State Parsing. In F. Karlsson (ed.): *Papers from the Seventh Scandinavian Conference of Linguistics*, pp. 410-432. Department of General Linguistics. Helsinki.

Huber, D. 1988. *Aspects of the Communicative Function of Voice in Text Intonation.* Department of Linguistics and Phonetics, University of Lund. Lund.

Karlsson, F. *et al* (eds.), 1995. *Constraint Grammar*. Mouton de Gruyter. Berlin and New York.

Ramshaw, L. and Marcus, M. 1995. Text chunking using transformation-based learning. In D. Yarowsky & K. Church. (eds.), *Proceedings of the Third Workshop on Very Large Corpora*, pp. 82-94. SIGDAT/ACL. Cambridge, Mass.

Salomaa, A. 1981. *Jewels of formal language theory*. Pitman Publ Ltd. London.

https://doi.org/10.1017/S1351324997001629 Published online by Cambridge University Press

Strangert, E. and Ejerhed, E. and Huber, D. 1993. Clause structure and prosodic segmentation. *Papers from the Seventh Swedish Phonetics Conference*. pp. 81-84. Uppsala University. Uppsala.

Strangert, E. 1990. Perceived Pauses, Silent Intervals and Syntactic Boundaries. *PHONUM*, volume 1, pp. 35-38. Department of Phonetics, University of Umeå. Umeå.

Strangert, E. 1990. Where do Pauses Occur in Texts Read Aloud? *Papers from the Twelfth Scandinavian Conference of Linguistics*, pp. 403-414. Linguistic Institute, University of Iceland, Reykavik.

https://doi.org/10.1017/S1351324997001629 Published online by Cambridge University Press