



LUND UNIVERSITY

Is the development of the English progressive in L1 children a U-shaped curve?

Carl-Staffan Svenbro

Carl-Staffan Svenbro

Autumn Term 2018/2019

Supervisor: Eva Klingvall

Centre for Languages and Literature

Lund University

Abstract

This essay is concerned with whether the progressive aspect is overgeneralized by children, using it with stative verbs. The study targets the age interval 1 to 10 years and adults. The matter was investigated in several child corpora and adult corpora. The interval 1 to 5 years was covered by the Brown (1973b) corpus with *Eve*, *Adam* and *Sarah*. Data for the interval 6 to 10 years were obtained from the Carterette and Jones corpus (1974b). Initially, a comprehensive list of verbs used in progressive constructions from the Brown (1973b) corpus was extracted using a *pattern-matching algorithm*. It recognized textual patterns typical of this aspect. Development and testing of such an algorithm is an additional objective of this study. The corpus tagging indicators of the child corpora were used to validate the algorithm. After selection of 5 stative verbs from the Brown (1973b) corpus, the two most frequent verbs, *have* and *hurt*, were examined more thoroughly. Analysis of the child data indicates that the verb *have* is not overgeneralized according to the criteria. On the basis of adjusted frequencies, the study finds tentative support for overgeneralization of the verb *hurt* in the progressive. No indications of other stative verbs being overgeneralized are identified.

Table of Contents

Abstract	2
1. Introduction.....	5
2. Background	7
2.1 Perfective vs. Imperfective aspect: a crosslinguistic perspective	7
2.2 The progressive.....	10
2.3 Stative verbs	11
2.4 Children’s language development	14
2.5 Overgeneralization.....	16
2.6 Earlier studies of the progressive.....	17
3. Methods and Materials.....	19
3.1 Corpus overview	19
3.2 The Computer Application	21
3.3 Application Output	23
3.4 The Algorithm	24
3.5 Selection of stative verbs	25
3.6 Alternative approaches	29
4. Results.....	30
4.1 Validation of the algorithm.....	30
4.2 General trends	31
4.3 Selected stative verbs.....	33
5. Discussion.....	38
5.1 Evaluation of the algorithm	38
5.2 General developments of the progressive.....	39
5.3 Developments of the progressive for stative verbs.....	40
5.4 Conclusion	45
6. Future developments.....	47
References.....	48
Appendix.....	52
Appendix 1: The CHILDES framework.....	52

Appendix 2: Age group distribution	53
Appendix 3: A non-linguistic example of U-Shaped Development	55
Appendix 4: Computer Application Window	56
Appendix 5: Computer Application Features	58
Appendix 6: Computer Application Output (Excel).....	59
Appendix 7: Computer Application Versions	62
Appendix 8: Algorithm, simple flowchart.....	63
Appendix 9: Algorithm, advanced flowchart	64
Appendix 10: Algorithm, validation, p-value	66
Appendix 11: The COCA corpus - queries, interface etc.	67
Appendix 12: Samples from the COCA corpus.....	69
Appendix 13: Data from the child corpora	73
Appendix 14: Output of verb lists.....	76
Appendix 15: Progressive frequency tables.....	79
Appendix 16: The VBA object model	82
Appendix 17: Source code description	83
Appendix 18: Source code	86

1. Introduction

Children's language development is analogous to the process of learning how to go up a ladder, as suggested by Bochner and Jones (2003, pp. 212-214). Small babies can produce sounds that resemble vowels, like "'ah' and 'oo'", but these sounds are not associated with any meaning or intention (Hardman¹, 2003, pp. 131-132). On the next step of the ladder, children start imitating and associating situations with sounds, but there are still no words, not even invented ones (pp. 16-17). The first real words (e.g. "Mum") are preceded by made-up words that express intention (pp. 16, 20). Subsequently, children become able to put the uninflected words together to form basic sentences, like "'cat go there'" or "'Daddy car'" (pp. 16, 21-22). Inflectional morphology, e.g. third person singular *-s*, represents the most difficult step of the language acquisition ladder (p. 214).

The ladder comparison seems convenient when describing children's language development, one step at the time. However, although many features are gradually learned, this is not always true. Cases related to rules with exceptions often tend to exhibit patterns of *U-shaped development*, during which children are learning but regressing. It has been observed for example in plural forms and past tense of irregular verbs how children appear to make mistakes as they become older. Such errors are typically caused by *overgeneralization*². For instance, the main rule for the plural form of a noun is normally created by adding *-s*. For some nouns this is not the case. For example, the plural form of *mouse* is *mice*, which is created without adding *-s*. In order to overgeneralize the plural form of a word (e.g. **mouses*), children must be able to recognize the general pattern for plural inflection (cf. Bowerman, 1982, p. 115; see also Section 2.5).

In this paper, I will investigate whether children overgeneralize the progressive by using stative verbs in this aspect. A tentative hypothesis is that such overgeneralization is commonplace. If it exists, a related question is whether it is connected with a frequency peak in the progressive aspect in general. In English, this kind of overgeneralization seems plausible as the progressive aspect resembles a rule with exceptions. The second aim of this essay is methodological. I will examine

¹ Hardman (2003) has authored one chapter in the book by Bochner and Jones (2003).

² Bowerman (1982, p. 115) also uses the term "overregularization". For more source citations, see Section 2.5.

if identification of progressive syntax can be done with an automatic, pattern-matching algorithm³. Such an algorithm would be capable of analyzing untagged corpora. The paper is based on data covering 1-10 years of age and an adult control group. The child corpora used for the study are *Eve, Adam* and *Sarah* (Brown, 1973b) and *Carterette and Jones* (1974b). Control corpora, with mainly adult language, are *Santa Barbara* (Du Bois et al., 2000-2005a) and the spoken section of *COCA* (Davies, 2008-).

³ In this essay, the pattern-matching algorithm is sometimes referred to as *the progressive algorithm* or simply as *the algorithm*.

2. Background

2.1 Perfective vs. Imperfective aspect: a crosslinguistic perspective

Imperfective aspect describes a situation from an inside perspective. It is often contrasted with the perfective aspect. The latter views an event as from the outside, without focusing on the internal properties (Huddleston and Pullum, 2002, p. 52; Comrie, 1976, p. 4). According to Saeed (2016), the perfective, unlike the imperfective, “focuses on the end points of a situation” (p. 131; see also Comrie, 1976, p. 19).

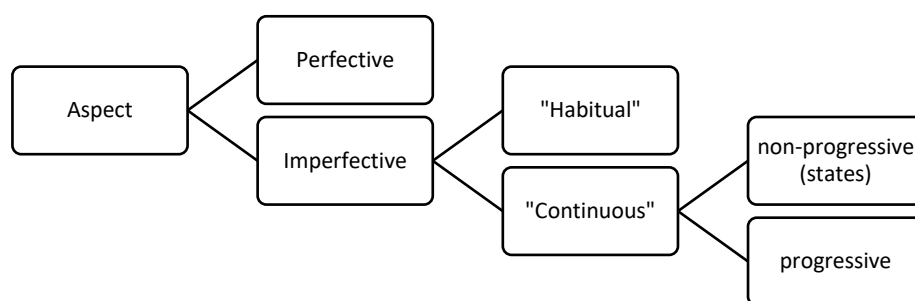
Authors like Huddleston and Pullum (2002, p. 124) and Comrie (1976, p. 12) distinguish between ‘perfective’ and ‘perfect’. Still, they observe that some authors use the terms interchangeably. As explained in the previous paragraph, ‘perfective’ describes a situation viewed as a whole. On the other hand, the term ‘perfect’ denotes past events with present relevance, as noted by Comrie. In this essay, a clear distinction between these two terms is made, in conformity with the authors cited above.

The imperfective aspect has two subcategories, “Habitual” and “Continuous”, according to Comrie (1976). These categories are mutually exclusive, which means that if one of them is true, the other must be false and vice versa (pp. 25-26). Habituality is not necessarily the same as being repetitive although this may sometimes be the case⁴ (pp. 27-28). For instance, Comrie (1976) observes that there are no repetitions in the sentences “*the Temple of Diana used to stand at Ephesus*” or “*Simon used to believe in ghosts*” (p. 27). Still those sentences are habitual according to Comrie’s classification. He claims that habituality is not characterized by repetition alone but also depends on whether the situation is extended in time enough not to be regarded as incidental (pp. 27-28).

⁴ For instance, the event described by the sentence “*the old professor used always to arrive late*” is habitual and repetitive/iterative (Comrie, 1976, p. 28).

Comrie (1976) finds it convenient to define habituality first and then “continuous” as not habitual but still imperfective (p. 26). *Continuous* can be either non-progressive or progressive. States are continuous and normally occur without the progressive aspect (cf. Comrie, 1976, pp. 25, 35, 51). An aspectual classification system, adapted from Comrie, is illustrated in Figure 1 below.

Figure 1. Aspectual classification for English (adaptation from Comrie, 1976, p. 25 [table 1], with *states* within parentheses added to the *non-progressive* category⁵)



An illustrative example of a language that, unlike English, shows clear distinction between perfective and imperfective aspect is Russian (cf. Comrie, 1976, p. 7; Huddleston and Pullum, 2002, p. 124). While English uses an auxiliary verb combined with the *-ing* ending to indicate imperfective *progressive aspect* (see next section), Russian marks aspect by using verb prefixes (Saeed, 2016, p. 131). Table 1 below contrasts Russian and English in terms of aspect.

⁵ As inferred from Comrie (1976, pp. 35, 51), states are generally continuous and stative verbs typically do not occur in the progressive. Furthermore, Comrie (1976, pp. 3-4 [footnote 3]) suggests that the progressive corresponds to the imperfective if habitual sense and stative verbs are disregarded. From this may be concluded that the 'non-progressive' in Figure 1 essentially represents states expressed with stative verbs.

Table 1. Some differences and similarities between Russian and English aspectual markings.

Features		Russian	English
Perfective / Imperfective	Clear distinction between perfective and imperfective in grammar	Yes (Huddleston and Pullum, 2002, p. 124)	No (Comrie, 1976, pp. 3-4 footnote; Huddleston and Pullum, 2002, p. 124)
Perfective	Associated semantic marking	Yes (Comrie, 1976, p. 19)	No
	Occurs both in past and non-past tense	Yes (Saeed, 2016, p. 132)	Yes ⁶
Imperfective	Verbal marking	Yes, as verb prefixes (Saeed, 2016, p. 131)	Yes, inflectional endings/suffixes combined with auxiliary verbs (Saeed, 2016, p. 131).

⁶ E.g. “I’ll write a letter” (Saeed, 2016, p. 132).

2.2 The progressive

As shown above, the progressive is a subcategory to the imperfective aspect in English. It is constructed by a form of the auxiliary verb *be* followed by the *-ing* form of another verb, the latter of which is often referred to as *present participle* (cf. Huddleston and Pullum, 2002, p. 103, 124, 1222; Biber et al. 1999, p. 460). Examples are *She is writing a book* and *I was just reading the newspaper*.

While the function of the progressive essentially is to mark imperfective aspect (cf. Huddleston and Pullum, 2002, p. 52), there are exceptions. For example, this aspect used with a form of *be* in present tense, may indicate future events (Biber et al., 1990, p. 470). Huddleston and Pullum (2002, p. 171) term the progressive with future reading *progressive futurate*, as exemplified below in (1). According to them, the associated form in (1) may indicate either a scheduled plan to phone the person or simply an intention to call her at an unspecified time in the evening. In contrast, the non-progressive counterpart in (2) should be understood as a scheduled event. As inferred⁷ from Huddleston and Pullum (2002, p. 163, 171), (1) does not express imperfective aspect. In effect, it stands to reason that it is in the perfective, despite the progressive form.

(1) "*I'm phoning her tonight.*" (Huddleston and Pullum, 2002, p. 171)

(2) "*I phone her tonight*" (Huddleston and Pullum, 2002, p. 171)

Another example of an ambiguous sentence in which the progressive does not necessarily carry imperfective meaning is (3) below. In that sentence, the progressive form is preceded by the modal auxiliary verb *will*. Huddleston and Pullum (2002, pp. 171-172) observe that (3) has two readings. It *could* be construed imperfectively, with the meaning that the speaker among others will already be in the process of flying to Bonn when the meeting ends. Another possible interpretation is

⁷ This inference has been made based on Huddleston and Pullum (2002, p. 163), first paragraph under "(b) Imperfectivity" and Huddleston and Pullum (2002, p. 171), first paragraph under "8.3 Non-aspectual uses of the progressive".

associated with a decision to fly to Bonn when the meeting ends. The second reading is perfective (p. 172), as it views the flight to Bonn as a whole event.

(3) “*When the meeting ends we’ll be flying to Bonn*” (p. 171)

2.3 Stative verbs

While the progressive aspect may occasionally be perfective, the reverse is more common, namely that the imperfective aspect is expressed without the progressive. States⁸ typically do not occur in progressive forms, for instance (cf. Comrie, 1976, p. 35)⁹. This is illustrated by the stative verbs *need* and *want*¹⁰ in the simple sentences (4) and (5) below and their seemingly ungrammatical counterparts (6) and (7).

(4) *I need a new jacket.* (imperfective state expressed with the stative verb *need*)

(5) *I want some more money.* (imperfective state expressed with the stative verb *want*)

(6) **I am needing a new jacket.*

(7) **I am wanting some more money.*

It appears the main reason that stative verbs occur rarely in the progressive is because they are imperfective in themselves. Huddleston and Pullum (2002, p. 170) mention several so-called “[v]erbs of cognition”, listed in (8) below.

(8) “*agree, believe, fear, forget, hope, intend, know, like, love, realise, regret, remember, suppose, think, understand, want, wish, wonder*” (p. 170)

⁸ States are generally continuous and thus imperfective (Comrie 1976, p. 51; Figure 1, p. 7).

⁹ Another example is habitual actions (see Section 2.1).

¹⁰ The stative verbs *need* and *want* are mentioned by Brown (1973a, p. 324) as examples of verbs that are not overgeneralized in the progressive by the children in the data (see quote on p. 17).

The verbs listed in (8) above occasionally occur in the progressive, as implied by Huddleston and Pullum (2002, p. 170). One way of understanding why this may be the case is to look at general features of verbs common and uncommon in associated forms. In the opinion of Biber et al. (1999, p. 473; see also Biber et al., 2002, pp. 164-165), mainly two properties determine whether a verb occurs in the progressive form commonly or uncommonly. These properties are expressed in (9) and (10) below.

(9) Criteria determining whether a verb is common in the progressive (Biber et al., 1999, p. 473)

A. Semantic role of the subject is Agent.

B. The action can be extended in time.

(10) Criteria determining whether a verb is uncommon in the progressive (Biber et al., 1999, p. 473)

A. Semantic role of the subject is Experiencer.

B. The action cannot be extended in time.

Conversely, it appears that the criteria in (9) and (10) provide indications why some individual sentences occur in the progressive and others do not. In sentences that do not match the criteria [A] and [B] in (9) or (10), one property may supersede the other. For example, in (11) below, the main subject clearly has the role of Experiencer rather than Agent. This sentence neither matches (9) nor (10) above, but rather (9)[B] and (10)[A]. The progressive aspect in this case may be explained by [B] taking priority over [A] in (9). According to Huddleston and Pullum (2002, p. 170), the progressive aspect in (11) marks focus on the current situation and short time duration.

(11) “*She’s regretting she stayed behind.*” (p. 170)

Some verbs may be stative in one sense and dynamic in another. For example, *hurt* in the sense of experiencing physical pain (cf. Longman, 2018a), is clearly stative. An example is given in (12) below.

(12) “*My back hurts.*” (Longman, 2018a)

On the other hand, constructions with *hurt* used with forms of *be* followed by *-ing* participle may express emotional pain in informal American English or urgent need (cf. Longman, 2018b). An example of the former is given in (13) below. The verb in a dynamic sense is exemplified in (14), where it is used in the progressive aspect to mark focus on the internal process.

(13) “*Martha’s going through a divorce and really hurting right now.*” (Longman, 2018b)

(14) *The prisoners were hurting the guards badly during their escape.*

In this essay, I will argue that stative verbs, like the ones discussed above, may be viewed as exceptions to a progressive rule. Before addressing that, however, we will make a digression into children’s language development (Section 2.4), overgeneralization (Section 2.5) and previous studies of the progressive aspect (Section 2.6).

2.4 Children's language development

When children learn to talk, they progress through several phases, all parts of the language acquisition process (Bochner and Jones, 2003, p. 14). Data supporting the notion of phases or stages have been obtained through large-scale studies of normal children, using standardized testing (pp. 15-16). Bochner and Jones (2003) propose a general model from infancy to the time before school entry. They suggest that there are five developmental stages¹¹, preceded by the preverbal stage (pp. 15-16).

These stages are outlined in Table 2 below, with the information partly supplemented by Hardman (2003). The first stage is characterized by imitation of sounds. The child becomes aware of the connection between sound utterances and communication (Bochner and Jones, 2003, p. 17). In the second stage that ability develops even more. This is typically a phase of more advanced or even word-like sounds connected with people, things and actions. (pp. 16, 19-20). However, not until stage 3 do children acquire their first words, for instance "Mum" and "dog" (p. 16). It precedes construction of basic sentences without morphological features in stage 4. Features like plural -s enter the child's language first during the fifth stage (p. 16). The advanced skills that have now been acquired are essential prerequisites for development of reading skills and numeracy skills (p. ix).

¹¹ Compare the five developmental stages proposed by Ingram (1989, p. 53) and "Piaget's six stages" accounted for by Goldbart (1988, p. 23), both cited by Bochner and Jones (2003, p. 14).

Table 2 . Stages in Children’s language development, as proposed by Bochner and Jones (2003, pp. 14-17) and Hardman (2003, pp. 131-132), especially the table by Bochner and Jones (2003, p. 16[table 2.1]), from which the table below is an adaptation with additional information added.

Stage	Description
Preverbal	Production of vowel sounds like “’ah’ and ‘oo’” without meaning or intention (Hardman, 2003, pp. 131-132).
Stage 1	“Preliminary skills”. From about ½ years of age, children imitate sounds and acquire understanding how to interact with the environment using sounds (Bochner and Jones, 2003, pp. 16-17).
Stage 2	Gestures for “pointing” at things and “ritualized sounds” ¹² associated with certain actions, objects or people (Bochner and Jones, 2003, pp. 16, 19). Vocal sounds similar to words called “protowords”, which express intention (McCarthy, 1954, cited by Ingram (1989, pp. 170-1) cited by Bochner and Jones (2003, p. 20); see also Bochner and Jones (2003, p. 16)).
Stage 3	First real words such as “’Mum’” and “’dog’” (Bochner and Jones, 2003, p. 16)
Stage 4	Simple sentences, generally without morphology (e.g. “’Daddy car’, ‘dog gone’, ‘boy fall down’; ‘cat go there’” [Bochner and Jones, 2003, p. 16])
Stage 5	Introduction of morphology in speech, such as inflectional -s in plural forms (Bochner and Jones, 2003, p. 16).

¹² Bochner and Jones (2003, p. 19) refer to such sounds as “performatives”.

2.5 Overgeneralization

The account given on children's language development in the previous section reflects a development that occurs in stages. The notion that children's understanding improves as they grow older is generally accepted in developmental psychology (Richards and Siegler, 1982, p. 37). There is, however, another phenomenon that seemingly contradicts this principle.

According to Bowerman (1982), when a child becomes aware of a rule, it sometimes uses it in a "blanket fashion" before it learns that there are exceptions to it. This leads to performance errors that are related to the child's analysis of the rule. Such errors may not be present in younger children, who have not yet become aware that there is a rule and therefore cannot apply it on all cases (p. 104). When the child grows older, it presumably learns to take both the rule and the exceptions to it into account, which increases its overall performance. The way of applying a rule on all cases, without considering the exceptions, is known as *overgeneralization*¹³. The associated dip in performance, which may be observed in developmental data, is sometimes referred to as *U-shaped development*¹⁴.

Richards and Siegler (1982, p. 51, 54) claim that language learning offers excellent examples of U-formed development (for a non-linguistic example, see Appendix 3, p. 55). The shape commonly appears when there is a general principle with relatively few exceptions. It is believed to involve three stages (Brown (1973) cited by Richards and Siegler (p. 54)). Initially, there is memorization of a limited number of cases with both regular and irregular instances. Later, children detect the underlying rule and start overgeneralizing it. In the final stage, they combine memorization with rule conformity.

Richards and Siegler (1982, p. 54) mention some examples, citing Cazden (1968). One of the children in the data, Eve, incorrectly used **comed* instead of the verb *came* and Sarah **goed* instead of *went* for past tense. Both cases illustrate U-formed development with

¹³ For example, Richards and Siegler (1982, p. 54) state that "[the children] induce the rule from the regular instances and overgeneralize it to the irregular ones as well."

¹⁴ For instance, Richards and Siegler (1982, p. 54) mention how a child (Sarah) supposedly made the "U-shaped curve" complete by returning to the accurate form later.

overgeneralization. These examples have obviously been collected from the Brown (1973b) child corpora *Eve* and *Sarah*, which are also part of the present study.

2.6 Earlier studies of the progressive

The progressive in English has been studied from different perspectives. For instance, Leech et al. (2009, p. 122), citing Hundt (2004a: 69), present a historical chart showing the long-term development of progressive forms from the 18th century until the end of the 20th century. It indicates continuous increase in progressive frequencies during the period with somewhat stronger increase and higher frequencies for American English than British English. Likewise, Biber et al. (2002, p. 158) have shown that the progressive aspect in spoken registers is more frequent in American than British English.

The question whether the progressive is overgeneralized among children is addressed by Brown (1973a, pp. 324-325). In his opinion, this aspect is not overgeneralized in the Brown (1973b) corpus¹⁵, as indicated by the quote below.

[...] the progressive in our data alone is not overgeneralized. And the opportunity for overgeneralization was there. The children need only have ignored the involuntary nature of the states and said *wanting*, *liking*, *needing*, or the like. Why should no errors occur with the progressive inflection when they do occur with all other inflections?

Brown (1973a, p.324)

In the context of my own study, I argue that stative verbs may be seen as an exception to a main rule. The general principle of this rule is that the imperfective aspect is expressed by the progressive in English¹⁶. This view seemingly conforms with the classification adapted from Comrie (1976, p. 25) in Figure 1, p. 8 (see especially footnote 5). In Section 2.5 we looked at overgeneralization among children, which sometimes occurs when there is a general rule with exceptions. The hypothesis that such a phenomenon may also be present in regard to stative verbs being used with progressive syntax seems plausible. The main aim of the present study is to verify

¹⁵ Notably, the corpus referred to in this paragraph is part of my study (see Section 3.1).

¹⁶ While both habitual actions and stative verbs may be regarded as exceptional cases in this view, this essay only considers the latter case.

whether this hypothesis holds on the basis of spoken corpora for the age interval 1-10 years (see Section 1). Methods and data used for the study will be addressed in the following section.

3. Methods and Materials

3.1 Corpus overview

This sub-section focuses on the corpora that have been used for this study. A contrastive summary of the corpora is provided in Table 4, p. 21. The Brown (1973b) corpus is based on data from three children: Sarah, Eve and Adam. These data are a result of a longitudinal study from 1962 until 1966. The age intervals during which the children were studied are provided in Table 3 below.

Table 3. Age intervals for the child participants in the Brown (1973b) study.

Child	Entry age		Exit age	
	<i>years</i>	<i>months</i>	<i>years</i>	<i>months</i>
<i>Adam</i>	2	3	5	2
<i>Eve</i>	1	6	2	3
<i>Sarah</i>	2	3	5	1

The dialect of the speech recorded in the corpus is American English (Brown, 1973a, p. 272). The corpus is encoded in accordance with the CHAT format (MacWhinney, 2000, p. 30; Brown, 1973b). The Eve corpus is much smaller than the others, because Eve’s family moved away after 20 recording sessions (Brown, 1973b). The Carterette and Jones (1974b; see also Carterette and Jones, 1974a, pp. 11-12) corpus also consists of American English Speech. The children were between 6 and 10 years of age¹⁷. The corpus is based on data from “54 first graders, 48 third graders, 48 fifth graders, and 24 adults¹⁸” (Carterette and Jones, 1974b, superscripted footnote added). The estimated word counts for the Brown (1973b) corpus and the Carterette and Jones (1974b) corpus are 362.000 and 69.000 words respectively (see Table 4, especially footnote 21 on p. 21).

¹⁷ As understood from Carterette and Jones (1974b), the samples were collected from “first-, third- and fifth-grade students”. The corpus files (Carterette and Jones, 1974c) indicate correspondence to ages 6, 8 and 10 years.

¹⁸ Adult speech in the Carterette and Jones (1974b) corpus is not included in my study.

Both the Carterette and Jones (1974b) corpus and the Brown (1973b) corpus belong to a corpus framework named CHILDES. The corpora belonging to this framework are annotated in a similar way. For a more comprehensive description of the CHILDES framework, see Appendix 1, p. 52.

The two control corpora of spoken English relevant for the study are Santa Barbara (Du Bois et al., 2000-2005a) and the spoken section of COCA (Davies, 2008-). Both corpora are based on spoken American English. Notably, COCA is an acronym for “Corpus of Contemporary American English” and Santa Barbara is a “Corpus of Spoken American English” (Du Bois et al., 2000-2005a; Davies, 2008-). COCA is roughly 400 times bigger than the Santa Barbara corpus¹⁹. The latter corpus is interesting for this study because of its absence of morphological annotations, which provides an opportunity to apply the progressive algorithm on an untagged corpus (see Section 3.4).

¹⁹ 118.000.000 words compared to 288.000 words (see Table 4 below, especially footnotes 23 and 24).

Table 4. Summary of the spoken corpora included in my study.

Corpus	Belongs to the CHILDES framework with consistent annotation	Analyzed with the computer application (see Section 3.2)	Dialect ²⁰	Morphologically Annotated	Approximate word count ²¹	Age Interval ²²	Reference
Adam, Eve and Sarah	✓	✓	AmE	✓	362.000	1-5 years	Brown (1973 b-c)
Carterette and Jones	✓	✓	AmE	✓	69.000	6-10 years	Carterette and Jones (1974b-c)
Santa Barbara		✓	AmE		288.000 ²³	Adults	Du Bois et al. (2000-2005a)
COCA, spoken section			AmE	✓	118.000.000 ²⁴	Adults	Davies (2008-)

Methods for computing frequencies of progressive occurrences in the child language corpora and the Santa Barbara corpus (see Table 4) will be addressed in Sections 3.2 and 3.3. The user interface for the spoken section of the COCA corpus (Davies, 2008-) is described in Appendix 11.

3.2 The Computer Application

The *computer application* was developed for doing relevant analyses on the corpora in this study, such as frequency analyses of the progressive aspect. The features of the computer application are briefly summarized in this subsection. I wrote the program in VBA for Microsoft Excel 2016,

²⁰ Sources supporting these claims are cited in the running text of this sub section.

²¹ To calculate word count, the version of the program numbered as 11 in Table A7.1 was used. The approximation was based on the numbers of blank spaces. The number of words was verified in document word counter in Microsoft Word 2016, which showed a 2% difference. For the latter calculation, I used an accumulated corpus, which will be described in Section 3.3.

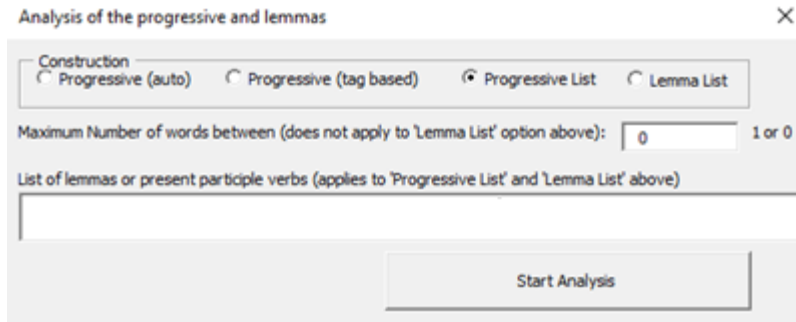
²² For example, on the basis on the output from A7.1[11], which obtained age information from each corpus file and included the result in one of the output tables.

²³ This figure differed about 15% from the figure mentioned on <http://www.linguistics.ucsb.edu/research/santa-barbara-corpus>, retrieved 2018, Nov 25, which may reflect slightly different approaches for calculating word count.

²⁴ This figure was obtained from <https://corpus.byu.edu/coca/help/texts.asp>, 2018, Nov 25.

using the object model (see Appendix 16). Figure 3 below gives an overview of its inputs and outputs. The computer application form is shown in Figure 2.

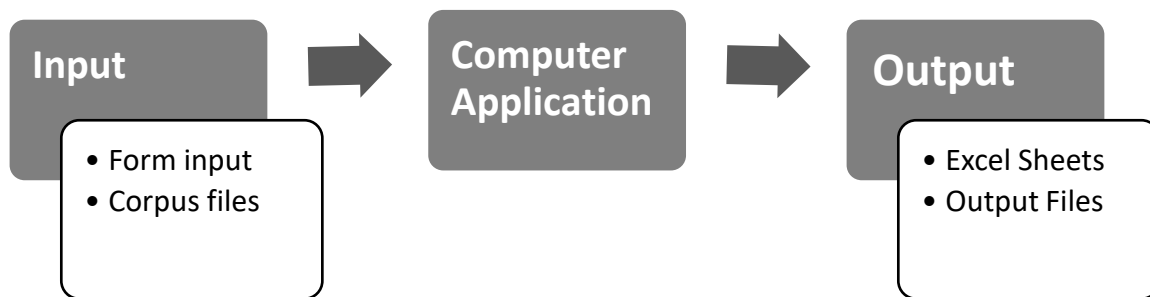
Figure 2. The user form of the application.



The application form has 4 options (‘Progressive (auto)’, ‘Progressive (tag based)’, ‘Progressive List’ and ‘Lemma List’). In addition, there are two textboxes, namely “Maximum Number of words between [...]” and “List of lemmas or present participle verbs [...]”. The button “Start Analysis” initiates the corpus analysis. A more detailed summary of the features is found in Table A5.1, p. 58.

Other inputs besides the form inputs shown in Figure 2 are the corpus files. Either all or a subset of corpus files must be copied to a certain sub folder. The outputs consist of Excel sheet data and annotated files. General frequencies of progressive forms for all corpus files are stored in worksheets. A more detailed account of the application output will be given in the next sub section.

Figure 3. Input and output components of the computer application.



3.3 Application Output

The age group distribution was essential for how the numerical outputs were organized (see Table 5 below). It was established based on data clustering in a scatter plot for all sub corpora (see Appendix 2, p. 53ff). It made it possible to calculate total sums of progressive frequencies and total word counts for all sub corpora and age groups. Doing these calculations for each age group solved the problem with differences in size among the corpus files.

The system made it possible to transfer frequencies and total word counts to individual age group specific columns for each corpus file in the Excel Worksheet containing the output data. A more detailed account how this was done is given in Appendix 6 (see especially the schematic Excel table on p. 61). Similar results could also be obtained with the additional option of including one extra word between the auxiliary verb and the present participle form (see Section 3.4). It was also possible to indicate method of calculation, either pattern-matching or CHAT tagging based. Lists could be specified for verbs occurring in progressive constructions, which applied only to the algorithm. Similarly, lemma lists generated statistics for occurrences of all associated verb forms.

Table 5. The relationship between age group number and age in years.

group number	Age (years)	Age \neq age group number
1	1	
2	2	
3	3	
4	4	
5	5	
6	6	
7	8	✓
8	10	✓
9	adult	✓

Another important type of worksheet output was verb lists generated with the progressive algorithm. These lists were particularly important for the selection process of stative verbs (see Section 3.5). Besides the worksheet output, the computer application even generated an

accumulated output file, with processed data from all sub corpora. This file was structured into sub corpus sections and annotated for the progressive aspect to simplify searches (see Section 3.5).

3.4 The Algorithm

The pattern-matching algorithm has two subtypes labelled ‘zero words between’²⁵ and ‘max one word between’. The meanings of these labels are described in (15) and (16) below. A simplified flowchart of the algorithm is included and described in Appendix 8, which also contains an Appendix reference to a more advanced flowchart.

(15) **Pattern-matching algorithm, subtype ‘zero words between’**

Assumption: frequencies of the progressive with no words between the auxiliary verb and the present participle form (such as *I am singing*) may be approximated by recognizing patterns consisting of forms of the auxiliary verb *be* followed by a word that ends with *-ing*.

(16) **Pattern-matching algorithm, subtype ‘max one word between’**

Assumption: Frequencies of the progressive with at most one word separating the auxiliary verb and the present participle form (e.g. *I am often singing* or *I am singing*) may be approximated by recognizing patterns consisting of forms starting with the auxiliary verb *be*, possibly followed by an arbitrary word, and concluded with a word that ends with *-ing*.

The computer application was customized for doing validation by contrasting results for the original CHAT tagging and the progressive algorithm. It made it possible to validate the algorithm by analyzing the child language corpora using both methods and comparing the results. Following

²⁵ Throughout this essay, if the subtype is not explicitly specified, the ‘zero words between’ case is the default case. For instance, if data from the pattern-matching algorithm are presented without subtype specification, it may be assumed that it applies to the case with no intermediate words separating the form of *be* and the present participle verb.

this approach, the contrastive frequencies could be visualized in diagrams. To obtain an objective indicator of the statistical significance of the progressive algorithm, *p-values* were calculated. A more detailed description how this was done is presented in Appendix 10. The results of the validations are presented in Section 4.1 and discussed in Section 5.1.

3.5 Selection of stative verbs

As described in Section 3.4., the main purpose of computing general frequencies of the progressive with two methods was to validate the accuracy of the algorithm. Once this validation had been done, the next step was to use the computer application to verify the tentative hypothesis in Section 1. This hypothesis is rephrased in (17) below.

- (17) **Tentative hypothesis:** Children commonly overgeneralize the progressive aspect at some time or interval during the first ten years by using it with stative verbs.

In the computer application window, the ‘Progressive List’ option was very similar to ‘Progressive (auto)’ (see Figure 4 and Figure 5 below). The only difference was that ‘Progressive List’ was restricted to a list of verbs rather than all verbs occurring in the progressive. Just like ‘Progressive (auto)’, ‘Progressive (list)’ targeted progressive constructions with either no words or at most one word separating the auxiliary and the present participle verb. In contrast, ‘Lemma list’ did not target progressive constructions. Therefore, specification of how many words there were in between was not applicable to ‘Lemma list’. What ‘Lemma list’ did was that it calculated how many different verb forms were related to certain verb lemmas. The feature did not check for textual patterns but for corpus tagging and was therefore only useful for analyzing the child language corpora in the study, which were grammatically annotated.

The settings for the two list-based features are demonstrated in Figure 4 and Figure 5 below. The verbs listed at the bottom of each form correspond to the stative verbs that were selected for the study. The selection criteria for these verbs will be described in the end of the current sub section, and the result of the selection in Section 4.3.

Figure 4

Analysis of the progressive and lemmas

Construction

Progressive (auto) Progressive (tag based) Progressive List Lemma List

Maximum Number of words between (does not apply to 'Lemma List' option above): 1 or 0

List of lemmas or present participle verbs (applies to 'Progressive List' and 'Lemma List' above)

Start Analysis

Figure 5

Analysis of the progressive and lemmas

Construction

Progressive (auto) Progressive (tag based) Progressive List Lemma List

Maximum Number of words between (does not apply to 'Lemma List' option above): 1 or 0

List of lemmas or present participle verbs (applies to 'Progressive List' and 'Lemma List' above)

Start Analysis

The resulting frequencies from the analyses with the settings specified in Figure 4 and Figure 5 above can be combined for calculations of a *frequency index*²⁶ for the total progressive-to-lemma ratio. The formula for this index for a given set of verbs (called V_{set}) is specified in (19) below. The analogous case for only one verb is shown in (18).

²⁶ The term *frequency index* is used by Mair (2006, p. 115). It is the ratio in decimal form between the frequency of *get*-passives and total frequencies of *get*-passives and *be*-passives (multiplied by 100). In this essay, the basic formula is essentially the same ($100 * (\text{some forms/all forms})$) although the parameters differ.

(18) **Definition.** For a given verb v , *frequency index* is defined as follows.

$$\text{Frequency index} = 100 * \frac{\text{Total frequency of the verb } v \text{ occurring in the progressive}}{\text{Total frequency of all forms of } v}$$

(19) **Definition.** For a given set of verbs, V_{set} , *frequency index* is defined as follows.

$$\text{Frequency index} = 100 * \frac{\text{Total frequency of all verbs in } V_{\text{set}} \text{ that occur in the progressive}}{\text{Total frequency of all verbs in } V_{\text{set}} \text{ in all their forms}}$$

After calculating frequency index for a verb list according to the formula in (19) above, the result was visualized in a diagram for the age groups 1-8 (see Section 4.3, especially Chart 4, p. 34). The form of the curve along with corpus examples provided indications if some of the verbs in the list were overgeneralized. Two verbs of special interest were analyzed one by one across the age groups 1-8.

As for age group 9, the same kind of analysis was carried out using the spoken section of the COCA corpus (see Appendix 11). Notably, the design of the computer application did not allow calculation of frequency index for the Santa Barbara corpus (Du Bois et al., 2000-2005a) since this corpus was not annotated for verb lemmas (see Section 6 for a possible future improvement in this regard).

The remaining part of this sub section will address the selection process for the stative verbs. The verbs were selected on the basis of two criteria, specified in (20) below (see Sections 2.3 and 3.3).

(20) **Criteria for selection of verbs for the stative-verb list**

- A. Each verb must occur in the progressive aspect in the Brown (1973b) corpus.
- B. In at least one sense, each verb must be state-like, lack human Agent and the action indicated by the verb must not be extendable in time²⁷ (cf. Section 2.3).

I will go through each criterion listed in (20) and explain my motivations behind it. *Firstly*, for the analysis to be relevant, the selected verbs *must* occur in the sub corpora for the youngest children (1-5 years). If not, the overgeneralization study would be limited to the older participants, which would render it useless. *Secondly*, at least one sense of the verb must be state-like, without human Agent and the associated action not extendable in time (see (12) and (14)²⁸, p. 12ff).

Once the selection process had been completed, preliminary examinations were done of all five stative verbs. Subsequently, two of them were chosen for more thorough analyses. Determination of whether a verb was qualified or not for such an analysis was done on the basis of progressive frequencies. Frequency developments across age group 1 to 8 and age group 9 were studied individually only for the two chosen stative verbs. Corpus examples were, however, obtained for all 5 verbs. Collection of corpus examples was done by performing searches for progressive annotations in the accumulated output file²⁹.

On the basis of frequency developments and corpus examples, it was investigated whether the two chosen verbs were possible candidates for overgeneralization. The single verb that remained after elimination of all other stative verbs (*hurt*) was contrasted with a random sample from the spoken section of the COCA³⁰ corpus (Davies, 2008-) in terms of progressive frequencies. Subsequently, frequency adjustments were made based on proportions of relevant corpus entries for different age groups.

²⁷ In this essay, this property may sometimes be described in terms of the verb itself not being extendable in time.

²⁸ (12) and (14) exemplify how a verb may be stative in one sense and dynamic in another.

²⁹ See Section 3.3.

³⁰ The results were not contrasted with the adult control corpus Santa Barbara (Du Bois et al., 2000-2005a). For a motivation, see Section 4.3.

3.6 Alternative approaches

To answer the research questions in Section 1, relevant child corpora were required. Most child corpora on the CHILDES corpora website³¹ were reviewed, but seemingly none of them, except for the ones finally included, met the criteria in terms of corpus size, age interval, normal language development etc. One corpus seriously considered at first was the *Bliss* corpus³¹ for ages 3 to 10. It turned out, however, to be of too limited size and was therefore discarded.

For evaluation of the initial hypotheses, I developed a computer application (see Section 3.2). A perhaps less time-consuming approach would have been to use the already existing CLAN software for analysis of the child corpora (see Appendix 1). This would have meant, however, that the untagged control corpus Santa Barbara had to be excluded. It is undoubtedly true that the study *could* have been done in such a way and achieved many of the same goals. On the other hand, an important methodological objective would have been lost. It concerned development of the progressive algorithm, which has a potential role in future studies for analysis of untagged speech data (see Section 6).

³¹ See <https://childes.talkbank.org/access/Eng-NA/>, link verified on Jan 5, 2019.

4. Results

4.1 Validation of the algorithm

In Section 3.3, I have demonstrated how frequencies of the progressive were calculated, using the computer application. The computations were based either on the progressive algorithm or the corpus tagging. The technical aspects of the algorithm and validation methods have been covered in Section 3.4. In this sub section, the results of the validation will be presented.

As discussed in Section 3.4, the reliability of the algorithm was indicated by the obvious correlation between the two curves with tag-based analysis vs. progressive algorithm analysis. In addition, p-values were calculated. Chart 1 shows these results for the ‘zero words between’ subtype (see (15), p. 24). Chart 2 presents similar results for the other subtype with ‘max one word between’. For a more specific account on how p-values were calculated, see Appendix 10.

Chart 1. Contrastive curves for development of progressive forms in the child language corpora (Brown (1973b) and Carterette and Jones (1974b) for age groups 1-8 (see Table 5, p. 23) for the subtype ‘0 words between’ (see Section 3.4). The two curves contrast the pattern-matching algorithm (AUTO) with tag-based identification (TAG).

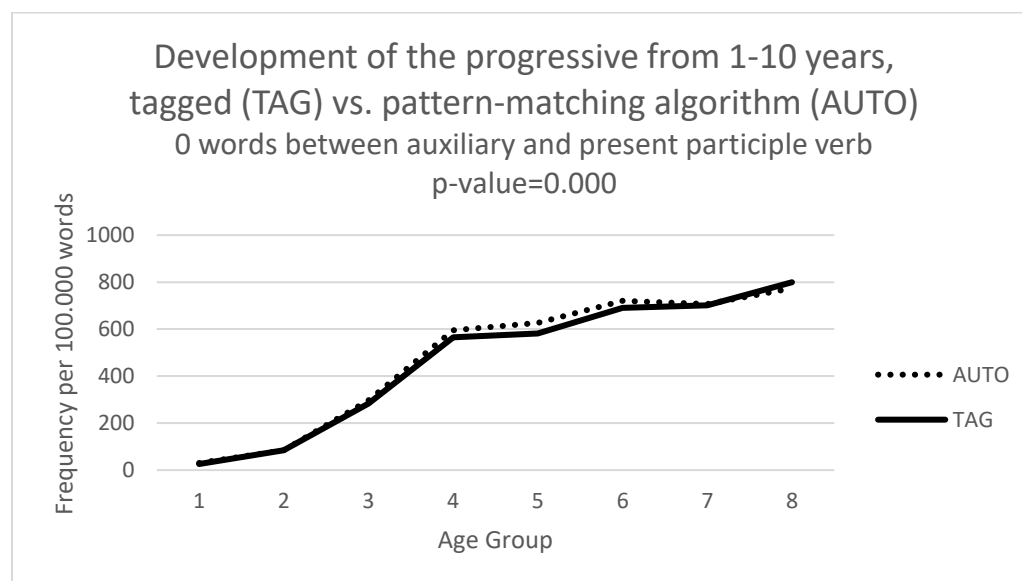
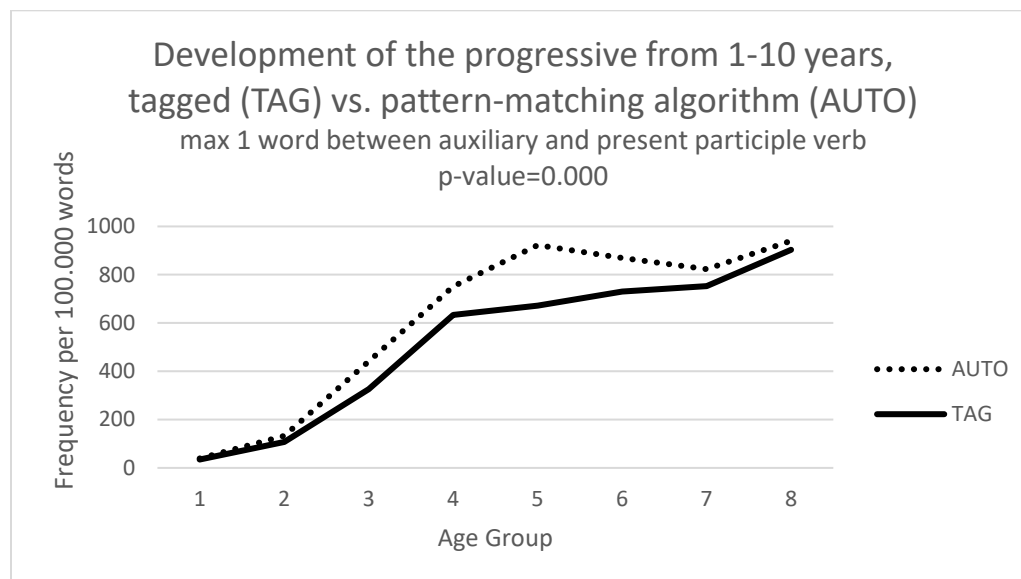


Chart 2. A similar type of development as reflected by Chart 1 above but for a different subtype, namely ‘max 1 word between’ (see Section 3.4).



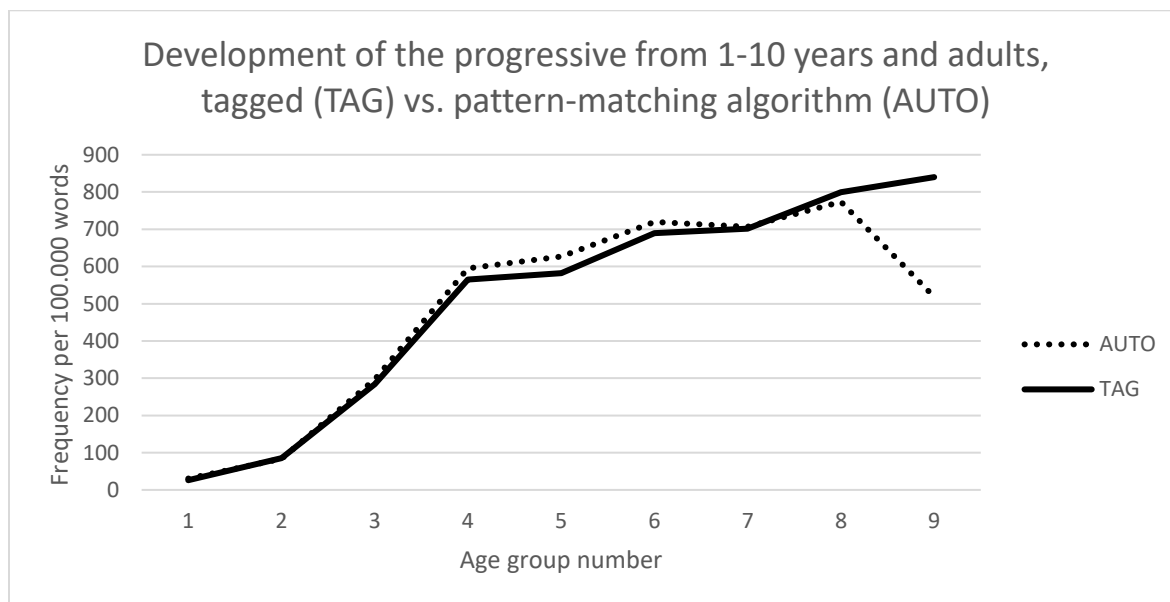
4.2 General trends

The overall development of the progressive in the interval 1-10 years (i.e. age group 1-8) is partly indicated by the charts in the previous section (Chart 1 and Chart 2). Only data connected with the adult control groups are missing in these charts. A comprehensive visualization of the frequency development for the progressive aspect is included in Chart 3 below³² for subtype ‘zero words between’ only³³ (see Section 3.4). The results will be discussed in Section 5.2.

³² Chart 3 includes child corpus data and data from both adult control groups.

³³ The differences between the two cases with either no word separators or max one word separating the auxiliary verb and the present participle word were not considered significant enough to motivate a separate analysis for the ‘max one word between’ case (see Section 3.4).

Chart 3. Development of the progressive for subtype ‘zero words between’ (see Section 3.4) from 1-10 years (i.e. group 1-9) and contrasted with the adult control groups. The age group interval 1-8 corresponds to Chart 1, p. 30. For group 9, TAG indicates measurement from the COCA (Davies,2008-) corpus and AUTO Santa Barbara corpus (Du Bois et al., 2000-2005a).



4.3 Selected stative verbs

The verb lists in Appendix 14 were only generated by computer application features involving the pattern-matching algorithm for the progressive aspect and not tag-based identification. Based on List A14.1, p. 76, and the criteria specified in (20), p. 28, the verbs listed in Table 6 below were selected.

Table 6. Stative verbs from the Brown (1973b) corpus selected for analysis.

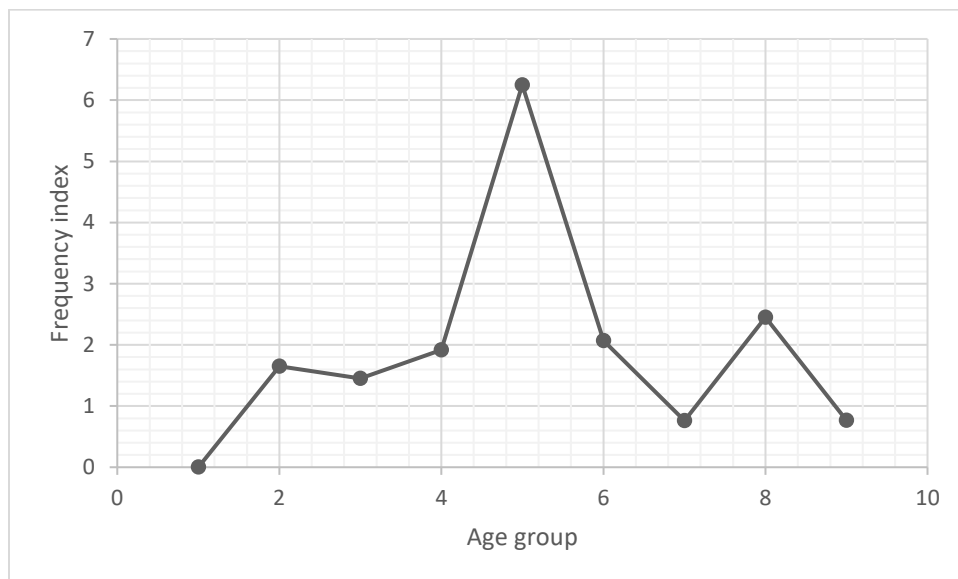
Base form	1 st Criterion The verb occurs in the progressive in the Brown (1973b) corpus	2 nd and 3 rd Criteria: In at least one sense,...		
		the verb is state-like	the verb lacks human Agent	the action of the verb is not extendable in time
<i>feel</i>	✓	✓	✓	✓
<i>have</i>	✓	✓	✓	✓
<i>hurt</i>	✓	✓	✓	✓
<i>shine</i>	✓	✓	✓	✓
<i>taste</i>	✓	✓	✓	✓

The five selected stative verbs (see Table 6 above) were analyzed in accordance with the account given in Section 3.5 for all child language corpora and COCA, spoken section (Davies, 2008-). The computer application was used with ‘progressive list’ and ‘lemma list’ specified in accordance with (21) below³⁴. The resulting data are visualized in Chart 4.

³⁴. The settings were specified as shown in Figure 4 and Figure 5, p. 26ff. The analysis of the COCA corpus (Davies, 2008-) was not done by the computer application. It is described in the end of Section 3.1.

(21)	Progressive verb list:	feeling, having, hurting, shining, tasting
	Lemma List:	feel, have, hurt, shine, taste

Chart 4. Development of *frequency index* (see Section 3.5, (19))³⁵, for the 5 stative verbs given in Table 6, above for age groups 1 to 9.



The selected stative verbs and their actual frequencies in the child corpora, with no words between the auxiliary and present participle verb, are specified in Table 7 below, which only includes the child corpora. Among the 22 instances of stative verbs occurring in the progressive, the two most frequent verbs are listed with examples in Table A13.1, p. 73. Verbs occurring in very low frequencies in associated forms (*feel, taste and shine*) are included in the Tables A13.4 until Table A13.7 along with their contexts.

³⁵ For Chart 4 it should be noted that frequency index for each age group was calculated with the formula $100 \cdot F_p / L_p$ where F_p is frequency of the five stative verbs occurring in the progressive and L_p the frequency of all verb forms connected with the verb lemma (see (18) and (19), p. 25ff). F_p was calculated with the progressive algorithm whereas L_p was calculated based on the corpus tagging.

Table 7. Absolute frequencies of occurrences of five stative verbs in the progressive in the child language corpora Brown (1973b) and Carterette and Jones (1974b). For example, there were 12 instances of *have* but only 6 examples of *hurt* in progressive constructions in the child data (with no words separating the auxiliary and present participle verb). All child corpus examples with *have* and *hurt* in this aspect are listed in Table A13.1, p. 73.

Verb	Frequency
Feeling	1
Tasting	1
Shining	2
Hurting	6
Having	12
total	22

In Section 5.3, evaluation of the results above will be followed by an analysis of the two most frequent stative verbs. According to the frequency list in Table 7, *have* was the most frequent stative verb in the progressive followed by *hurt*. Developmental curves for frequency index of *have* and *hurt* in this aspect in relation to all verb forms are shown below (see Chart 5 and Chart 6). Absolute frequencies for all age groups are listed in Table A15.1, in which they are contrasted with absolute frequencies including all verbs.

Chart 5. Frequency index development from age group 1 to 9 (see (19) in Section 3.5) for the verb form *having* in progressive syntax in relation to all verb forms of *have*. Group 9 represents the COCA corpus, spoken section (Davies, 2008-).

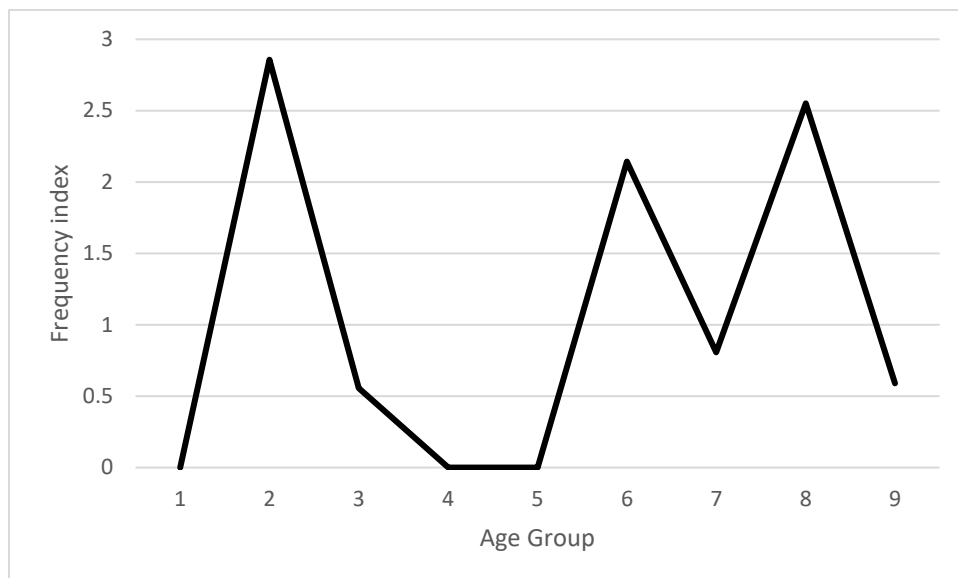
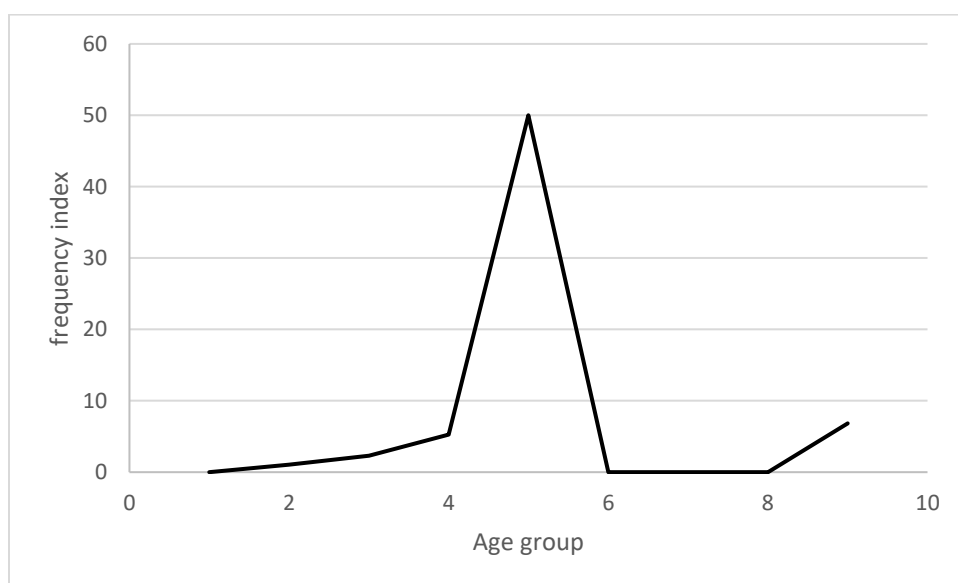


Chart 6. Frequency index development for age group 1 to 9 (see Section 3.5, (19)) for the verb form *hurting* in the progressive in relation to all verb forms of the lemma HURT. Group 9 represents the COCA corpus, spoken section (Davies, 2008-).



The underlying data for the charts presented above (Chart 4 (p. 34), Chart 5 (p. 36), and Chart 6 (p. 36)) are listed in Table A15.1, Table A15.2, and Table A15.3. In addition, the frequency table for the individual verb *hurt* (Table A15.3) has an additional column for adjusted frequencies, taking the proportions of stative senses into account. The same table contains an additional column with corpus size for each age group, which is used for calculation of relative frequencies per 100.000 words.

5. Discussion

This section is mainly dedicated to discussing and interpreting the results provided in Section 4 and trying to answer the research questions specified in Section 1. All 5 selected stative³⁶ verbs will be analyzed in Section 5.3. The two verbs most frequently found in the progressive, namely *hurt* (6 occurrences) and *have* (12 occurrences) (see Table 7, p. 35) will be evaluated more thoroughly. A flowchart summarizing the entire study is included in Table 9, p. 46. Throughout this section, this flowchart will regularly be referred to as a reminder of the various stages in the process.

5.1 Evaluation of the algorithm

General frequency developments of the progressive aspect are computed with two different methods, namely tag-based identification and a pattern-matching algorithm. In Chart 3, p. 32, the curves of these two methods are placed side by side for group 1 to 8. The algorithm plays an important role in this study. *Firstly*, it allows inclusion of an untagged corpus, Santa Barbara (Du Bois et al., 2000-2005a). It serves as an illustration how corpus analysis not necessarily depends on morphological annotations. *Secondly*, the algorithm also has a methodological value and potential for future studies as discussed in Section 6.

The validation method involves analyzing all child language data on the basis of both the algorithm and existing corpus tagging (see Sections 3.4). Comparing the outcomes of these two methods gives an indication how well the progressive algorithm is correlated to the corpus tagging. The results of the validation have been presented in Section 4.1. Chart 1 and Chart 2 on p. 30ff show the contrastive curves side by side for each subtype ('zero words between' or 'max one word between').

A quick look at these charts gives the impression that the curves are well correlated with the correlation being somewhat stronger in Chart 1. Calculations of *p-values* support this observation. Based on the *p-value* computation described in Appendix 10, p. 66, it may be concluded that the

³⁶ The verbs were stative in at least one sense.

algorithm is fairly accurate at predicting progressive frequencies in accordance with the corpus tagging, since the p-values for both subtypes are approximately zero.

5.2 General developments of the progressive

Sections 3.2 and 3.3 outline how the computer application³⁷ can be used to calculate relative³⁷ progressive frequencies for age groups³⁸ 1-8 using either of the two methods described above. To calculate the same type of frequencies for group 9 (corresponding to adult language), the Santa Barbara corpus (Du Bois et al., 2000-2005a) and the COCA corpus, spoken section (Davies, 2008-) are used. In Chart 3 on p. 32, all these frequency results are then combined. This chart may be seen as an overview of the frequency development of the progressive from childhood to adulthood³⁹ (see stage A in the study flowchart in Table 9, p. 46). It gives the impression of a continuous increase of this aspect throughout the whole period. The steadily increasing trend indicates that speakers overall tend to use progressive forms more frequently as they grow older.

It can be reasonably assumed that possible overgeneralization among individual stative verbs has no noticeable impact on the general trend presented in Chart 3. Table A15.1, p. 80, with absolute frequencies, supports this assumption. It suggests that the numbers of progressive frequencies of stative verbs are relatively small compared to progressive frequencies in general when including all verbs.

Interestingly, the two curves in Chart 3, p. 32, clearly differ for age group 9, i.e. the adult control group. While the result generated from the COCA corpus, spoken section (Davies, 2008-) essentially conforms to the general trend, there appears to be a dip in the curve where group 9 is calculated based on the Santa Barbara corpus (Du Bois et al., 2000-2005a). This can be explained by the fact that the spoken section of the COCA corpus is more than 400 times bigger than the Santa Barbara corpus (see Table 4, p. 21). The COCA corpus can therefore be argued to be much

³⁷ The term *relative frequency* denotes frequency per 100.000 words in this essay.

³⁸ See Table 5, page 23.

³⁹ Data for Chart 3 were only obtained for the subtype ‘zero words between’ because the differences between frequencies for this subtype and the other subtype appeared to be relatively small (compare Chart 1 and Chart 2 on page 31).

more reliable than the Santa Barbara corpus, since results from the latter undoubtedly are more susceptible to deviations among its individual sub corpora. Chart A2.1 indicates a wide frequency distribution among these.

5.3 Developments of the progressive for stative verbs

The main criterion in this study for selection of stative verbs is that they must be used in the progressive by children 1-5 years of age. To be valid, a stative verb must also lack human Agent, be state-like and not extendable in time for at least one of its senses. The five verbs selected from Table A14.1 appear to meet these criteria (see stages B and C in the study flowchart, p. 46).

It may be argued that the number of stative verbs in this aspect, 22 (see Table 7, p. 35), is small. Conclusions drawn from such data must necessarily be tentative. Three of the five verbs (*feel*, *shine* and *taste*) occur just one or two times in progressive forms. Verbs with so low frequencies do not even seem worthwhile to analyze over a 10 years interval. From the surrounding contexts of these 3 verbs (see Tables A13.4 until Table A13.7), it is clear that the children do not use them in a stative sense (see for instance (23) and (24) below). There is one exception, however. The verb *feel* is used once in a progressive construction in a seemingly stative sense (see (22) below). That construction is provided with context in Table A13.4. Because there is only one case with *feel* in the progressive with ‘zero words between’, the verb is not analyzed further.

(22) “*Percy's feeling well .*” (Sarah, line 1328; filename:030110.cha (Brown, 1973c), Group: 3)

(23) “*he's shining his shoes?*” (Adam, line 252; filename:040511.cha (Brown, 1973c), Group: 4)

(24) “*I'm tasting some .*” (Adam, line 2752; filename:041023.cha (Brown, 1973c), Group: 4)

As indicated in Section 4.3, two of the stative verbs, *have* and *hurt*, were more frequent than the rest and were therefore analyzed in terms of frequency development across the age groups, using the computer application. Corpus examples with these verbs are listed in Table A13.1 (p. 73). Their respective frequency index development curves are illustrated by Chart 5 and Chart 6, p.

36ff⁴⁰ (Section 4.3). Chart 5 illustrates development for the verb *have* in the progressive. There are several intervals where frequency index (see (19), p. 27) is significantly higher among the younger age groups than for the adult control group. Such overuse might be interpreted as overgeneralization, e.g. in (25) below. In example (26), however, the progressive aspect refers to a future event⁴¹ and does not have stative meaning (see Section 2.2).

(25) “*what you was having on you nose ?*” (Table A13.1 [8], p. 73)

(26) “*Sue (.) we're having noodles*” (Table A13.1 [9])

Other examples where the verb *have* is not really a candidate for overgeneralization are (27) and (28) below. In these examples, the progressive suggests limited time duration and some degree of volition. It is presumably motivated here and not connected with overgeneralization.

(27) “*... we were having a science discussion ...*” (Table A13.1 [13])

(28) “*...they were having dinner and then one of the boys in the room they that he had the keys and and then he got out but he climbed through the window...*” (Table A13.1 [16])

Review of all 12 examples with the verb *have* occurring in the progressive⁴² suggests there is only one clear case of possible overgeneralization, namely (25) above. It may thus be concluded that the verb *have* is not overgeneralized in progressive constructions among these children (see stage F in the study flowchart, p. 46). Still, this verb appears to be associated with comparatively high frequency indexes⁴³ (see Chart 5, p. 36; Table A13.1). Could proportionally higher usage of *have* in such forms be characteristic of child language? It may partly be associated with a stronger tendency to express past activities using nominalized forms, such as in “*we were having a science discussion*” in (27) rather than using a verb, like *discuss*, in the past progressive (a similar example is given in (28) above).

⁴⁰ Compare stages D and E in the study flowchart on p. 46.

⁴¹ The surrounding context contradicts the possibility that this is an event in progress.

⁴² See Table A13.1.

⁴³ Frequencies of *have* in the progressive in proportion to frequencies of all forms of *have*.

The other chosen verb, *hurt*, occurs 6 times in the progressive in the child data. It is seemingly overgeneralized on the basis of the verb examples in Table A13.1 (p. 73). The frequency development seen in Chart 6, p. 36, is, however, slightly misleading. The central peak is based on data with only one instance of *hurt* in this aspect, in relation to 2 instances including all forms of *hurt*. This generates a frequency index of 50 ($100 * 1 / 2$) for age group 5. If that single progressive occurrence was not there, the frequency for the adult control group would be higher than for the children. In such case, the chart alone would not support overgeneralization.

Because of the inconclusive analysis of the verb *hurt* above, it is necessary to also consider the proportion between stative⁴⁴ and non-stative occurrences. These proportions are estimated on the basis of the actual examples from the child corpora and 20 random examples from the COCA control corpus (see Table A12.2). It appears that *hurt* occurs in the progressive in a purely stative sense⁴⁵ proportionally much more often in the child corpora. In the COCA sample (Table A12.2) it seems there is just 1 instance of 20 with *hurt* in a stative sense but with progressive syntax⁴⁶. This essentially invalidates the result shown by Chart 6, p. 36, for the adult control corpus. In contrast, all child corpus examples that include *hurt* in the progressive form in Table A13.1 (p. 73) seem to suggest experience of physical pain in a stative sense (see Table 8, p. 43).

⁴⁴ In this context, classification as *stative* means the verb in the progressive may be overgeneralized. For instance, if the progressive construction is a special expression that is lexicalized, it counts as *non-stative* since it could then not be overgeneralized.

⁴⁵ See Section 2.3, p. 10, for a more detailed discussion on stative and non-stative senses of the verb *hurt*.

⁴⁶ A common construction in the adult language sample from COCA, spoken section (Davies, 2008-), is the form of *be hurting* in the progressive which may be used in informal American English to indicate emotional pain (see Section 2.3, p. 10). Since the expression is lexicalized it may be argued that it cannot be overgeneralized (see footnote 44 above).

Table 8. All examples from Table A13.1 (p. 28) with the verb *hurt* used in the progressive. For specific corpus references, see Table A13.1.

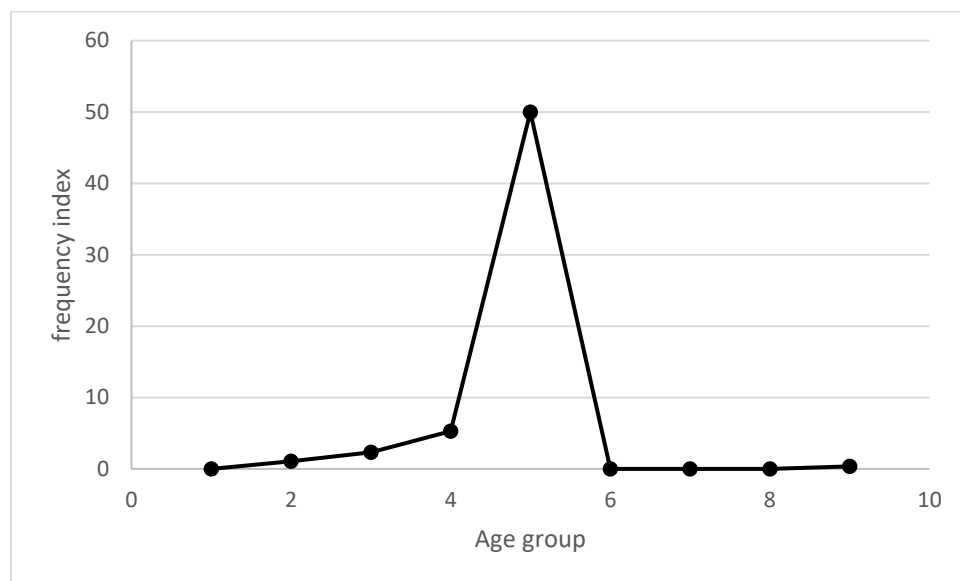
verb	Corpus example	Cross reference to entry in Table A13.1 (p. 73)	age group	Used in a stative, non-agentive sense that clearly implies sensation of physical pain
hurting	... my neck is hurting	20	2	✓
hurting	...my head's hurting...	21	3	✓
hurting	... see this is hurting	22	3	✓ ⁴⁷
hurting	...I need it on it's hurting my neck...	23	4	✓ ⁴⁸
hurting	... something's hurting me	24	4	✓ ⁴⁸
hurting	... no (.) my back was hurting	25	5	✓

The proportions of stative examples with the verb *hurt* occurring in progressive constructions in the child corpora and the random sample from COCA, spoken section (Davies, 2008-), are presumably 100% (see Table 8 above) and 5% (see Table A12.2) respectively. These proportions may be used to adjust the data on which Chart 6, p. 36, is based. Chart 7 below is a version of Chart 6 adjusted in this manner.

⁴⁷ By studying the extended corpus, there is no indication that *hurt* used in the progressive in this particular example could possibly refer to anything else than sensation of physical pain.

⁴⁸ In “[...]it's hurting my neck[...]” and “[...]something's hurting me [...]” (see Table 8), the pain is seemingly related to pieces of clothing causing involuntary states of discomfort (see Table A13.2 and Table A13.3 for extended contexts).

Chart 7. Frequency index development for the verb *hurt* in the progressive. It is based on the same data as Chart 6, p. 36, with adjustments made in accordance with the proportions of stative examples among the corpus entries. For additional information, see description of Chart 6.



Based on the adjusted data presented in Chart 7 above, it appears there may indeed be a tendency to overgeneralize the verb *hurt* in the progressive among children 3-5 years of age (see stages G and F in the study flowchart, p. 46). It can be claimed that the frequency of relevant examples in the underlying data is low (see Table A15.3, p. 80). On the other hand, the near absence of stative examples in the COCA sample (see above) is an important piece of evidence supporting overgeneralization.

Obviously, the elevated frequencies of the verb *hurt* are not associated with increased progressive frequencies overall. As observed in Section 5.2, the general frequency development for progressive forms is found to be quite steady from group 1 to 9 (i.e. ages 1-10 + adult control group), with no significant deviation from the rising trend (see Chart 3, p. 32). In contrast, Chart 7 above shows clear elevation for the intermediate age groups 3 to 5 years. This contrast cannot be explained by a difference in calculation method, i.e. calculation of relative frequencies⁴⁹ versus frequency indexes⁵⁰. In fact, Table A15.3 indicates a similar pattern as Chart 7 for adjusted relative

⁴⁹ *Relative frequencies*: Progressive frequencies per 100.000 words.

⁵⁰ *Frequency index*: Progressive frequencies divided by frequencies of all related verb forms. This value is then multiplied by 100.








frequencies with a raised formation in the same age interval (see Chart A15.1, p. 82). This observation adds extra support to the hypothesis that *hurt* in the progressive is overgeneralized in the child data.

5.4 Conclusion

The obvious interpretation of the results is that there are few instances of overgeneralization in the data and only for one verb. Like Brown (1973a, p. 324) points out⁵¹, the children (1-5 years) do not use verbs such as *like*, *want* and *need* in the progressive (see quote on page 17). The same conclusion holds true for the older children, 6-10 years, according to my study of the Carterette and Jones corpus (1974b). One conceivable explanation might be that the children are almost never exposed to sound sequences of *-ing* participle constructions of these and other comparable verbs. For example, if a child never hears *liking* or *wanting* it may not be inclined to use these sound sequences in its language. *Hurt* is a special case. Children are presumably exposed to progressive constructions of this verb in its dynamic sense from time to time (see Section 2.3, p. 11). Meanwhile, it may be assumed on the basis of the corpus examples that the stative sense is relevant to them linguistically. These circumstances may influence their language and increase their tendency to overgeneralize this verb.

⁵¹ The same data on which Brown (1973a, p. 324) bases his conclusion are also included in my study (see page 45).

Table 9. The flowchart below is a condensed overview of the selections leading up to the conclusion that the verb *hurt* is overgeneralized in the progressive.

A.	General analysis of the progressive 1-10 years + adults	See Section 4.2, especially Chart 3, p. 32
		
B.	List of verbs used in the progressive based on children of ages 1-5 (Brown, 1973b).	See Sections 4.3 and 5.3. See also Appendix 14.
		
C.	Selection of 5 stative verbs from the list in (B) <i>feel, have, hurt, shine, taste</i>	See Section 4.3, especially Table 6, p. 33.
		
D.	Selection of the two most frequent verbs in (C)	see Section 4.3, especially Table 7, p. 35.
		
E.	Selected verbs in (D): <i>have and hurt</i>	See Section 5.3
		
F.	Dismissal of <i>have</i> in (E) on the basis of examples in the child corpora. Only <i>hurt</i> remains for further analysis.	See Sections 4.3 and 5.3
		
G.	Frequency adjustment for <i>hurt</i> in the progressive.	See Section 5.3 (especially Chart 7, p. 44).
		
F.	<i>Tentative conclusion:</i> the verb <i>hurt</i> is overgeneralized in the progressive by the children	See Section 5.4

6. Future developments

Brown (1973a, p. 324) claims⁵² there is no overgeneralization of the progressive aspect for the age interval 1-5 years in the Brown (1973b) corpus. He points out that the verbs *want*, *like* and *need* among others do not occur in progressive constructions. This study, which is partly based on the same data, has seemingly found one exception, namely the verb *hurt* used in the progressive form. There is no evidence for overgeneralization of other stative verbs.

One verb, *have*, is used proportionally⁵³ more in the progressive by children than adults in some age intervals, however. In Section 5.3 it was hypothesized that this overuse might be characteristic of children's speech. An in-depth study of the most common progressive verbs might be of assistance in discovering other cases of similar overuse among children. As my study has shown, the progressive in general seems to undergo a steady increase in terms of relative frequencies⁵⁴ from childhood to adulthood. What do the corresponding trends look like for the most common progressive verbs? Does it matter for the trend shape whether the calculations are based on relative frequencies or frequency indexes⁵⁵? Are some verbs used proportionally more often in this aspect by children than adults? If so, for what reasons? How do trend shapes for individual verbs differ from the general case?

Another interesting development is of methodological importance. It involves the construction of an automatic algorithm for identification of frequencies for all lemma forms for specified verbs. Combined with the progressive algorithm, it would be a *frequency index algorithm*. Validation of this new algorithm could be done by contrasting it with computations based entirely on corpus tagging, in an analogous way as the progressive algorithm has been validated in the present study. Such an approach would make it possible to analyze frequency indexes for verb lists in untagged data. It might be useful in a scenario with large quantities of speech data that are not morphologically annotated.

⁵² See quote on p. 17

⁵³ In proportion to all forms of *have*.

⁵⁴ Relative frequency: number of occurrences per 100 000 words

⁵⁵ Frequency index: see definition in (19), p. 27.

References

- Biber D., Conrad S. and Leech G (2002). *Longman student grammar of spoken and written English*. Harlow : Longman
- Biber, D., Johansson, S., Leech, G., Conrad, S. and Finegan, E. (1999). *Longman grammar of spoken and written English*. Harlow : Longman, 1999.
- Bochner, S., & Jones, J. (2003). *Child language development : learning to talk, Second Edition* [Electronic resource]. London : Whurr Publishers, 2003. retrieved 20 Oct, 2018 from <http://ludwig.lub.lu.se/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=cat01310a&AN=lovisa.005006569&site=eds-live&scope=site>
- Bowerman M. (1982). *Starting to Talk Worse: Clues to Language Acquisition from Children's Late Speech Errors* in Strauss, S. & Stavy, R. *U-shaped Behavioral Growth*. Academic Press.
- Brown, R. (1973a). *A first language : the early stages*. London : Allen & Unwin, cop. 1973.
- Brown, R. (1973b). Brown Corpus, retrieved 23 Oct, 2018 from <https://childes.talkbank.org/access/Eng-NA/Brown.html>
- Brown, R. (1973c). Brown Corpus [Corpus data] retrieved 4 Oct, 2018 from <https://childes.talkbank.org/data/Eng-NA/Brown.zip>
- Carterette, E. C. & Jones, M. H. (1974a). *Informal Speech, Alphabetic & Phonemic Texts With Statistical Analyses And Tables*. London:University of California Press, Ltd., 1974
- Carterette, E. C. & Jones, M. H. (1974b). *Carterette and Jones Corpus*. retrieved 24 Oct 2018 from <https://childes.talkbank.org/access/Eng-NA/Carterette.html>
- Carterette, E. C. & Jones, M. H. (1974c). *Carterette and Jones Corpus* [Corpus Data]. retrieved 23 Oct 2018 from <https://childes.talkbank.org/data/Eng-NA/Carterette.zip>

- Comrie, B. (1976). *Aspect : an introduction to the study of verbal aspect and related problems*. Cambridge : Cambridge Univ. Press, 1976.
- Davies, Mark. (2008-) *The Corpus of Contemporary American English (COCA): 560 million words, 1990-present*. retrieved Oct-Nov, 2018 from <https://corpus.byu.edu/coca/>
- Du Bois, John W., Wallace L. Chafe, Charles Meyer, Sandra A. Thompson, Robert Englebretson, and Nii Martey. (2000-2005a). *Santa Barbara corpus of spoken American English, Parts 1-4*. Philadelphia: Linguistic Data Consortium. retrieved 4 Oct 2018 from <http://www.linguistics.ucsb.edu/research/santa-barbara-corpus>
- Du Bois, John W., Wallace L. Chafe, Charles Meyer, Sandra A. Thompson, Robert Englebretson, and Nii Martey. (2000-2005b). *Santa Barbara corpus of spoken American English*. retrieved Oct 24, 2018 from <http://www.linguistics.ucsb.edu/research/santa-barbara-corpus>
- Google Inc. (2018). *Google Chrome* [Web Browser]. Version 70.0.3538.102 (Official Build) (64-bit). Google Inc (2018).
- Hardman, C. A. (2003). Phonological development and Intelligibility. In Bochner, S., & Jones, J. *Child language development. : learning to talk* (pp. 131-142).[Electronic resource]. London : Whurr Publishers, 2003. retrieved from <http://ludwig.lub.lu.se/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=cat01310a&AN=lovisa.005006569&site=eds-live&scope=site>
- Huddleston, R. & Pullum, G.K. (2002). *The Cambridge Grammar of the English Language*. Cambridge: Cambridge UP. ISBN 978-0521431460
- Kreider, R. (2016). *How to determine P value using Excel - Rebecca Kreider* [Video file]. retrieved from <https://www.youtube.com/watch?v=XDJPbUXD2no>, Nov 28, 2018
- Leech, G., Hundt M., Mair C. and Smith N., (eds). 2009. *Change in Contemporary English: A Grammatical Study*. Cambridge: CUP

- Longman (2018a). *Longman Dictionary of Contemporary English Online* [entry: hurt]. retrieved Nov, 30, 2018, from <https://www.ldoceonline.com/dictionary/hurt>
- Longman (2018b). *Longman Dictionary of Contemporary English Online* [entry: be hurting]. retrieved Nov, 30, 2018, from <https://www.ldoceonline.com/dictionary/be-hurting>
- MacWhinney, B. (2000). *The CHILDES Project Tools for Analyzing Talk, Third Edition. Volume II: The Database*. Mahwah, N.J. ; London : Lawrence Erlbaum, 2000.
- Mair, Christian. 2006. *Twentieth-Century English: History, Variation, and Standardization*. Cambridge: CUP
- Microsoft (2018a). *Object model (Excel)*. retrieved Nov 26, 2018 from <https://docs.microsoft.com/en-us/office/vba/api/overview/excel/object-model>
- Microsoft (2018b). *Application object (Excel)*. retrieved Nov 26, 2018 from [https://docs.microsoft.com/en-us/office/vba/api/excel.application\(object\)](https://docs.microsoft.com/en-us/office/vba/api/excel.application(object))
- Microsoft (2018c). *Application.Worksheets property (Excel)*. retrieved Nov 26, 2018 from <https://docs.microsoft.com/en-us/office/vba/api/excel.application.worksheets>
- Microsoft (2018d). *Sheets object (Excel)*. retrieved Nov 26, 2018 from <https://docs.microsoft.com/en-us/office/vba/api/excel.sheets>
- Microsoft (2018e). *Worksheet object (Excel)*. retrieved Nov 26, 2018 from <https://docs.microsoft.com/en-us/office/vba/api/excel.worksheet>
- Microsoft (2018f). *Worksheet.Cells property (Excel)*. retrieved Nov 26, 2018 from <https://docs.microsoft.com/en-us/office/vba/api/excel.worksheet.cells>
- Richards D. & Siegler R. (1982). U-Shaped Behavioral Curves: It's Not Whether You're Right or Wrong, It's Why in Strauss, S. & Stavy, R. *U-shaped Behavioral Growth*. Academic Press.
- Saeed, J. I. (2016). *Semantics*. Chichester: Wiley-Blackwell, 2016.

- Strauss, S., & Stavy, R. (1982). *U-shaped behavioral growth*. [Electronic resource]. New York : Academic Press, 1982. retrieved from <http://ludwig.lub.lu.se/login?url=http://search.ebscohost.com.ludwig.lub.lu.se/login.aspx?direct=true&db=cat01310a&AN=lovisa.004783587&site=eds-live&scope=site>
- van de Weijer, J. (2018). *Joost van de Weijer* [web page]. retrieved from <http://www.sol.lu.se/en/person/JoostvandeWeijer/>, Nov 28, 2018

Appendix

Appendix 1: The CHILDES framework

All corpora with child language that are used for this study are part of a framework named CHILDES. In other words, both the Brown (1973b) corpus and the Carterette and Jones (1974b) corpus are part of this framework. The reference corpora containing adult spoken language, namely the Santa Barbara corpus (Du Bois et al., 2000-2005a-b) and the COCA corpus (Davies, 2008-) are not parts of the CHILDES framework.

CHILDES is an acronym for “Child Language Data Exchange System” (MacWhinney, 2000, p. 9). Besides the child language corpora mentioned above, there are many other corpora in this framework, e.g. *Bates*, *Belfast* and *Bliss* (MacWhinney, 2000, p. 15). The most essential property that the corpora in the CHILDES framework share is a transcription format called CHAT (p. 9). This means that the corpora are annotated in a similar way for grammatical features, which makes it possible for a computer program to analyze them together. A computer program particularly customized for analysis of the CHILDES corpora is CLAN (p. 9). However, that program is not used for my study.

From an inside look of the corpus files of the Brown corpus (1973c), it is obvious how the CHAT transcription system works. For instance, in the file 010600a.cha (Brown, 1973c) from the Eve corpus, present participle is transcribed with the tag “PRESP”, such as in the statement “*Mr Fraser's drinking coffee* .”. In the same sentence, the auxiliary verb *be* is annotated with “aux|be” (see (A1.1) below). On the basis of these annotations, a computer program may search for occurrences of the progressive aspect.

(A1.1)

```
*MOT: Mr Fraser's drinking coffee .
%mor: n:prop|Mr n:prop|Fraser~aux|be&3S part|drink-PRESP n|coffee .
```

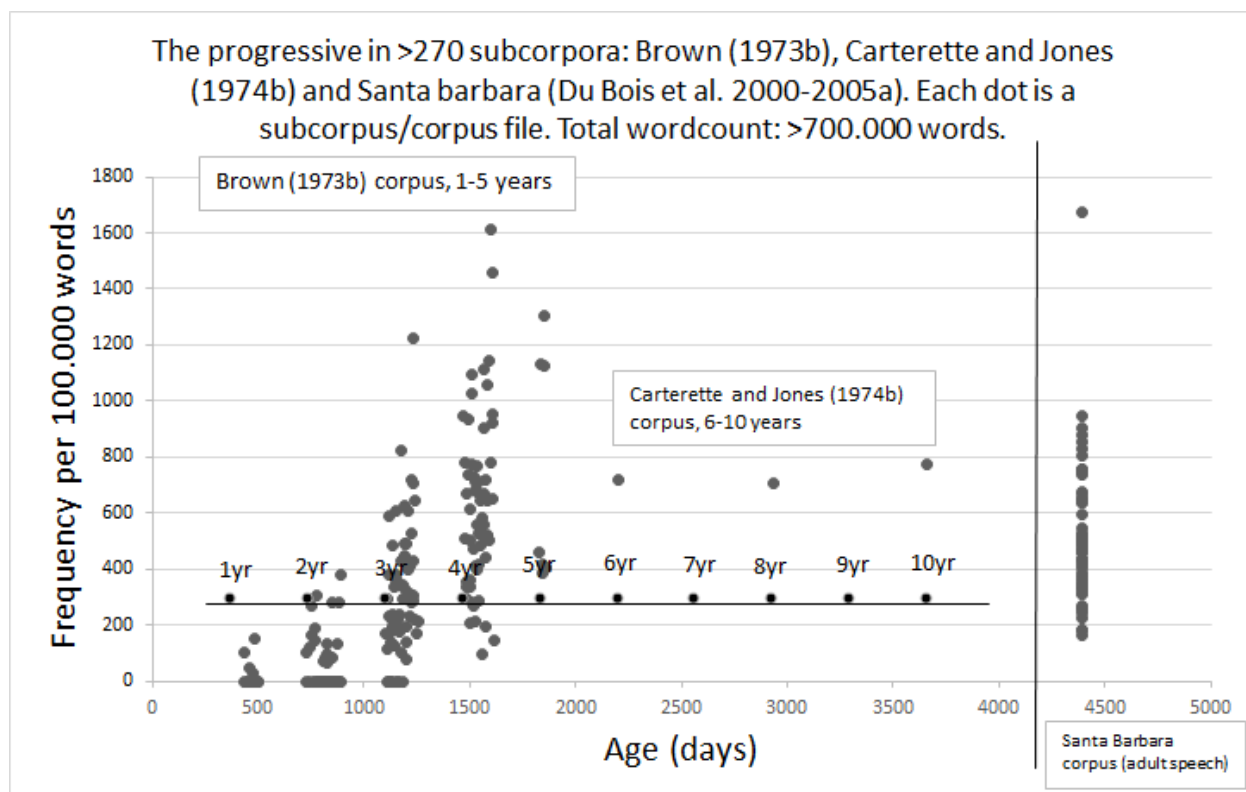
Screen image from the file 010600a.cha in Brown (1973c) corpus as it appears in Google Chrome (Google Inc., 2018).

Appendix 2: Age group distribution

This appendix presents a brief overview of the age group classification system. The age group classification was essentially determined on the basis of visualized data. Using the frequency outputs, I made an age/frequency chart in which each subcorpus was represented by a dot (see Chart A2.1 below). Certain clustering patterns were observed for the Brown (1973b-c) corpus, around the age markers for 1,2,3,4 and 5 years. The first 5 age groups were established based on this distribution. The three obvious age groups for the Carterette and Jones Corpus (1974b) corresponded to 6, 8 and 10 years. A final age category (group 9) for the adult control group was determined. The age groups are summarized in Table 5, p. 23.

A wide distribution was found among the main corpora in terms of word count (see Table 4, p. 21), and the same applies to the sub corpora (see Table A6.1, p. 61). Chart A2.1 therefore does not provide a realistic illustration of general developments of the progressive aspect over time. It is mainly relevant for exhibiting the data clustering phenomenon in the Brown (1973b) corpus.

Chart A2.1. Distribution of progressive frequencies based on sub corpora (i.e. corpus files). Each dot represents a subcorpus / corpus file. It should be noted that the “Age (days)” scale does not apply to the plot for the Santa Barbara corpus, which has been assigned an arbitrary value (4393) to make it possible to include it in the same chart.



Appendix 3: A non-linguistic example of U-Shaped Development

An example of a non-linguistic experiment that reflects U-Shaped development is the *temperature experiment*. Strauss, Ankori, Orpaz, and Stavy (1977), cited by Richards and Siegler (1982, p. 56), showed children of various ages two vessels, which contained water of equal temperature (10°C). All children were asked two hypothetical questions about water mixed from these two vessels into a third container. The questions are paraphrased in (A3.1) below⁵⁶. In case the children produced an incorrect answer to question (A3.1)[2], a follow-up question was asked, as indicated by (A3.1)[2a].

- (A3.1) [1] Would the mixed water be cold?
 [2] What temperature would the mixed water have?
 [2a] In the case of wrong response in [2], the children were asked to motivate their answer.

While all children produced the right answer to (A3.1) [1], only the oldest children, early adolescents (group C)⁵⁷, managed to respond correctly to question (A3.1) [2]. Both the youngest children under 13 years of age (group A) and the children in the intermediate age range (group B) incorrectly believed the water temperature would be the double if the containers were mixed together. However, group B performed worse in the motivation part (A3.1) [2a], claiming the mixed water would be colder than the water in the two first vessels. The youngest children believed it would be “cold just as before” although the temperature numerically would be the double in their view. They did not see the contradiction. A U-formed performance curve was observed. Richards and Siegler (1982, p. 56) explain the contradiction in the response produced by the children in group B with “an analogy of addition producing directional changes (10 + 10 = 20; cold + cold = colder)”. Consequently, it seems plausible that the error tendency in group B was caused by overgeneralization of an additive rule.

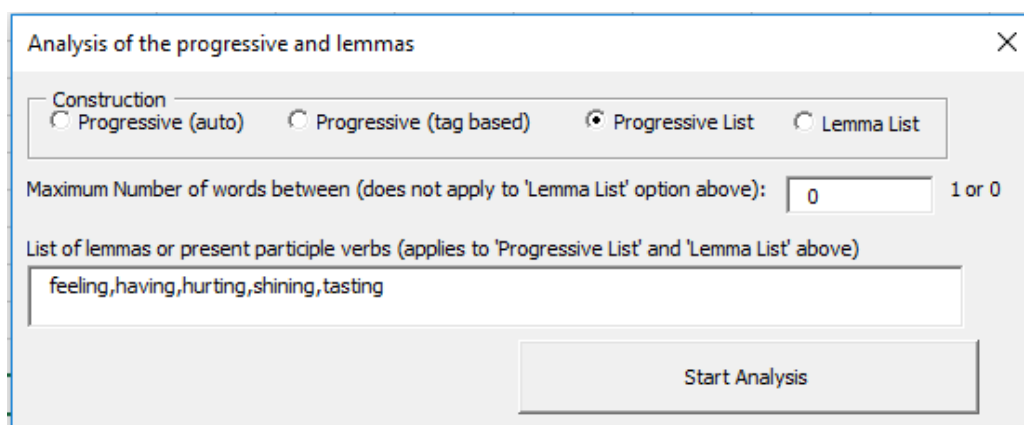
⁵⁶ The numbering of these questions is intended to make this passage easier to follow for the reader. It is not a part of Richards and Siegler’s (1982) account.

⁵⁷ The age group classification (A,B and C) is not mentioned in Richards and Siegler (1982) but added by me to make the account of the experiment easier to follow for the reader.

Appendix 4: Computer Application Window

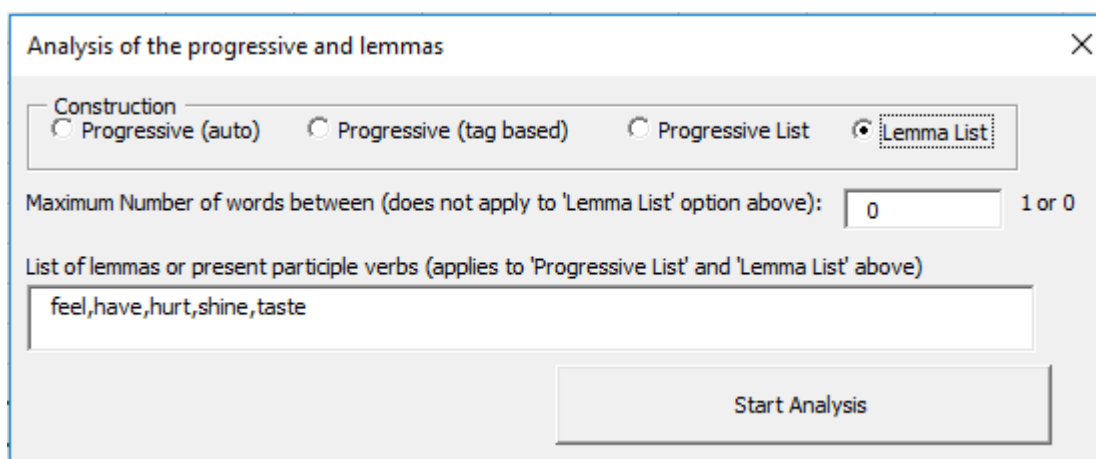
Figure A4.1 and Figure A4.2 are screenshots of the computer application form with different settings. The settings used in these two form views are applied for producing developmental curves for 5 stative verbs based on *frequency index* (see (18), p. 27). See Sections 3.5 and 4.3 for a discussion of methods and results. Figure A4.3 shows how the different user form controls are connected with the source code, which is provided in full in Appendix 18.

Figure A4.1



The screenshot shows a dialog box titled "Analysis of the progressive and lemmas" with a close button (X) in the top right corner. The "Construction" section has four radio buttons: "Progressive (auto)", "Progressive (tag based)", "Progressive List" (which is selected), and "Lemma List". Below this, there is a text input field for "Maximum Number of words between (does not apply to 'Lemma List' option above):" with the value "0" and "1 or 0" to its right. A larger text input field below that contains the text "feeling,having,hurting,shining,tasting". At the bottom right of the dialog is a "Start Analysis" button.

Figure A4.2



The screenshot shows the same dialog box as Figure A4.1, but with the "Lemma List" radio button selected. The text input field for "Maximum Number of words between..." still contains "0". The larger text input field now contains the text "feel,have,hurt,shine,taste". The "Start Analysis" button remains at the bottom right.

Figure A4.3. User control names in VBA code (see Table A4.1 below for numbered key).

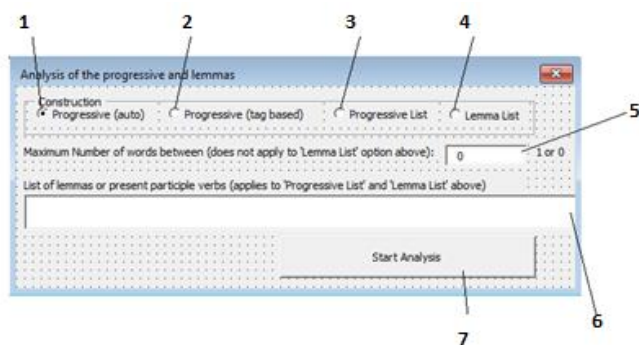


Table A4.1. Numbered key for Figure A4.3 above.

1	xProgressiveAuto
2	xProgressiveTagBased
3	xProgressiveList
4	xLemmaList
5	MaxWordsBetween
6	xListOfLemmasOrProgressives
7	CalculateProgressive

Appendix 5: Computer Application Features

Table A5.1. Summary of the four main computer application features and their relevance for other user form fields

Control	Description	“Maximum Number of words between” is relevant	“List of lemmas or present participle verbs” is relevant
Progressive (auto)	Analysis of the progressive aspect with either 1 or 0 words between the auxiliary verb and the present participle verb (e.g. <i>I am singing</i> vs. <i>I am often singing</i>). The analysis is based on the pattern matching algorithm (see Section 3.4)	✓	
Progressive (tag based)	Similar analysis as for Progressive (auto) above, except it is based on corpus tagging and not the algorithm. Only applicable to the CHILDES corpora, which have CHAT tagging.	✓	
Progressive List	Similar analysis as for Progressive (auto) above, except it is limited to certain verbs specified in the list box.	✓	✓
Lemma List	Analysis of verb lemma frequencies for a list of verb lemmas, such as ‘feel,have’.		✓

Appendix 6: Computer Application Output (Excel)

In Section 3.3, the structure of the Excel Sheet Output from the computer application is described in general terms. In this appendix section, a more detailed account will be provided, as well as a schematic model of the Excel sheet structure (see Table A6.1, p. 61). I will first focus on a single record and its different properties. After presenting this information, I will move on to explaining the entire output shown in Table A6.1. In the final part of this appendix section, verb lists are discussed.

In line with the plan above, we will continue by having a closer look at the entry of the first row in Table A6.1, p. 61. The row number for this row, i.e. 1, is specified in the first column. The entry on the first row, like each of the other rows, represents one corpus file that was analyzed. I often refer to such corpus files as *sub corpora*.

In Table A6.1, each such corpus file or subcorpus (except for the Santa Barbara corpus (Du Bois et al., 2000-2005a)) is related to the computed progressive frequency (column A), time stamp indicating age (column B) and name of subcorpus/file, often equivalent to the name of the child (Column C). Furthermore, Column D contains the approximate word count in the subcorpus. Age in days (column E) refers to specific age. Column F contains relative frequencies per 100.000 words (calculated as $A2 * 100000 / D2$ for row 2). Column H contains age group number, which corresponds to the group division presented in Table 5, p. 23.

Columns I-Z for row 2 contain data that have already been specified for the same row. For example, I2 is same as A2 and J2 equals D2. The reason the results were duplicated in this manner is that it allowed calculation of totals in the rows 272 and 273 at the bottom of the table by just summing up each column. Such calculations necessarily involved the other sub corpora in the worksheet. When taking all corpus files or sub corpora into account (listed from row 2 until row 271 in Table A6.1, p. 61), relative frequencies for age group 1 were calculated based on the total number of words and instances of the progressive for group 1 by creating total sums for the columns I and J.

Above, I have attempted to provide an adequate description of one single row in Table A6.1, which is included below on p. 61 (row 2). While a very similar description applies to all Brown

(1973b) sub corpora, it could not be entirely generalized to Carterette and Jones (1974b). The Santa Barbara corpus (Du Bois et al., 2000-2005a) was even more atypical. For instance, in the Carterette and Jones (1974b) there were only 3 corpus files, each of which represented age specified as year as one digit only. Accordingly, column E reflected age in days for 6, 8 or 10 years for this corpus (2203, 2933 and 3663 days respectively). For the Santa Barbara (Davies, 2008-) control corpus, age was set to an arbitrary value (4393) for all sub corpora, which did not reflect the ages of the participants. I argue that these discrepancies are acceptable since the results were based on a division of age groups rather than exact ages. Hence, it did not matter exactly how old the children were as long as their ages counted in years were known. Similarly, as Santa Barbara was a control corpus, it was not necessary to know the ages of the participants. It was sufficient to classify them as adults and place them in group 9 (see Table 5) .

Table A6.1. Adaptation of an Excel sheet from the program version indicated by A7.1[11]. Due to lack of space, the column headings, given in row 1, are shortened and series of rows and columns are excluded (marked with ...).

Explanations for these column headings are provided in the running text of the current appendix section when relevant.

	A	B	C	D	E	F	H	I	J	K	L	...	Y	Z
1	F	Time Stamp	Corp.	Wcount	Age days	Rel Freq.	Gr.	G(1),F	G(1) WCount	G(2) F	G(2) WCount	...	G(9) F	G(9) WCount
2	1	1;09.00	Eve	3533	473	28.304557	1	1	3533			...		
3	3	1;10.00	Eve	1971	485	152.207002	1	3	1971			...		
4	0	1;10.00	Eve	2062	485	0	1	0	2062			...		
5	0	1;11.00	Eve	1702	497	0	1	0	1702			...		
6	0	1;11.00	Eve	2717	497	0	1	0	2717			...		
7	0	2;00.00	Eve	1868	730	0	2			0	1868	...		
8	2	2;00.00	Eve	1991	730	100.452034	2			2	1991	...		
9	4	2;01.00	Eve	3264	742	122.54902	2			4	3264	...		
10	0	2;01.00	Eve	2911	742	0	2			0	2911	...		
11	7	2;02.00	Eve	4270	754	163.934426	2			7	4270	...		
12	9	2;02.00	Eve	3356	754	268.1764	2			9	3356	...		
13	4	2;03.00	Eve	2731	766	146.466496	2			4	2731	...		

271	29	12;01.01	Santa Barbara	10791	4393	269	9					...	29	287946
272								7	23037	82	97596	...	1492	287946
273									30		84	...		518

Appendix 7: Computer Application Versions

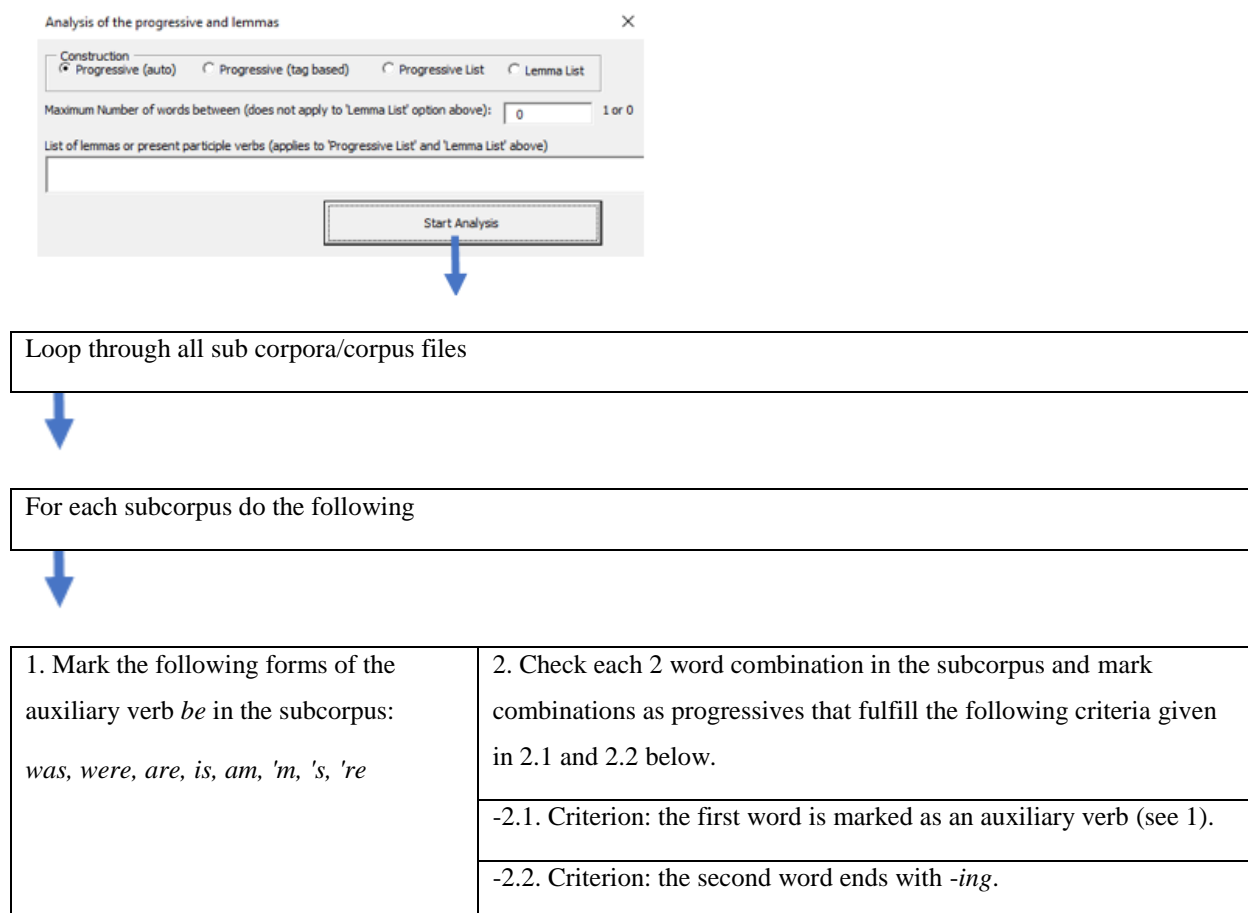
Table A7.1. Different versions of the computer application.

No	Path
1	AdvancedVersion01
2	AdvancedVersion02
3	AdvancedVersion03
4	AdvancedVersion04Old
5	AdvancedVersion05Progressive
6	AdvancedVersion06Lemma
7	AdvancedVersion07VersionWithoutPrespAlone
8	AdvancedVersion08FiveOptions8Verbs
9	AdvancedVersion09Hurting
10	AdvancedVersion10-PValues
11	AdvancedVersion11-GeneralData
12	AdvancedVersion12Having
13	AdvancedVersion13HappeningCorrected
14	AdvancedVersion14HavingCorrected
15	AdvancedVersion15HurtCorrected
16	AdvancedVersion16-8-Statives
17	AdvancedVersion17-5-Statives
18	AdvancedVersion18feel
19	AdvancedVersion19Shine
20	AdvancedVersion20Taste

Appendix 8: Algorithm, simple flowchart

The pattern-matching algorithm of subtype ‘zero words between’ (see (15), p. 24) is essentially carried out by the computer application by locating instances in the corpus texts where a form of the auxiliary verb *be* was followed by a present participle verb. For a flowchart indicating the programmatical steps is presented in Figure A9.1, which includes line number to the function that contains the actual algorithm in Appendix 18. While this function will not be explained in detail, a simplified account of the algorithm is provided in a general flowchart shown in Figure A8.1 below.

Figure A8.1. Simplified flowchart illustrating the progressive algorithm. The different steps of this flowchart will be further explained in the running text of this section.



Essentially, Figure A8.1 above illustrates how the algorithm (subtype ‘zero words between’) works by iterating through all sub corpora that are contained in a certain computer folder. For each

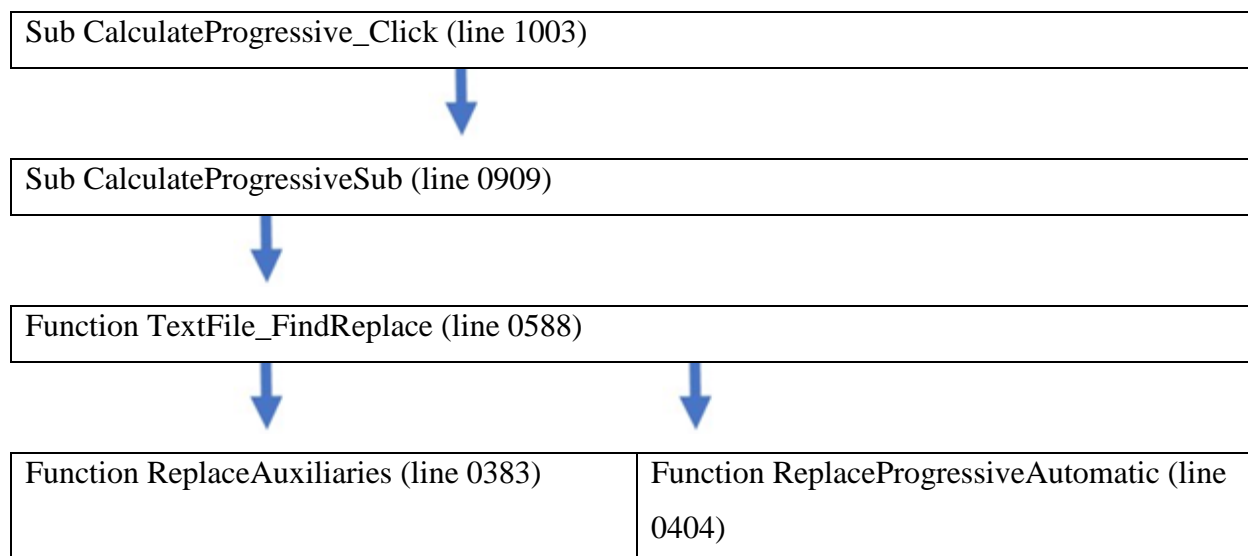
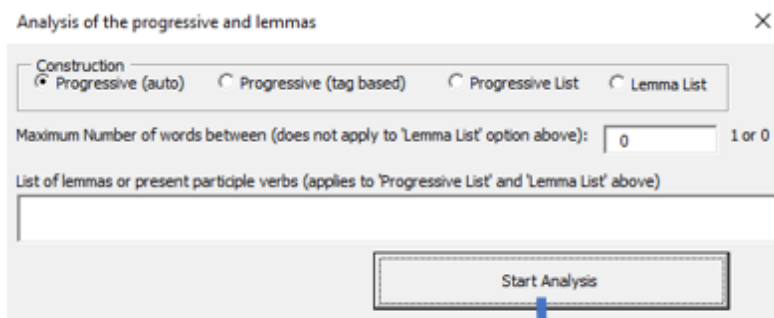
subcorpus, it first marks different forms of the auxiliary verb *be*. Then it checks each 2-word combination. If the first word is marked as an auxiliary and the second word ends with *-ing*, the combination is marked as progressive.

The flowchart shown in Figure A8.1 as well as the related discussion above cover the algorithm subtype ‘zero word between’, which is described in (15) on p. 24. The method for analyzing the other subtype of the algorithm, ‘max one word between’, is very similar. Instead of checking consecutive 2-words combinations, as indicated by Figure A8.1[2], each 2-word-combination is evaluated where the constituents are either consecutive or separated by an arbitrary word. The criteria listed as 2.1 and 2.2 in the same figure are still valid for the second subtype, however.

Appendix 9: Algorithm, advanced flowchart

Please note that Figure A9.1 below refers to line number connected with lines in the source code included in Appendix 18 (e.g. line 1003, line 0909 etc.). The settings indicated by the user form in Figure A9.1 accurately reflect the passage through different subroutines and functions until the execution of the pattern-matching algorithm in the final step. Please note that the code execution as illustrated in this figure primarily takes place from top to bottom and secondarily from left to right. For a detailed description of the source code elements that occur this figure, please refer to Appendix 17.

Figure A9.1



Appendix 10: Algorithm, validation, p-value

This section contains an account on how p-values were calculated to confirm the statistical significance of the progressive algorithm for zero or at most one word between the auxiliary verb and the present participle verb (see Section 3.4). Chart 1 (see p. 30) presents the curves (from age group 1 to 8) for the algorithm analysis and the tag-based analysis side by side for the ‘zero words between’ case. In this chart, the curve marked AUTO represents the outcome from the algorithm. The other curve (marked “TAG”) corresponds to analysis based on corpus tagging. The p-value was calculated with the null hypothesis that tag based frequencies were not correlated to automatic, pattern-matching based frequencies and under the assumption that p was statistically significant only when $p < 0.05$.

The analysis was done in Microsoft Excel 2016, following instructions from a video tutorial by Rebecca Kreider (Kreider, 2016), which described p-value calculation for a situation involving one column with predicted outcomes and another column with actual outcomes. The computational situation appeared to be analogous to the conditions of my study, in which the progressive algorithm predicted frequencies that were calculated based on the corpus tagging (predicted vs. actual outcomes).

In accordance with Kreider’s (2016) tutorial, I added an analytic tool pack to MS Excel 2016 (under Options > Add Ins). The analysis was carried out using the “Data Analysis” command (in the Data > Analysis group). After selecting the regression option in the “Data Analysis” dialog, Excel cell references were added to the text boxes “Input Y Range” and “Input X Range”, in the same order as described in the video tutorial. The p-value calculations given in Chart 1 and Chart 2 (p. 30) were further double-checked by Joost van de Weijer, statistician and researcher at Lund University (van de Weijer, 2018), on November 27, 2018. He confirmed that the calculations were technically correct. Consequently, the correlation between the algorithm prediction and the corpus tagging analysis was statistically significant. Important statistical indicators are provided in Table A10.1 below for the two subtypes.

Table A10.1

	subtype 'zero words between'	Subtype 'max one word between'
estimate of the effect	1.006	1.10
standard error	0.030	0.09
t-value	32.994	12.09
p-value	0.000	0.000

Appendix 11: The COCA corpus - queries, interface etc.

In this appendix section we will address how the searches in the spoken section of the COCA corpus (Davies, 2008-) were done. Please note that when COCA is referred to in this essay, it is always implied that it is the spoken section. There were essentially three types of data that were necessary to acquire from the COCA corpus. These three types are specified in (A11.1) below. The calculations and methods involving obtaining data in this subsection only consider progressive constructions with no words separating the auxiliary verb (forms of *be*) and the present participle verb. Table A11.2 below summarizes the queries used for each type of data specified in (A11.1[A-C]).

(A11.1)	<p>Data necessary to obtain from the COCA corpus, spoken section for this study.</p> <p>A. frequencies of the progressive overall with no words between <i>be</i> and the <i>-ing</i> verb.</p> <p>B. frequencies of the progressive for individual verbs</p> <p>C. frequencies for verb lemmas (each one including all verb forms)</p>
---------	--

Table A11.2. Search queries for different data types.

Data description	Search query	Comments
See (A11.1[A])	[vb*] [v?g*]	This query may be obtained by clicking [POS] to the right of the input box 2 times and then by selecting verb.[BE], then verb.ING.
See (A11.1[B])	For example: _vb* <i>feeling</i>	
See (A11.1[C])	For example: FEEL	

The reason (A11.1[A]) was needed was because it contributed to providing the full picture of development of the progressive from childhood to adulthood. (A11.1[B]) and (A11.1[C]) were necessary for the same reason regarding developments of individual verbs or groups of verbs in relation to all lemma forms of these verbs. This aspect of the study is further addressed in Section 3.5.

Queries for the COCA corpus related to the same verbs as specified in Figure 5, p. 26, are listed in Table A11.3 below. Also, in this case, frequency index may be calculated according to (19), p. 27. Notably, for the COCA corpus (Davies, 2008-), a list of verbs cannot be analyzed together but must be checked separately.

Table A11.3. COCA queries for calculation of frequency index as defined in (19), p. 27.

Queries for progressive constructions	Queries for all lemma forms
_vb* feeling	FEEL
_vb* having	HAVE
_vb* hurting	HURT
_vb* shining	SHINE
_vb* tasting	TASTE

On the basis of Table A11.2, the primary data that were needed could be obtained. Table A11.4 below describes the steps how to apply these queries to obtain data.

Table A11.4

Step	Description
1	Navigate to https://corpus.byu.edu/coca/ after login
2	Click 'Sections' and select 'Spoken' under (1) to the left
3	Do not check the box next to 'Sections'
4	Click 'Options' and 'Group by': Lemmas, and set Hits to 1000
5	Enter the query: <code>_vb* _v?g*</code> (or another query specified in Table A11.2 above).
6	Click 'Find matching strings'

The total size of the spoken section of the COCA corpus, 118 million words, may be obtained from <https://corpus.byu.edu/coca/help/texts.asp> (retrieved Nov 28, 2018). Each frequency generated with the queries in Table A11.2 above appears when following the steps in Table A11.4.

Appendix 12: Samples from the COCA corpus

This appendix lists random examples for the verbs *have* (Table A12.1) and *hurt* (Table A12.2) occurring in the progressive with no words between the form of *be* and the present participle verb.

The examples were retrieved from the COCA corpus (Davies, 2008-) on November 20, 2018, using the KWIC option.

Table A12.1. query: _vb* having

No	Example
1	Sara is having a baby in a month .
2	if you 're having a bad day , the angrier you go to the plate
3	a chronically shy engineer and a museum curator who 's been having a bad-hair day since she was 12
4	Laura Mabbott 's fifth-graders are having a bake sale to protect a coral reef thousands of miles away
5	They 're having a benefit for neuromuscular transplantation .
6	HANNITY : You have been having a big impact on these races .
7	HANNITY : You have been having a big impact on these races .
8	That the girl and the other guy were having a big row and Three-Arm walked in on it .
9	Hartley was having a blast .
10	Two-time U.S. silver medalist Cohen is having a breakthrough season
11	I 'm having a challenging moment .
12	right now we 're having a conflict of interest .
13	NICHOLS# If we 're really , we 're having a contest we sing a couple of songs to pump everybody up
14	And we were having a conversation about the talk that black parents would give
15	In my mind we were having a conversation , but she was saying , " You 're not
16	can tell you I 'm having a damn good time getting there .
17	when he 's having a discussion with someone .
18	SCHINDLER I 'm having a drink , come on in , we 'll have a drink
19	or perhaps was having a drunken hallucination .
20	I think domestically politically it is having a fall out .

Table A12.2. query: _vb* hurting

No	Example	Stative sense
1	Mr-HAMILTON : It 's been hurting a little bit in practice , but today it felt great .	✓
2	WINFREY : Uh-huh . So you are hurting a lot in this relationship .	
3	they 're hurting a lot of people	
4	It is all of our responsibility to notice that children are hurting all along the way .	
5	HODA-KOTB# Mm-Hm. KATHIE-LEE-GIFFORD# People are hurting all over the place .	
6	We are- we are hurting all the time .	
7	This is crazy . It 's hurting America .	
8	It 's hurting Americans ,	
9	This is hurting Americas security .	
10	wait , we 're hurting among white men .	
11	did n't seem to show any empathy for people who are hurting and for what 's going on out there .	
12	hich is n't really going to help someone who 's hurting and in danger of losing their home	
13	It means finding somebody who 's hurting and saying , " What can I do to help ? "	
14	We ca n't go and do this anymore because it 's hurting and tearing me up because you 're taking over my job	
15	and went to a dentist and you 're hurting and they 've got to do it immediately	
16	People are hurting and we need to offer a solution .	
17	I mean , we 're hurting as a unit .	
18	I think it 's because people are hurting at home .	
19	Trade sanctions are hurting badly -	
20	Italy 's 20 percent . And it 's hurting because Americans are known as the big spenderscs	

Appendix 13: Data from the child corpora

Table A13.1. Corpus examples for the two most common verbs in Table 7, p. 35, occurring in the progressive in Brown (1973b) and Carterette and Jones (1974b). The .cha files are included in the compressed files Brown (1973c) and Carterette and Jones (1974c), the latter being abbreviated as C&J in the table. Eve, Adam and Sarah refer to the Brown (1973b) corpus. The trailing dots have been added to mark boundaries between utterances in the original CHAT tagging. Special characters such as & have been removed. The line numbers were found by editing the original corpus files in Notepad ++⁵⁸.

No	Verb form	Corpus example	Child, file reference	Age group
8	having	... what what you was having on you nose ? ... (line 7613)	Eve, filename:020100a.cha	2
9	having	... Sue (.) we're having noodles (line 1869)	Eve, filename:020200b.cha	2
10	having	... we're having supper (line 975)	Sarah, filename:030623.cha	3
11	having	... we were having softball today for PE you know (line 4216)	C&J, fifth.cha	8
12	having	... when I went to my other school and they were -um they were having you know those fights in the saloons (line 6293-6294)	C&J, fifth.cha	8
13	having	... we were having a science discussion (line 13507)	C&J, fifth.cha	8
14	having	... we were having a model Seder you know (line 15827)	C&J, fifth.cha	8
15	having	...and then when they were having art Miss Skator would always go in their room for art (line 17254-17255)	C&J, fifth.cha	8
16	having	... and then they lock the other people in this room and where where they were having dinner and and then one of the boys in the room they that he had the keys and and then he got out but he climbed through the window (line 2488-2491)	C&J, first.cha	6
17	having	... well we were boys were having a race with the boys... (line 10349)	C&J, first.cha	6
18	having	...he's having babies... (line 12069)	C&J, first.cha	6
19	having	and then once -um at my friends house -um -um -um I got this hose you know we were having a water fight . (line 9314-9315)	C&J, third.cha	7
20	hurting	... my neck is hurting (line 887)	Sarah, filename:020802.cha	2
21	hurting	...my head's hurting ... (line 77)	Adam, filename:030529.cha	3
22	hurting	... see this is hurting (line 1325)	Sarah, filename:030926a.cha	3
23	hurting	...I need it on it's hurting my neck... (line 670)	Adam, filename:040511.cha	4
24	hurting	... something's hurting me (line 1186)	Adam, filename:041023.cha	4
25	hurting	... no (.) my back was hurting (line 984)	Sarah, filename:050010.cha	5

⁵⁸ <https://notepad-plus-plus.org/>, Retrieved on December 03, 2018

Table A13.2. Example from *Adam*, filename:041023.cha (Brown , 1973c), age group 4. Underlining is added for the progressive construction with *hurt*.

*URS: wait a minute . (line 1149)
 *URS: I have_to put this on you . (line 1152)
 *URS: let's try .(line 1156)
 *CHI: I'm getting too big . (line 1159)
 *URS: you have_to ask me when you want it off . (line 1162)
 *CHI: I don't have_to need it off .(line 1167)
 *URS: you're just going to keep it on ? (line 1171)
 *CHI: yes . (line 1176)
 *URS: alright . (line 1180)
 *URS: here's your watch . (line 1183)
 *CHI: something's hurting me . (line 1186)
 *CHI: this is it . (line 1189)

Table A13.3. Example from *Adam*, filename:040511.cha (Brown , 1973c), age group 4. Underlining has been added for the progressive example with *hurt*.

*CHI: I need it on . (line 666)
 *CHI: it's hurting my neck . (line 670)
 *URS: is it too tight (.) Adam ? (line 673)
 *CHI: no . (line 676)
 *CHI: now it's not hurting my neck . (line 680)

Table A13.4. Example from *Sarah*, filename: 030110.cha (Brown , 1973c), age group 3. Underlining has been added for the progressive example with *feel*.

*CHI: <ding@o (.) dong@o> [/] (.) ding@o dong@o . (line 1325)
 *CHI: Percy's feeling well . (line 1328)
 *CHI: Percy's feel [//] feeling well . (line 1332)
 *CHI: o:h (.) dat [: that] tickles . (line 1337)

Table A13.5. Example from *Sarah*, filename:040508.cha (Brown , 1973c), Group: 4, with underlining added for *shining* in the progressive.

- *GAI: what are they singing ? (line 1428)
- CHI: I'm shinin(g) in a x x x . (line 1431)
- *CHI: x x x . (line 1434)
- *CHI: (a)n(d) I see dis [: this] is you . (line 1435)

Table A13.6. Example from *Adam*, filename:040511.cha (Brown , 1973c), Group: 4. Underlining has been added for the progressive construction with *shine*.

- *CHI: he's shining his shoes ? (line 252)
- *CHI: are you shining the baby's shoes ? (line 256)

Table A13.7. Example from *Adam*, filename:041023.cha (Brown , 1973c), age group 4. The progressive with the verb *taste* is underlined.

- *CHI: eat it . (line 2746)
- *CHI: hold your hand . (line 2749)
- *CHI: I'm tasting some . (line 2752)
- *CHI: how does it taste ? (2755)

Appendix 14: Output of verb lists

List A14.1. The following list contains 183 verbs occurring in the progressive⁵⁹ in the child corpus Brown (1973b-c), sorted by frequency with stative verbs marked. The list includes false hits since the progressive algorithm was used to extract the data. The same is valid for all lists in this appendix section.

*going(363), getting(64), coming(51), making(39), playing(34), doing(31), looking(26), something(21), putting(20), trying(19), talking(18), eating(17), raining(15), crying(13), taking(12), cutting(11), moving(11), sleeping(11), cooking(10), sticking(10), falling(9), missing(8), turning(8), working(8), driving(7), walking(7), breaking(6), calling(6), happening(6), hiding(6), **hurting**(6), standing(6), starting(6), running(5), telling(5), blowing(4), flying(4), holding(4), jumping(4), laying(4), morning(4), saying(4), sitting(4), swimming(4), thinking(4), writing(4), aching(3), beating(3), being(3), biting(3), building(3), burning(3), carrying(3), drawing(3), fighting(3), giving(3), **having**(3), hitting(3), joking(3), keeping(3), nothing(3), peeking(3), reading(3), smacking(3), swinging(3), tearing(3), touching(3), wagging(3), wearing(3), asking(2), boiling(2), bowling(2), bringing(2), climbing(2), dancing(2), dripping(2), fixing(2), following(2), fooling(2), freezing(2), hopping(2), kicking(2), laughing(2), leaking(2), letting(2), parking(2), picking(2), pointing(2), pulling(2), pumping(2), rolling(2), screaming(2), scribbling(2), shaking(2), **shining**(2), showing(2), slipping(2), someping(2), stirring(2), whistling(2), backing(1), bleeding(1), bombing(1), boxing(1), catching(1), changing(1), checking(1), chewing(1), clapping(1), cleaning(1), closing(1), coloring(1), coughing(1), *ing*(1), covering(1), crawling(1), ding(1), dining(1), dreaming(1), dribbling(1), drinking(1), dropping(1), drowning(1), drying(1), failing(1), **feeling**(1), finding(1), fishing(1), growing(1), guessing(1), hanging(1), hatching(1), helping(1), howling(1), itching(1), kidding(1), knitting(1), knocking(1), learning(1), lightning(1), living(1), marching(1), minding(1), mixing(1), painting(1), passing(1), paying(1), peeling(1), pouring(1), praying(1), pushing(1), remembering(1), resting(1), riding(1), ringing(1), ripping(1), rocking(1), shedding(1), shooting(1), sing(1), skipping(1), snapping(1), snowing(1), soaking(1), speaking(1), spilling(1), splashing(1), squeezing(1), starving(1), staying(1), stopping(1), stucking(1), **tasting**(1), thing(1), tipping(1), tricking(1), using(1), waiting(1), waking(1), washing(1), watching(1), whirling(1), yelling(1)*

⁵⁹ Only progressive constructions with no words separating the form of *be* and the present participle verb are included. The same applies to all lists in this appendix section.

List A14.2. This list contains 226 progressive verbs 1-10 years, in Brown (1973b-c) + Carterette and Jones corpus (1974b-c) including false hits, sorted by frequency.

going(545), getting(73), coming(62), playing(52), making(46), looking(41), doing(39), trying(32), talking(29), something(27), eating(23), putting(21), raining(19), taking(17), sleeping(16), crying(14), moving(13), cutting(12), falling(12), having(12), standing(12), sticking(12), walking(12), cooking(10), working(10), driving(9), laying(9), missing(9), sitting(9), turning(9), hiding(8), holding(8), running(8), starting(8), building(7), happening(7), jumping(7), reading(7), saying(7), screaming(7), swimming(7), telling(7), breaking(6), calling(6), crawling(6), hitting(6), hurting(6), nothing(6), riding(6), thinking(6), watching(6), writing(6), blowing(5), swinging(5), bleeding(4), flying(4), fooling(4), helping(4), kidding(4), laughing(4), morning(4), shaking(4), showing(4), wearing(4), aching(3), beating(3), being(3), biting(3), bringing(3), burning(3), carrying(3), climbing(3), drawing(3), fighting(3), giving(3), joking(3), keeping(3), limping(3), painting(3), peeking(3), picking(3), scratching(3), smacking(3), tearing(3), throwing(3), touching(3), wagging(3), adding(2), arguing(2), asking(2), boiling(2), bowling(2), chasing(2), coughing(2), ing(2), dancing(2), drinking(2), dripping(2), fishing(2), fixing(2), following(2), freezing(2), growing(2), hanging(2), hopping(2), imitating(2), interesting(2), kicking(2), killing(2), leading(2), leaking(2), learning(2), letting(2), listening(2), parking(2), pointing(2), practicing(2), pulling(2), pumping(2), rolling(2), saving(2), scribbling(2), shining(2), slipping(2), snapping(2), someping(2), stirring(2), studying(2), tipping(2), waiting(2), washing(2), whistling(2), yelling(2), backing(1), barking(1), beginning(1), bombing(1), boxing(1), bubbling(1), bucking(1), catching(1), celebrating(1), cellingibratening(1), changing(1), checking(1), chewing(1), clapping(1), cleaning(1), closing(1), coloring(1), covering(1), cracking(1), cussing(1), digging(1), ding(1), dining(1), dreaming(1), dribbling(1), dropping(1), drowning(1), drying(1), dying(1), failing(1), feeling(1), finding(1), giggling(1), guessing(1), hatching(1), howling(1), hurrying(1), itching(1), kissing(1), knitting(1), knocking(1), licking(1), lifting(1), lightning(1), living(1), lying(1), marching(1), minding(1), mixing(1), opening(1), passing(1), paying(1), peeling(1), petting(1), pouring(1), praying(1), pushing(1), remembering(1), resting(1), ringing(1), ripping(1), rocking(1), serving(1), shedding(1), shooting(1), sing(1), singing(1), skipping(1), sniffing(1), snowing(1), soaking(1), speaking(1), speeding(1), spilling(1), splashing(1), sprinkling(1), squeezing(1), squishing(1), staring(1), starving(1), staying(1), steering(1), stopping(1), stucking(1), tasting(1), teaching(1), thing(1), tricking(1), using(1), waking(1), waving(1), whirling(1), wondering(1)

List A14.3. This list contains 383 progressive verbs with frequency within brackets, 1-10 years + adulthood, in Brown (1973b-c) + Carterette and Jones corpus (1974b-c) + the Santa Barbara corpus (Du Bois et al., 2000-2005a), including false hits, sorted by frequency.

going(724), doing(152), getting(133), coming(94), talking(94), trying(90), looking(77), making(63), saying(62), something(62), playing(56), thinking(53), taking(47), sitting(46), having(36), moving(33), working(31), telling(30), nothing(28), putting(28), eating(27), being(24), happening(21), standing(21), wondering(21), raining(20), sleeping(20), starting(19), walking(18), driving(17), reading(17), falling(15), crying(14), cutting(14), calling(13), giving(13), hiding(13), interesting(13), kidding(13), running(13), turning(13), showing(12), sticking(12), cooking(11), laying(11), watching(11), building(10), holding(10), hurting(10), missing(10), asking(9), dying(9), screaming(9), swimming(9), waiting(9), growing(8), leaving(8), living(8), losing(8), paying(8), riding(8), throwing(8), using(8), amazing(7), breaking(7), dancing(7), fighting(7), jumping(7), keeping(7), wearing(7), bleeding(6), blowing(6), carrying(6), crawling(6), drinking(6), feeling(6), flying(6), hitting(6), laughing(6), teaching(6), willing(6), writing(6), beating(5), becoming(5), beginning(5), biting(5), feeding(5), helping(5), listening(5), staying(5), swinging(5), boring(4), bringing(4), changing(4), climbing(4), dripping(4), finding(4), fooling(4), hanging(4), hoping(4), joking(4), killing(4), leaking(4), learning(4), morning(4), opening(4), painting(4), picking(4), pouring(4), pulling(4), shaking(4), studying(4), tearing(4), aching(3), acting(3), anything(3), arguing(3), backing(3), burning(3), cleaning(3), drawing(3), dreaming(3), everything(3), forming(3), kicking(3), letting(3), limping(3), peaking(3), planning(3), pushing(3), rolling(3), sailing(3), scratching(3), shedding(3), singing(3), smacking(3), stirring(3), tempting(3), touching(3), wagging(3), wanting(3), wedding(3), adding(2), assuming(2), baking(2), bandaging(2), boiling(2), bowling(2), bubbling(2), catching(2), charming(2), chasing(2), checking(2), closing(2), coughing(2), ing(2), dealing(2), departing(2), dropping(2), enjoying(2), expecting(2), facing(2), filming(2), fishing(2), fixing(2), following(2), freezing(2), fucking(2), hopping(2), imitating(2), kissing(2), leading(2), manifesting(2), meeting(2), parking(2), passing(2), pointing(2), practicing(2), praying(2), proceeding(2), pumping(2), reaching(2), realizing(2), resting(2), saving(2), scribbling(2), seeking(2), sharing(2), shining(2), slipping(2), snapping(2), someping(2), speeding(2), stealing(2), sucking(2), terrifying(2), thing(2), tipping(2), traveling(2), visiting(2), voting(2), waking(2), washing(2), whistling(2), wrapping(2), yelling(2), aborting(1), absorbing(1), abusing(1), according(1), agreeing(1), annoying(1), anticipating(1), astounding(1), azing(1), balancing(1), banging(1), barking(1), bearing(1), bombing(1), boxing(1), breathing(1), bucking(1), buying(1), carjacking(1), causing(1), celebrating(1), cellingibratening(1), chewing(1), choosing(1), clapping(1), coloring(1), combining(1), commenting(1), compensating(1), competing(1), confusing(1), connecting(1), contributing(1), covering(1), cracking(1), creating(1), creeping(1), crumbling(1), cussing(1), decorating(1), depending(1), depressing(1), digging(1), ding(1), dining(1), disputing(1), dragging(1), dribbling(1), drowning(1), drying(1), during(1), dwibbling(1), eliminating(1), emptying(1), entertaining(1), evolving(1), exasperating(1), existing(1), failing(1), flashing(1), floating(1), flowing(1), fondling(1), frightening(1), giggling(1), guessing(1), guiding(1), guilting(1), hatching(1), healing(1), hearing(1), herding(1), howling(1), humping(1), hurrying(1), hustling(1), imagining(1), intending(1), itching(1), jamming(1), jerking(1), knitting(1), knocking(1), landing(1), lecturing(1),

lending(1), licking(1), lifting(1), lightning(1), lying(1), marching(1), masquerading(1), minding(1), mixing(1), multiplying(1), needing(1), noticing(1), nursing(1), ordering(1), peeling(1), petting(1), pounding(1), preparing(1), racing(1), reasoning(1), recording(1), rediscovering(1), redoing(1), reflecting(1), refreshing(1), remembering(1), requesting(1), resenting(1), returning(1), revealing(1), ringing(1), ripping(1), rocking(1), rotting(1), rubbing(1), scanning(1), scoring(1), selling(1), sending(1), serving(1), shaping(1), shipping(1), shooting(1), shopping(1), shoveling(1), sing(1), skiing(1), skipping(1), slapping(1), slowgaiting(1), slowing(1), smoking(1), sniffing(1), snowing(1), soaking(1), speaking(1), spilling(1), spinning(1), splashing(1), sprinkling(1), squeezing(1), squishing(1), staring(1), starving(1), steering(1), stopping(1), stringing(1), sticking(1), stumbling(1), subtracting(1), suffering(1), suggesting(1), tasting(1), teaming(1), teething(1), testifying(1), tracking(1), training(1), transferring(1), tricking(1), unwilling(1), upsetting(1), varying(1), warming(1), wavering(1), waving(1), whirling(1), wiggling(1), winning(1), witnessing(1), worrying(1), wrestling(1)

Appendix 15: Progressive frequency tables

Table A15.1 shows absolute frequencies of the progressive aspect (for ‘zero words between’, see Section 3.4) and approximated total word count for each age group (see Table 5, p. 23) based on the child language corpora (Brown (1973b) and Carterette and Jones (1974b) and the COCA corpus, spoken section (Davies, 2008-). Notably, the Santa Barbara corpus (Du Bois et al., 2000-2005a) is not found in the table. It also shows frequencies (and all associated verb forms) for the 5 stative verb specified in Section 4.3. Data for the two most frequent of these verbs are included in A15.2 and A15.3 below. The term frequency index, which occurs in the descriptions of all tables in this appendix section, is defined in (18) and (19) on p. 27.

Table A15.1. Age group 9 refers to the spoken section of the COCA corpus, which is the adult control group. See e.g. Sections 3.3 and 4.3 for more information.

age group	progressive freq. (general)	5 stative verbs (progressive freq.)	5 stative verbs (all verb forms)	frequency index (5 stative verbs)	total word count
1	7	0	16	0.00	23049
2	82	3	182	1.65	97596
3	373	4	276	1.45	125743
4	621	5	261	1.92	104407
5	70	1	16	6.25	11166
6	144	3	145	2.07	20005
7	152	1	131	0.76	21519
8	209	5	204	2.45	27015
9	993066	14613	1906166	0.77	118167133

Table A15.2. Frequencies for the verb *have* in the progressive (compare Table A15.1). The numbers in the table are computed based on Brown (1973b), Carterette and Jones (1974b) and the COCA corpus, spoken section (Davies, 2008-) (cf. Table A15.3).

A. age group	B. have (prog. freq.)	C. have (all verb forms)	D. Frequency index (B*100/C)
1	0	13	0.00
2	2	70	2.86
3	1	180	0.56
4	0	204	0.00
5	0	14	0.00
6	3	140	2.14
7	1	124	0.81
8	5	196	2.55
9	10493	1791180	0.59

Table A15.3. Frequencies for the verb *hurt* in the progressive (compare Table A15.1) with frequency index, adjusted frequency index and relative frequencies. The frequency adjustments only affect age group 9 (see discussion in Section 5.3). The data are based on Brown (1973b), Carterette and Jones (1974b) and the COCA corpus, spoken section (Davies, 2008-).

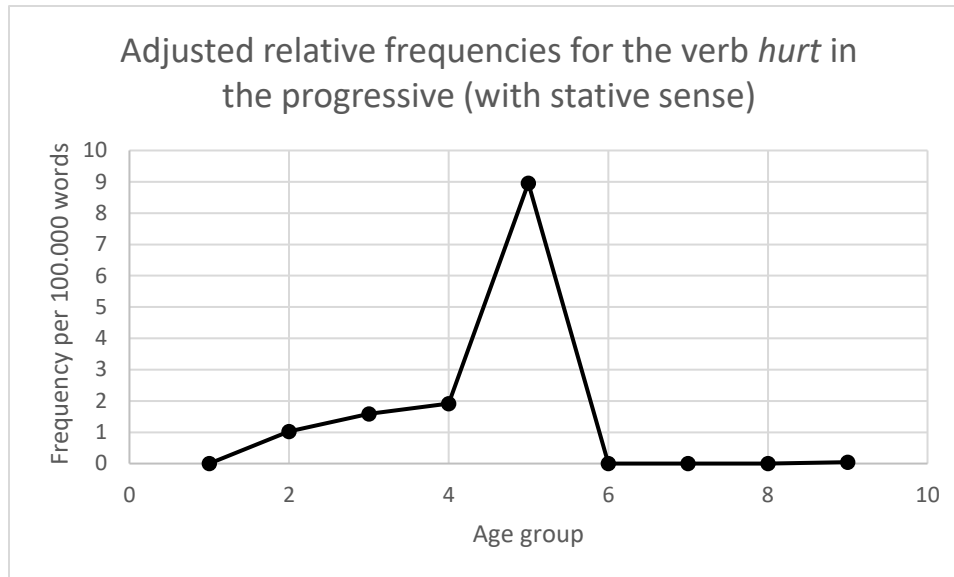
A. age group	B. <i>hurt</i> (progressive freq.)	C. <i>hurt</i> (all verb forms)	D. Frequency index (B*100/C)	E. Adjusted frequency index ⁶⁰	F. Corpus size (for age group)	G. Adjusted relative frequency ⁶¹
1	0	3	0.00	0	23037	0
2	1	95	1.05	1.05	97596	1.024632157
3	2	86	2.33	2.33	125743	1.590545796
4	2	38	5.26	5.26	104407	1.915580373
5	1	2	50.00	50.00	11166	8.955758553
6	0	5	0.00	0	20005	0
7	0	3	0.00	0	21519	0
8	0	5	0.00	0	27015	0
9	1109	16201	6.85	0.3425 (6.85)	118000000 ⁶²	0.05 (0.94)

⁶⁰ See discussion in Section 5.3. Data before adjustment are within brackets.

⁶¹ Relative frequency of *hurt* in the progressive per 100.000 words after frequency adjustment. Calculated as $B*100000/F$ (with columns B and F) before adjustment. Data before adjustments are placed within brackets.

⁶² This figure was obtained from <https://corpus.byu.edu/coca/help/texts.asp>, 2018, Nov 25.

Chart A15.1. Adjusted relative frequencies for *hurt* in the progressive with stative interpretation. Based on column G in Table A15.3 above.



Appendix 16: The VBA object model

The computer application was developed in VBA for Microsoft Excel 2016 using the object model (cf. Microsoft, 2018a). The object model allows various worksheet manipulations through events, methods and properties of the *Application object* (cf. Microsoft, 2018b). Data may, for instance, be populated in cells of Excel worksheets using the *Application.Worksheets* property (cf. Microsoft, 2018c). That property represents a *sheets* collection (Microsoft, 2018c), which typically contains *Worksheet* objects (Microsoft, 2018d).

Like the application object described above, the *Worksheet* object has events, methods and properties (Microsoft, 2018e). One of its properties is the *cells* property (Microsoft, 2018f). This property may be specified with row and column. For example, the VBA command given in (A16.1) below, gives the result in Figure A16.2 below.

(A16.1) *Application.Worksheets(1).Cells(1, 1) = 1*

Figure A16.2. Screen shot from the relevant Microsoft Excel 2016 Worksheet, after running the command in (A16.1).

	A	B
1	1	
2		

The object model (Microsoft, 2018a) is highly relevant for the source code of the computer application, which is described in Appendix 17. Familiarity with this model is essential to technical understanding of relations between inputs (i.e. user form input and corpus files), program code and outputs (Excel sheets and data files).

Appendix 17: Source code description

This appendix section contains a brief technical description of the computer application source code listed in Appendix 18. All line number references in the running text refer to that appendix. There are four options in the user form (see Figure 2, p. 22). I will go through them one by one and describe, in a simplified way, the main steps involved.

Section A17.A. Progressive (auto)

Progressive (auto) is the first option from the left on the user form. When selecting this option and clicking ‘Start Analysis’, the code enters *CalculateProgressive_Click* (line 1003)⁶³. The execution continues next with *CalculateProgressiveSub* (line 0909).

When looping through all corpus files, each corpus file is processed by *TextFile_FindReplace* (line 0588). After various data collection procedures and content standardizations, the code reaches *ReplaceAuxiliaries* (line 0383). This procedure marks most forms of the auxiliary verb *be* with a tag. Finally, a procedure named *ReplaceProgressiveAutomatic* (line 0404) is executed (for a

⁶³ The same procedure is entered for all options when clicking the same button in the user form.

corpus file in the same loop). What it does is that it loops through all word pairs in the current corpus file in the main loop and checks each pattern if it matches the progressive pattern, with accommodations made for the two subtypes ‘zero words between’ and ‘max one word between’. After exiting the main loop in *CalculateProgressiveSub*, some additional outputs are made to files and sheets.

Section A17.B. Progressive (tag based)

This is the second option from the left on the user form. When selecting this and clicking ‘Start Analysis’, the code enters *CalculateProgressive_Click* (line 1003). The execution continues next with *CalculateProgressiveSubTaggedAlgorithm* (line 0850).

Looping through the corpus files, each corpus file is processed by *TextFile_FindReplace_TAGGED_ALGORITHM* (line 0204). It counts the number of instances of the progressive aspect based on corpus tagging in the function *CountProgressivesInCHATTaggedText* (line 0137), which involves pairwise comparisons of corpus tagging markers (such as -PRES) with zero or maximally one word in between. On completion of the main loop, some additional file and Excel outputs are generated.

Section A17.C. Progressive list

This is the third option from the left on the user form. After ‘Start Analysis’ click, the execution point moves to *CalculateProgressive_Click* (line 1003). The execution continues next with *CalculateProgressiveSub* (line 0909). Looping through all corpus files, each corpus file is processed by *TextFile_FindReplace* (line 0588). After data collection procedures and standardizations, the code execution point gets to *ReplaceAuxiliaries* (line 0383). This procedure marks forms of *be* with a tag. Then *ReplaceProgressiveAutomatic* (line 0404) is executed. It loops through the word pairs in the current subcorpus. It marks the progressive pattern ONLY if the second word matches an item specified in the verb list, which has been entered by the user in the

user form (line 0442). It also accommodates for the second subtype ‘max one word between’. After exiting the main loop in *CalculateProgressiveSub*, some additional outputs are made.

Section A17.D. Lemma list

This is the fourth option from the left. As indicated by the footnotes, the selections of identifier names do not always reflect the task for this option. The code first enters *CalculateProgressive_Click* (line 1003). The execution goes on with *CalculateProgressiveSubTaggedAlgorithm* (line 0850)⁶⁴. Looping through the corpus files, each corpus file is processed by *TextFile_FindReplace_TAGGED_ALGORITHM* (line 0204)⁶⁵. It counts instances of all lemma forms in the function *CountProgressivesInCHATTaggedText* (line 0137 and line 0189)⁶⁶. On completion of the outer loop, additional outputs are carried out.

⁶⁴ Although *CalculateProgressiveSubTaggedAlgorithm* implies computation of the progressive, in this case this will not be done. Instead, instances of all forms of verb lemmas are counted.

⁶⁵ *TextFile_FindReplace_TAGGED_ALGORITHM* implies that an algorithm based on corpus tagging is applied. This will not be done in this instance. Instead, all lemma forms will be counted.

⁶⁶ The name *CountProgressivesInCHATTaggedText* does not accurately reflect the task, which concerns computations of lemma forms rather than frequencies of the progressive.

Appendix 18: Source code

This appendix contains the source code of the computer application in the program version specified by Table A7.1[11]. For a summarized description related to this code, see Appendix 17. The code resides behind the user form object demonstrated in Figure A4.3, p. 57, with the user controls numbered. A key to the control names is provided in Table A4.1.

```
[0001]
[0002] 'This computer application
[0003] 'was created by Carl-Staffan Svenbro in 2018/2019 for
[0004] 'his D-level Essay project in English linguistics
[0005] 'at Lund University
[0006]
[0007]
[0008]
[0009]
[0010]
[0011] Option Explicit
[0012]
[0013] Private Const ProjectFilePath = "/FileLibrary/"
[0014] Private Const EnumProgressiveVerbList = 10000
[0015] Private Const EnumLemmaList = 10001
[0016] Private Const EnumRegular = 10002
[0017]
[0018] Public Function RemoveNewLineAndDoubleSpace(xStr As String) As String
[0019] xStr = Replace(xStr, Chr(13), " ", , , vbBinaryCompare)
[0020] xStr = Replace(xStr, Chr(10), " ", , , vbBinaryCompare)
[0021] xStr = Replace(xStr, Chr(13), " ", , , vbBinaryCompare)
[0022] xStr = Replace(xStr, Chr(10), " ", , , vbBinaryCompare)
[0023] xStr = Replace(xStr, " ", " ", , , vbBinaryCompare)
[0024] xStr = Replace(xStr, " ", " ", , , vbBinaryCompare)
[0025] RemoveNewLineAndDoubleSpace = xStr
[0026] End Function
[0027]
[0028]
[0029]
[0030]
[0031]
[0032]
[0033] Public Function OutputResultsGetGroupTAGGEDALG _
[0034] (TextBody As String, ChildDate As String, ChildName As String, _
[0035] WordCount As Long, StatSheet As String, PROGRESSIVECount As Long) As Integer
[0036] Dim xStr As String
[0037] Dim i As Long
[0038] Dim j As Long
[0039] Dim WordString As String
[0040] Dim WStat As Worksheet
[0041] Dim WStatRow As Long
[0042] Dim ChildDateArray() As String
[0043] Dim days As Integer
[0044] Dim Counter As Long
[0045] Dim GroupNo As Long
[0046]
[0047]
[0048] xStr = " " + TextBody + " "
[0049]
[0050] Counter = PROGRESSIVECount
[0051]
[0052] Set WStat = Worksheets(StatSheet)
```

```

[0053] WStatRow = 1
[0054] Do While WStat.Cells(WStatRow, 1) <> ""
[0055]     WStatRow = WStatRow + 1
[0056] Loop
[0057]
[0058] WStat.Cells(WStatRow, 1) = Counter
[0059] WStat.Cells(WStatRow, 2) = ChildDate
[0060] WStat.Cells(WStatRow, 3) = ChildName
[0061] WStat.Cells(WStatRow, 4) = WordCount
[0062]
[0063] ChildDate = Replace(ChildDate, ".", ";", , , vbBinaryCompare)
[0064] ChildDateArray = Split(ChildDate, ";")
[0065]
[0066] days = Val(ChildDateArray(0)) * 365 + _
[0067] Val(ChildDateArray(1)) * 12 + Val(ChildDateArray(2))
[0068]
[0069] WStat.Cells(WStatRow, 5) = days
[0070] WStat.Cells(WStatRow, 6) = "=A" + Trim(Str(WStatRow)) _
[0071] + "*100000/D" + Trim(Str(WStatRow))
[0072]
[0073] GroupNo = GetGroupNumberForDays(days)
[0074] WStat.Cells(WStatRow, 8) = GroupNo
[0075] If GroupNo = 1 Then WStat.Cells(WStatRow, 9) = Counter
[0076] If GroupNo = 1 Then WStat.Cells(WStatRow, 10) = WordCount
[0077] If GroupNo = 2 Then WStat.Cells(WStatRow, 11) = Counter
[0078] If GroupNo = 2 Then WStat.Cells(WStatRow, 12) = WordCount
[0079] If GroupNo = 3 Then WStat.Cells(WStatRow, 13) = Counter
[0080] If GroupNo = 3 Then WStat.Cells(WStatRow, 14) = WordCount
[0081] If GroupNo = 4 Then WStat.Cells(WStatRow, 15) = Counter
[0082] If GroupNo = 4 Then WStat.Cells(WStatRow, 16) = WordCount
[0083] If GroupNo = 5 Then WStat.Cells(WStatRow, 17) = Counter
[0084] If GroupNo = 5 Then WStat.Cells(WStatRow, 18) = WordCount
[0085]
[0086]
[0087]
[0088] If GroupNo = 6 Then WStat.Cells(WStatRow, 19) = Counter
[0089] If GroupNo = 6 Then WStat.Cells(WStatRow, 20) = WordCount
[0090] If GroupNo = 7 Then WStat.Cells(WStatRow, 21) = Counter
[0091] If GroupNo = 7 Then WStat.Cells(WStatRow, 22) = WordCount
[0092] If GroupNo = 8 Then WStat.Cells(WStatRow, 23) = Counter
[0093] If GroupNo = 8 Then WStat.Cells(WStatRow, 24) = WordCount
[0094]
[0095]
[0096]
[0097]
[0098]     OutputResultsGetGroupTAGGEDALG = GroupNo
[0099] End Function
[0100]
[0101]
[0102]
[0103]
[0104]
[0105]
[0106] Function xxxxxGetPROGRESSIVEForChildTAGGEDAlgorithm(xStr As String) As String
[0107]     Dim MySheet As Worksheet
[0108]     Dim CurrentRow As Long
[0109]     Dim Lemma As String
[0110]     Dim Progressive As String
[0111]     Dim Output As String
[0112]     Dim WordArray() As String
[0113]     Dim j As Long
[0114]     Dim NewStr As String
[0115]     NewStr = ""
[0116]     Output = xStr
[0117]     Output = LCase(Output)
[0118]
[0119]     Output = Replace(Output, "%mor:" + Chr(9), "mor", , , vbBinaryCompare)
[0120]     Output = Replace(Output, Chr(13) + Chr(9), " ", , , vbBinaryCompare)
[0121]     Output = Replace(Output, Chr(10) + Chr(9), " ", , , vbBinaryCompare)
[0122]     Output = Replace(Output, Chr(9), "%", , , vbBinaryCompare)

```

```

[0123] Output = Replace(Output, "*", "%", , , vbBinaryCompare)
[0124]
[0125] WordArray = Split(Output, "%", , vbBinaryCompare)
[0126] For j = LBound(WordArray) + 1 To UBound(WordArray)
[0127]     If WordArray(j) = "chi:" Then NewStr = NewStr + WordArray(j + 1)
[0128]
[0129] Next j
[0130] NewStr = Replace(NewStr, " ", " ", , , vbBinaryCompare)
[0131] NewStr = Replace(NewStr, " ", " ", , , vbBinaryCompare)
[0132] NewStr = Replace(NewStr, " ", " ", , , vbBinaryCompare)
[0133]
[0134] xxxxxGetPROGRESSIVEForChildTAGGEDAlgorithm = NewStr
[0135] End Function
[0136]
[0137] 'LINECHECK 0137
[0138] Function CountProgressivesInCHATTaggedText(ByRef CHATTaggedText As String, _
[0139] NumberOfWordsBetween As Integer, CurrentMode As Long, LemmaList As String) As Long
[0140] Dim StringPartArray() As String
[0141] Dim j As Long
[0142] Dim xCount As Long
[0143] Dim OutputString As String
[0144] Dim WordMinus1Aux As Boolean
[0145] Dim WordMinus2Aux As Boolean
[0146] OutputString = ""
[0147] xCount = 0
[0148] LemmaList = "," + Replace(LCase(LemmaList), " ", "", , , vbBinaryCompare) + ","
[0149] LemmaList = Replace(LemmaList, ",", " ", , , vbBinaryCompare)
[0150] CHATTaggedText = LCase(CHATTaggedText)
[0151] CHATTaggedText = Replace(CHATTaggedText, "aux|be", "aux#be", , , vbBinaryCompare)
[0152]
[0153] If CurrentMode = EnumRegular Then
[0154] CHATTaggedText = Replace(CHATTaggedText, "&", "", , , vbBinaryCompare)
[0155] CHATTaggedText = Replace(CHATTaggedText, "&", "", , , vbBinaryCompare)
[0156] End If
[0157]
[0158] If CurrentMode = EnumLemmaList Then
[0159] CHATTaggedText = Replace(CHATTaggedText, "&", "|", , , vbBinaryCompare)
[0160] CHATTaggedText = Replace(CHATTaggedText, "&", "|", , , vbBinaryCompare)
[0161] CHATTaggedText = Replace(CHATTaggedText, "-", "|", , , vbBinaryCompare)
[0162] CHATTaggedText = Replace(CHATTaggedText, "-", "|", , , vbBinaryCompare)
[0163] End If
[0164] If CurrentMode <> EnumRegular And CurrentMode <> EnumLemmaList Then
[0165] MsgBox ("Warning: Current mode is neither regular nor lemma. process will terminate")
[0166] End
[0167] End If
[0168] StringPartArray = Split(CHATTaggedText, "|", , vbBinaryCompare)
[0169]
[0170] For j = LBound(StringPartArray) + 1 + NumberOfWordsBetween To UBound(StringPartArray)
[0171]
[0172] If CurrentMode = EnumRegular Then
[0173] WordMinus1Aux = (InStr(1, StringPartArray(j - 1), "aux#be", vbBinaryCompare) > 0)
[0174] WordMinus2Aux = False
[0175] If NumberOfWordsBetween = 1 Then WordMinus2Aux = _
[0176] (InStr(1, StringPartArray(j - 2), "aux#be", vbBinaryCompare) > 0)
[0177]
[0178]
[0179] If InStr(1, StringPartArray(j), "-presp", vbBinaryCompare) > 0 And _
[0180] (WordMinus1Aux Or WordMinus2Aux) Then
[0181] xCount = xCount + 1
[0182] StringPartArray(j) = "&" + StringPartArray(j)
[0183] End If
[0184]
[0185]
[0186]
[0187] End If
[0188]
[0189] 'LINECHECK 0189
[0190] If CurrentMode = EnumLemmaList Then
[0191] If InStr(1, LemmaList, "," + LCase(StringPartArray(j)) + ",", vbBinaryCompare) > 0 Then
[0192] xCount = xCount + 1

```



```

[0193]         StringPartArray(j) = "&" + StringPartArray(j)
[0194]     End If
[0195] End If
[0196]
[0197]     OutputString = "|" + OutputString + StringPartArray(j - 1) + "|"
[0198]     Next j
[0199]     OutputString = OutputString + StringPartArray(UBound(StringPartArray))
[0200]
[0201] CHATTaggedText = Replace(OutputString, "|", "", , , vbBinaryCompare)
[0202] CountProgressivesInCHATTaggedText = xCount
[0203] End Function
[0204] 'LINECHECK 0204
[0205] Public Function TextFile_FindReplace_TAGGED_ALGORITHM(FilePathInput As String, _
[0206]     JustFileName As String, FilePathOutput As String, StatSheet As String, _
[0207]     FileSuffix As String, ByRef TotalTaggedProgressiveString, _
[0208]     NumberOfWordsBetween As Integer, CurrentMode As Long, LemmaList As String) As String
[0209] Dim TextFile As Integer
[0210] Dim i As Integer
[0211] Dim FileContent As String
[0212] Dim ChildDate As String
[0213] Dim ChildName As String
[0214] Dim WordCount As Long
[0215] Dim GroupNo As Integer
[0216] Dim PROGRESSIVECountText As String
[0217] Dim PROGRESSIVECount As Long
[0218] Dim OldFilePathInput As String
[0219] PROGRESSIVECountText = ""
[0220] PROGRESSIVECount = -1
[0221] TextFile = FreeFile
[0222] Open FilePathInput For Binary As TextFile
[0223] FileContent = Space(LOF(TextFile))
[0224] Get #TextFile, , FileContent
[0225] Close TextFile
[0226] FileContent = xxxxxMakeLowercase(FileContent)
[0227]
[0228] FileContent = xxxxxStandardize(FileContent)
[0229]
[0230]
[0231] ChildDate = xxxxxGetDate(FileContent, False)
[0232] ChildName = xxxxxGetName(FileContent, False)
[0233] PROGRESSIVECountText = xxxxxGetPROGRESSIVEForChildTAGGEDAlgorithm(FileContent)
[0234] FileContent = xxxxxGetForChild(FileContent)
[0235]
[0236] PROGRESSIVECount = CountProgressivesInCHATTaggedText(PROGRESSIVECountText _
[0237]     , NumberOfWordsBetween, CurrentMode, LemmaList)
[0238] WordCount = GetStringCount(FileContent, " ")
[0239] FileContent = xxxxxRemoveSpecialCharacters(FileContent)
[0240] PROGRESSIVECountText = RemoveNewLineAndDoubleSpace(PROGRESSIVECountText)
[0241]
[0242] TextFile = FreeFile
[0243] OldFilePathInput = FilePathOutput
[0244] FilePathOutput = Replace(FilePathOutput, ".txt", FileSuffix + ".txt", , , vbBinaryCompare)
[0245] Open FilePathOutput For Output As TextFile
[0246]
[0247] Print #TextFile, FileContent
[0248]
[0249] Close TextFile
[0250]
[0251]
[0252]
[0253] FilePathOutput = Replace(OldFilePathInput, ".txt", FileSuffix + "-TAGGED" + ".txt" _
[0254]     , , , vbBinaryCompare)
[0255] Open FilePathOutput For Output As TextFile
[0256]
[0257] Print #TextFile, PROGRESSIVECountText
[0258]
[0259] Close TextFile
[0260]
[0261]
[0262] GroupNo = OutputResultsGetGroupTAGGEDALG(FileContent, ChildDate, _

```

```

[0263]     ChildName, WordCount, StatSheet, PROGRESSIVECount)
[0264]
[0265] FileContent = FileContent + Chr(10) + Chr(13) + "Corpus " + ChildName + _
[0266]     ", filename:" + JustFileName + ", Group:" + Str(GroupNo) + Chr(10) + _
[0267]     Chr(13) + Chr(10) + Chr(13) + Chr(10) + Chr(13) + Chr(10) + Chr(13)
[0268]
[0269] TotalTaggedProgressiveString = TotalTaggedProgressiveString + _
[0270]     PROGRESSIVECountText + Chr(10) + Chr(13) + "Corpus " + ChildName _
[0271]     + ", filename:" + JustFileName + ", Group:" + Str(GroupNo) + Chr(10) _
[0272]     + Chr(13) + Chr(10) + Chr(13) + Chr(10) + Chr(13) + Chr(10) + Chr(13)
[0273] TextFile_FindReplace_TAGGED_ALGORITHM = FileContent
[0274] End Function
[0275]
[0276]
[0277]
[0278]
[0279]
[0280]
[0281]
[0282]
[0283]
[0284]
[0285] Public Function ReadTextBodyFromFile(fileName As String)
[0286]     Dim FileContent As String
[0287]     Dim TextFile As Integer
[0288]     Dim xPath As String
[0289]     If InStr(1, fileName, "\", vbBinaryCompare) > 0 Or InStr(1, fileName, "/", vbBinaryCompare) > 0 Then MsgBox ("Error: 445")
[0290]     TextFile = FreeFile
[0291]
[0292]     xPath = ThisWorkbook.Path + ProjectFilePath + fileName
[0293]     Open xPath For Binary As TextFile
[0294]     FileContent = Space(LOF(TextFile))
[0295]     Get #TextFile, , FileContent
[0296]     Close TextFile
[0297]     ReadTextBodyFromFile = FileContent
[0298] End Function
[0299]
[0300]
[0301]
[0302] Public Sub WriteTextBodyToFile(fileName As String, FileContent As String)
[0303]     Dim TextFile As Integer
[0304]     Dim xPath As String
[0305]     TextFile = FreeFile
[0306]     If InStr(1, fileName, "\", vbBinaryCompare) > 0 Or InStr(1, fileName, "/", vbBinaryCompare) > 0 Then MsgBox ("Error: 446")
[0307]     xPath = ThisWorkbook.Path + ProjectFilePath + fileName
[0308]     Open xPath For Output As TextFile
[0309]
[0310]     Print #TextFile, FileContent
[0311]
[0312]     Close TextFile
[0313] End Sub
[0314]
[0315] Function ReplaceProgressiveBasedOnList(xStr As String, MinRow As Long, MaxRow As Long) As String
[0316]     Dim MySheet As Worksheet
[0317]     Dim CurrentRow As Long
[0318]     Dim Lemma As String
[0319]     Dim Progressive As String
[0320]     Dim Output As String
[0321]
[0322]     Output = xStr
[0323]
[0324]     Set MySheet = ThisWorkbook.Worksheets("Progressive")
[0325]
[0326]     For CurrentRow = MinRow To MaxRow
[0327]         Lemma = MySheet.Cells(CurrentRow, 1)
[0328]         Progressive = MySheet.Cells(CurrentRow, 2)
[0329]         Output = Replace(Output, " AUX# " + Progressive, " AUX# " + Progressive + "&")
[0330]     Next CurrentRow
[0331]     ReplaceProgressiveBasedOnList = Output
[0332]

```

```

[0333]
[0334]
[0335] End Function
[0336]
[0337] Function TransformRawSpokenCorpusSantaBarbara(xStr As String) As String
[0338]     Dim Output As String
[0339]     Dim i As Long
[0340]     Dim CharacterRemoveArray(100) As String
[0341]     Dim ReplaceChar As String
[0342]     Dim OldOutput As String
[0343]     Dim StrArray() As String
[0344]     Dim DoRemove As Boolean
[0345]
[0346]     Output = xStr
[0347]
[0348]
[0349]     For i = 1 To 9
[0350]         Output = Replace(Output, Trim(Str(i)), "0", , , vbBinaryCompare)
[0351]     Next i
[0352]     Output = Replace(Output, "0", ":", , , vbBinaryCompare)
[0353]     StrArray = Split(Output, ":", , vbBinaryCompare)
[0354]     For i = LBound(StrArray) To UBound(StrArray)
[0355]         If StrArray(i) = UCase(StrArray(i)) Then StrArray(i) = " "
[0356]     Next i
[0357]     Output = Join(StrArray, " ")
[0358]
[0359]
[0360]     OldOutput = Output
[0361]
[0362]
[0363]     Do
[0364]         OldOutput = Output
[0365]         Output = Replace(Output, " ", " ", , , vbBinaryCompare)
[0366]     Loop Until OldOutput = Output
[0367]
[0368]
[0369]
[0370]     TransformRawSpokenCorpusSantaBarbara = LCase(Output)
[0371]
[0372] End Function
[0373]
[0374]
[0375] Public Function GetStringCount(xStr As String, CountedSubString As String) As Long
[0376]     Dim Temp As String
[0377]     Dim Temparray() As String
[0378]     Temp = xStr
[0379]     Temparray = Split(Temp, CountedSubString)
[0380]     GetStringCount = UBound(Temparray)
[0381] End Function
[0382] 'LINECHECK 0383 (next line)
[0383] Public Function ReplaceAuxiliaries(xStr As String) As String '1
[0384]     Dim Output As String
[0385]     Output = xStr
[0386]     Output = Replace(Output, " was ", " (was)AUX# ")
[0387]     Output = Replace(Output, " were ", " (were)AUX# ")
[0388]     Output = Replace(Output, " are ", " (are)AUX# ")
[0389]     Output = Replace(Output, " is ", " (is)AUX# ")
[0390]     Output = Replace(Output, " am ", " (am)AUX# ")
[0391]     Output = Replace(Output, " 'm ", " ('m)AUX# ")
[0392]     Output = Replace(Output, " 's ", " ('s)AUX# ")
[0393]     Output = Replace(Output, " 're ", " ('re)AUX# ")
[0394]
[0395]     ReplaceAuxiliaries = Output
[0396]
[0397] End Function
[0398]
[0399]
[0400]
[0401]
[0402]

```

```

[0403] 'LINECHECK 0404 (next line)
[0404] Function ReplaceProgressiveAutomatic(xStr As String, VerbListSheet As String, _
[0405]     NumberOfWordsBetween As Integer, _
[0406]     CurrentMode As Long, WordList As String) As String
[0407]
[0408]     Dim Output As String
[0409]     Dim WordArray() As String
[0410]     Dim j As Long
[0411]     Dim WS As Worksheet
[0412]     Dim xROW As Long
[0413]     Dim Minus1WordAux As Boolean
[0414]     Dim Minus2WordAux As Boolean
[0415]     WordList = "," + WordList + ","
[0416]     WordList = Replace(WordList, ",", ",, ", , vbBinaryCompare)
[0417]     WordList = Replace(WordList, ",,", ",, ", , vbBinaryCompare)
[0418]     xROW = 1
[0419]     Set WS = Worksheets(VerbListSheet)
[0420]
[0421]     Do While WS.Cells(xROW, 1) <> ""
[0422]         xROW = xROW + 1
[0423]     Loop
[0424]
[0425]     Output = xStr
[0426]     Output = LCase(Output)
[0427]
[0428]     WordArray = Split(Output, " ", , vbBinaryCompare)
[0429]     For j = LBound(WordArray) + NumberOfWordsBetween + 1 To UBound(WordArray)
[0430]
[0431]
[0432]         Minus2WordAux = False
[0433]         Minus1WordAux = (InStr(1, WordArray(j - 1), "aux#", vbBinaryCompare) > 0)
[0434]         If NumberOfWordsBetween = 1 Then Minus2WordAux = (InStr(1, WordArray(j - 2), "aux#", vbBinaryCompare) > 0)
[0435]         If Right(WordArray(j), 3) = "ing" And (Minus1WordAux Or Minus2WordAux) _
[0436]             And CurrentMode = EnumRegular Then
[0437]             WS.Cells(xROW, 1) = WordArray(j)
[0438]             WordArray(j) = WordArray(j) + "&"
[0439]             xROW = xROW + 1
[0440]         End If
[0441]
[0442]         'LINECHECK 0442
[0443]         If InStr(1, WordList, "," + WordArray(j) + ", ", vbBinaryCompare) > 0 _
[0444]             And (Minus1WordAux Or Minus2WordAux) And CurrentMode = EnumProgressiveVerbList Then
[0445]             WS.Cells(xROW, 1) = WordArray(j)
[0446]             WordArray(j) = WordArray(j) + "&"
[0447]             xROW = xROW + 1
[0448]         End If
[0449]
[0450]
[0451]
[0452]     Next j
[0453]     Output = Join(WordArray, " ")
[0454]     ReplaceProgressiveAutomatic = Output
[0455] End Function
[0456]
[0457]
[0458] Function xxxxxRemoveSpecialCharacters(xStr As String) As String
[0459]     Dim i As Long
[0460]     Dim NewStr As String
[0461]     Dim IsCapitalLetter As Boolean
[0462]     Dim IsLowerCaseLetter As Boolean
[0463]     Dim IsSpace As Boolean
[0464]     Dim IsApos As Boolean
[0465]     NewStr = xStr
[0466]     NewStr = LCase(NewStr)
[0467]     NewStr = Replace(NewStr, "in(g)", "ing", , , vbBinaryCompare)
[0468]
[0469]     For i = 1 To 127
[0470]         IsCapitalLetter = (i >= 65 And i <= 90)
[0471]         IsLowerCaseLetter = (i >= 97 And i <= 122)
[0472]         IsSpace = (i = 32)

```

```

[0473]     IsApos = (i = 39)
[0474]     If Not (IsCapitalLetter Or IsLowerCaseLetter Or IsSpace Or IsApos) Then _
[0475]         NewStr = Replace(NewStr, Chr(i), " ", , , vbBinaryCompare)
[0476]     Next i
[0477]     NewStr = Replace(NewStr, "", " ", , , vbBinaryCompare)
[0478]     For i = 1 To 100
[0479]         NewStr = Replace(NewStr, " ", " ", , , vbBinaryCompare)
[0480]     Next i
[0481]     xxxxxRemoveSpecialCharacters = LCase(NewStr)
[0482] End Function
[0483]
[0484] Function xxxxxMakeLowercase(xStr As String) As String
[0485]     xxxxxMakeLowercase = LCase(xStr)
[0486] End Function
[0487]
[0488] Function xxxxxStandardize(xStr As String) As String
[0489]
[0490]     Dim TempStr As String
[0491]     TempStr = xStr
[0492]     TempStr = Replace(TempStr, "chi|8;|", "chi|8;01.01|", , , vbBinaryCompare)
[0493]     TempStr = Replace(TempStr, "chi|6;|", "chi|6;01.01|", , , vbBinaryCompare)
[0494]     TempStr = Replace(TempStr, "chi|10;|", "chi|10;01.01|", , , vbBinaryCompare)
[0495]     xxxxxStandardize = TempStr
[0496]
[0497]
[0498] End Function
[0499]
[0500] Function xxxxxGetDate(xStr As String, SantaBarbara As Boolean) As String
[0501]     Dim TempStr As String
[0502]     Dim xDate As String
[0503]     Dim j As Long
[0504]     TempStr = xStr
[0505]
[0506]
[0507]     If SantaBarbara Then
[0508]         xxxxxGetDate = "12;01.01"
[0509]         Exit Function
[0510]     End If
[0511]
[0512]
[0513]     xDate = ""
[0514]     TempStr = LCase(TempStr)
[0515]     Dim WordArray() As String
[0516]
[0517]     WordArray = Split(TempStr, "|", , vbBinaryCompare)
[0518]     j = LBound(WordArray)
[0519]     Do
[0520]         If InStr(1, WordArray(j), ";", vbBinaryCompare) > 0 And _
[0521]             InStr(1, WordArray(j), ".", vbBinaryCompare) > 0 Then
[0522]             xDate = WordArray(j)
[0523]             j = UBound(WordArray) + 1
[0524]
[0525]         End If
[0526]         j = j + 1
[0527]     Loop Until j >= UBound(WordArray)
[0528]     xxxxxGetDate = xDate
[0529]
[0530] End Function
[0531]
[0532]
[0533] Function xxxxxGetName(xStr As String, SantaBarbara As Boolean) As String
[0534]
[0535]     Dim xName As String
[0536]
[0537]     If SantaBarbara Then
[0538]         xxxxxGetName = "SantaBarbara"
[0539]         Exit Function
[0540]     End If
[0541]
[0542]     xStr = LCase(xStr)

```

```

[0543] xName = "carterette"
[0544] If InStr(1, xStr, LCase("CHI Sarah Target"), vbBinaryCompare) Then xName = "Sarah"
[0545] If InStr(1, xStr, LCase("CHI Adam Target"), vbBinaryCompare) Then xName = "Adam"
[0546] If InStr(1, xStr, LCase("CHI Eve Target"), vbBinaryCompare) Then xName = "Eve"
[0547]     xxxxxGetName = xName
[0548]
[0549] End Function
[0550]
[0551]
[0552] Function xxxxxGetForChild(xStr As String) As String
[0553]     Dim MySheet As Worksheet
[0554]     Dim CurrentRow As Long
[0555]     Dim Lemma As String
[0556]     Dim Progressive As String
[0557]     Dim Output As String
[0558]     Dim WordArray() As String
[0559]     Dim j As Long
[0560]     Dim NewStr As String
[0561]     NewStr = ""
[0562]     Output = xStr
[0563]     Output = LCase(Output)
[0564]
[0565]     Output = Replace(Output, Chr(10) + Chr(9), " ", , , vbBinaryCompare)
[0566]     Output = Replace(Output, Chr(13) + Chr(9), " ", , , vbBinaryCompare)
[0567]
[0568]     Output = Replace(Output, Chr(9), "%", , , vbBinaryCompare)
[0569]     Output = Replace(Output, "*", "%", , , vbBinaryCompare)
[0570]
[0571]     WordArray = Split(Output, "%", , vbBinaryCompare)
[0572]     For j = LBound(WordArray) + 1 To UBound(WordArray)
[0573]         If WordArray(j) = "chi:" Then NewStr = NewStr + WordArray(j + 1)
[0574]
[0575]     Next j
[0576]     NewStr = Replace(NewStr, " ", " ", , , vbBinaryCompare)
[0577]     NewStr = Replace(NewStr, " ", " ", , , vbBinaryCompare)
[0578]     xxxxxGetForChild = NewStr
[0579]
[0580]
[0581] End Function
[0582]
[0583]
[0584]
[0585]
[0586]
[0587] 'LINECHECK 0588 (next line)
[0588] Public Function TextFile_FindReplace(FilePathInput As String, JustFileName As String, _
[0589]     FilePathOutput As String, StatSheet As String, VerbListSheet As String, _
[0590]     SubTotalsSheet As String, FileSuffix As String, SantaBarbara As Boolean, _
[0591]     NumberOfWordsBetween As Integer, CurrentMode As Long, WordList As String) As String
[0592] Dim TextFile As Integer
[0593] Dim i As Integer
[0594] Dim FileContent As String
[0595] Dim ChildDate As String
[0596] Dim ChildName As String
[0597] Dim WordCount As Long
[0598] Dim GroupNo As Integer
[0599]
[0600] TextFile = FreeFile
[0601] Open FilePathInput For Binary As TextFile
[0602] FileContent = Space(LOF(TextFile))
[0603] Get #TextFile, , FileContent
[0604] Close TextFile
[0605] FileContent = xxxxxMakeLowercase(FileContent)
[0606] FileContent = xxxxxStandardize(FileContent)
[0607] If SantaBarbara = True Then
[0608]     FileContent = TransformRawSpokenCorpusSantaBarbara(FileContent)
[0609] End If
[0610] ChildDate = xxxxxGetDate(FileContent, SantaBarbara)
[0611] ChildName = xxxxxGetName(FileContent, SantaBarbara)
[0612] If SantaBarbara = False Then FileContent = xxxxxGetForChild(FileContent)

```

```

[0613] WordCount = GetStringCount(FileContent, " ")
[0614] FileContent = xxxxxRemoveSpecialCharacters(FileContent)
[0615]
[0616] FileContent = ReplaceAuxiliaries(FileContent)
[0617] FileContent = ReplaceProgressiveAutomatic(FileContent, _
[0618]     VerbListSheet, NumberOfWordsBetween, CurrentMode, WordList)
[0619]
[0620]
[0621] TextFile = FreeFile
[0622]
[0623] FilePathOutput = Replace(FilePathOutput, ".txt", FileSuffix + ".txt", , , vbBinaryCompare)
[0624] Open FilePathOutput For Output As TextFile
[0625]
[0626] Print #TextFile, FileContent
[0627]
[0628] Close TextFile
[0629]
[0630]
[0631]
[0632] GroupNo = OutputResultsGetGroup(FileContent, ChildDate, ChildName, WordCount, StatSheet)
[0633] FileContent = FileContent + Chr(10) + Chr(13) + "Corpus " + ChildName _
[0634]     + ", filename:" + JustFileName + ", Group:" + Str(GroupNo) + Chr(10) + Chr(13) _
[0635]     + Chr(10) + Chr(13) + Chr(10) + Chr(13) + Chr(10) + Chr(13)
[0636] TextFile_FindReplace = FileContent
[0637] End Function
[0638]
[0639] Public Function OutputResultsGetGroup(TextBody As String, ChildDate As String, ChildName As String, _
[0640] WordCount As Long, StatSheet As String) As Integer
[0641]     Dim xStr As String
[0642]     Dim i As Long
[0643]     Dim j As Long
[0644]     Dim WordString As String
[0645]     Dim WStat As Worksheet
[0646]     Dim WStatRow As Long
[0647]     Dim ChildDateArray() As String
[0648]     Dim days As Integer
[0649]     Dim Counter As Long
[0650]     Dim GroupNo As Long
[0651]
[0652]     xStr = " " + TextBody + " "
[0653]
[0654]     Counter = GetStringCount(xStr, "&")
[0655]
[0656]     Set WStat = Worksheets(StatSheet)
[0657]     WStatRow = 1
[0658]     Do While WStat.Cells(WStatRow, 1) <> ""
[0659]         WStatRow = WStatRow + 1
[0660]     Loop
[0661]
[0662]     WStat.Cells(WStatRow, 1) = Counter
[0663]     WStat.Cells(WStatRow, 2) = ChildDate
[0664]     WStat.Cells(WStatRow, 3) = ChildName
[0665]     WStat.Cells(WStatRow, 4) = WordCount
[0666]
[0667]     ChildDate = Replace(ChildDate, " ", ";", , , vbBinaryCompare)
[0668]     ChildDateArray = Split(ChildDate, ";")
[0669]
[0670]     days = Val(ChildDateArray(0)) * 365 + Val(ChildDateArray(1)) * 12 + Val(ChildDateArray(2))
[0671]     WStat.Cells(WStatRow, 5) = days
[0672]     WStat.Cells(WStatRow, 6) = "=A" + Trim(Str(WStatRow)) + "*100000/D" + Trim(Str(WStatRow))
[0673]
[0674]     GroupNo = GetGroupNumberForDays(days)
[0675]     WStat.Cells(WStatRow, 8) = GroupNo
[0676]     If GroupNo = 1 Then WStat.Cells(WStatRow, 9) = Counter
[0677]     If GroupNo = 1 Then WStat.Cells(WStatRow, 10) = WordCount
[0678]     If GroupNo = 2 Then WStat.Cells(WStatRow, 11) = Counter
[0679]     If GroupNo = 2 Then WStat.Cells(WStatRow, 12) = WordCount
[0680]     If GroupNo = 3 Then WStat.Cells(WStatRow, 13) = Counter
[0681]     If GroupNo = 3 Then WStat.Cells(WStatRow, 14) = WordCount
[0682]     If GroupNo = 4 Then WStat.Cells(WStatRow, 15) = Counter

```

```

[0683] If GroupNo = 4 Then WStat.Cells(WStatRow, 16) = WordCount
[0684] If GroupNo = 5 Then WStat.Cells(WStatRow, 17) = Counter
[0685] If GroupNo = 5 Then WStat.Cells(WStatRow, 18) = WordCount
[0686]
[0687]
[0688] If GroupNo = 6 Then WStat.Cells(WStatRow, 19) = Counter
[0689] If GroupNo = 6 Then WStat.Cells(WStatRow, 20) = WordCount
[0690] If GroupNo = 7 Then WStat.Cells(WStatRow, 21) = Counter
[0691] If GroupNo = 7 Then WStat.Cells(WStatRow, 22) = WordCount
[0692] If GroupNo = 8 Then WStat.Cells(WStatRow, 23) = Counter
[0693] If GroupNo = 8 Then WStat.Cells(WStatRow, 24) = WordCount
[0694]
[0695]
[0696]
[0697] If GroupNo = 9 Then WStat.Cells(WStatRow, 25) = Counter
[0698] If GroupNo = 9 Then WStat.Cells(WStatRow, 26) = WordCount
[0699]
[0700]
[0701] OutputResultsGetGroup = GroupNo
[0702] End Function
[0703]
[0704]
[0705] Public Function GetGroupNumberForDays(days As Integer) As Integer
[0706] Dim GroupNo As Integer
[0707] If days < 615 Then GroupNo = 1
[0708] If days > 615 And days < 1000 Then GroupNo = 2
[0709] If days > 1000 And days < 1357 Then GroupNo = 3
[0710] If days > 1357 And days < 1736 Then GroupNo = 4
[0711] If days > 1736 Then GroupNo = 5
[0712]
[0713]
[0714] If days = 2203 Then GroupNo = 6
[0715] If days = 2933 Then GroupNo = 7
[0716] If days = 3663 Then GroupNo = 8
[0717]
[0718]
[0719]
[0720] If days = 4393 Then GroupNo = 9
[0721]
[0722]
[0723] GetGroupNumberForDays = GroupNo
[0724] End Function
[0725]
[0726]
[0727] Public Sub AddSecondSeriesForDividing5Groups(StatSheet As String)
[0728] Dim xStr As String
[0729] Dim i As Long
[0730] Dim j As Long
[0731] Dim WordString As String
[0732] Dim WStat As Worksheet
[0733] Dim WStatRow As Long
[0734] Dim ChildDateArray() As String
[0735] Dim days As Integer
[0736] Dim Counter As Long
[0737]
[0738] Set WStat = Worksheets(StatSheet)
[0739]
[0740] WStatRow = 1
[0741] Do While WStat.Cells(WStatRow, 1) <> ""
[0742] WStatRow = WStatRow + 1
[0743] Loop
[0744]
[0745]
[0746] For i = 0 To 9
[0747] WStat.Cells(WStatRow + i, 5) = 365 * (i + 1)
[0748] WStat.Cells(WStatRow + i, 7) = 300
[0749] WStat.Cells(WStatRow + i, 8) = i + 100
[0750] Next i
[0751]
[0752] WStat.Cells(WStatRow + 0, 9) = "=SUM(I2:I" + Trim(Str(WStatRow - 1)) + ")"

```



```

[0753] WStat.Cells(WStatRow + 0, 10) = "=SUM(J2:J" + Trim(Str(WStatRow - 1)) + ")"
[0754] WStat.Cells(WStatRow + 0, 11) = "=SUM(K2:K" + Trim(Str(WStatRow - 1)) + ")"
[0755] WStat.Cells(WStatRow + 0, 12) = "=SUM(L2:L" + Trim(Str(WStatRow - 1)) + ")"
[0756] WStat.Cells(WStatRow + 0, 13) = "=SUM(M2:M" + Trim(Str(WStatRow - 1)) + ")"
[0757] WStat.Cells(WStatRow + 0, 14) = "=SUM(N2:N" + Trim(Str(WStatRow - 1)) + ")"
[0758] WStat.Cells(WStatRow + 0, 15) = "=SUM(O2:O" + Trim(Str(WStatRow - 1)) + ")"
[0759] WStat.Cells(WStatRow + 0, 16) = "=SUM(P2:P" + Trim(Str(WStatRow - 1)) + ")"
[0760] WStat.Cells(WStatRow + 0, 17) = "=SUM(Q2:Q" + Trim(Str(WStatRow - 1)) + ")"
[0761] WStat.Cells(WStatRow + 0, 18) = "=SUM(R2:R" + Trim(Str(WStatRow - 1)) + ")"
[0762]
[0763] WStat.Cells(WStatRow + 1, 10) = "=I" + Trim(Str(WStatRow)) + "*"10000/J" + Trim(Str(WStatRow)) + ""
[0764] WStat.Cells(WStatRow + 1, 12) = "=K" + Trim(Str(WStatRow)) + "*"10000/L" + Trim(Str(WStatRow)) + ""
[0765] WStat.Cells(WStatRow + 1, 14) = "=M" + Trim(Str(WStatRow)) + "*"10000/N" + Trim(Str(WStatRow)) + ""
[0766] WStat.Cells(WStatRow + 1, 16) = "=O" + Trim(Str(WStatRow)) + "*"10000/P" + Trim(Str(WStatRow)) + ""
[0767] WStat.Cells(WStatRow + 1, 18) = "=Q" + Trim(Str(WStatRow)) + "*"10000/R" + Trim(Str(WStatRow)) + ""
[0768]
[0769]
[0770] WStat.Cells(WStatRow + 0, 19) = "=SUM(S2:S" + Trim(Str(WStatRow - 1)) + ")"
[0771] WStat.Cells(WStatRow + 0, 20) = "=SUM(T2:T" + Trim(Str(WStatRow - 1)) + ")"
[0772] WStat.Cells(WStatRow + 0, 21) = "=SUM(U2:U" + Trim(Str(WStatRow - 1)) + ")"
[0773] WStat.Cells(WStatRow + 0, 22) = "=SUM(V2:V" + Trim(Str(WStatRow - 1)) + ")"
[0774] WStat.Cells(WStatRow + 0, 23) = "=SUM(W2:W" + Trim(Str(WStatRow - 1)) + ")"
[0775] WStat.Cells(WStatRow + 0, 24) = "=SUM(X2:X" + Trim(Str(WStatRow - 1)) + ")"
[0776]
[0777] WStat.Cells(WStatRow + 1, 20) = "=S" + Trim(Str(WStatRow)) + "*"10000/T" + Trim(Str(WStatRow)) + ""
[0778] WStat.Cells(WStatRow + 1, 22) = "=U" + Trim(Str(WStatRow)) + "*"10000/V" + Trim(Str(WStatRow)) + ""
[0779] WStat.Cells(WStatRow + 1, 24) = "=W" + Trim(Str(WStatRow)) + "*"10000/X" + Trim(Str(WStatRow)) + ""
[0780]
[0781]
[0782]
[0783]
[0784] WStat.Cells(WStatRow + 0, 25) = "=SUM(Y2:Y" + Trim(Str(WStatRow - 1)) + ")"
[0785] WStat.Cells(WStatRow + 0, 26) = "=SUM(Z2:Z" + Trim(Str(WStatRow - 1)) + ")"
[0786]
[0787] WStat.Cells(WStatRow + 1, 26) = "=Y" + Trim(Str(WStatRow)) + "*"10000/Z" + Trim(Str(WStatRow)) + ""
[0788]
[0789]
[0790] End Sub
[0791]
[0792]
[0793] Sub CreateSubTotals(VerbListSheet As String, SubTotalsSheet As String)
[0794] Dim WSVerbList As Worksheet
[0795] Dim x
[0796] Set WSVerbList = ActiveWorkbook.Worksheets(VerbListSheet)
[0797]
[0798] WSVerbList.Select
[0799] WSVerbList.Sort.SortFields.Clear
[0800] WSVerbList.Sort.SortFields.Add _
[0801] Key:=Range("A1:A5000"), SortOn:=xlSortOnValues, Order:=xlAscending, _
[0802] DataOption:=xlSortNormal
[0803] With WSVerbList.Sort
[0804] .SetRange Range("A1:A5000")
[0805] .Header = xlGuess
[0806] .MatchCase = False
[0807] .Orientation = xlTopToBottom
[0808] .SortMethod = xlPinYin
[0809] .Apply
[0810] End With
[0811] MsgBox ("Just click OK in the following form.")
[0812] WSVerbList.Range("A1:A5000").Subtotal GroupBy:=1, Function:=xlCount, TotalList:=Array(1), _
[0813] Replace:=True, PageBreaks:=False, SummaryBelowData:=True
[0814] WSVerbList.Cells.ClearOutline
[0815] WSVerbList.Columns("A:B").Select
[0816] Selection.Copy
[0817] Sheets(SubTotalsSheet).Select
[0818] Columns("A:B").Select
[0819] Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
[0820] :=False, Transpose:=False
[0821] Application.CutCopyMode = False
[0822] ActiveWorkbook.Worksheets(SubTotalsSheet).Sort.SortFields.Clear

```

```

[0823] ActiveWorkbook.Worksheets(SubTotalsSheet).Sort.SortFields.Add Key:=Range _
[0824] ("B1:B5000"), SortOn:=xlSortOnValues, Order:=xlDescending, DataOption:= _
[0825] xlSortNormal
[0826] With ActiveWorkbook.Worksheets(SubTotalsSheet).Sort
[0827] .SetRange Range("A1:B5000")
[0828] .Header = xlGuess
[0829] .MatchCase = False
[0830] .Orientation = xlTopToBottom
[0831] .SortMethod = xlPinYin
[0832] .Apply
[0833] End With
[0834]
[0835] ActiveWorkbook.Worksheets(SubTotalsSheet).Sort.SortFields.Clear
[0836] ActiveWorkbook.Worksheets(SubTotalsSheet).Sort.SortFields.Add Key:=Range _
[0837] ("B1:B5000"), SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:= _
[0838] xlSortNormal
[0839] With ActiveWorkbook.Worksheets(SubTotalsSheet).Sort
[0840] .SetRange Range("A1:B5000")
[0841] .Header = xlGuess
[0842] .MatchCase = False
[0843] .Orientation = xlTopToBottom
[0844] .SortMethod = xlPinYin
[0845] .Apply
[0846] End With
[0847]
[0848] End Sub
[0849]
[0850] 'LINECHECK 0850
[0851] Private Sub CalculateProgressiveSubTaggedAlgorithm(StatSheet As String, FileSuffix As String, _
[0852] NumberOfWordsBetween As Integer, CurrentMode As Long, LemmaList As String)
[0853] Dim StrFile As String
[0854]
[0855] Dim WS As Worksheet
[0856] Dim WS2 As Worksheet
[0857] Dim MegaCorpus As String
[0858]
[0859] Dim TotalTaggedProgressiveString As String
[0860] TotalTaggedProgressiveString = ""
[0861] MegaCorpus = ""
[0862] Set WS2 = Worksheets(StatSheet)
[0863] Call WS2.Cells.Clear
[0864] WS2.Cells(1, 1) = "#Progressive Count"
[0865] WS2.Cells(1, 2) = "#Time Stamp"
[0866] WS2.Cells(1, 3) = "#Name"
[0867] WS2.Cells(1, 4) = "#SpaceCount"
[0868] WS2.Cells(1, 5) = "#Days"
[0869] WS2.Cells(1, 6) = "#Per 100000 Words"
[0870] WS2.Cells(1, 7) = "#Second serie"
[0871] WS2.Cells(1, 8) = "#Group No"
[0872] WS2.Cells(1, 9) = "#Group1 Freq"
[0873] WS2.Cells(1, 10) = "#Group1 SpaceCount"
[0874] WS2.Cells(1, 11) = "#Group2 Freq"
[0875] WS2.Cells(1, 12) = "#Group2 SpaceCount"
[0876] WS2.Cells(1, 13) = "#Group3 Freq"
[0877] WS2.Cells(1, 14) = "#Group3 SpaceCount"
[0878] WS2.Cells(1, 15) = "#Group4 Freq"
[0879] WS2.Cells(1, 16) = "#Group4 SpaceCount"
[0880] WS2.Cells(1, 17) = "#Group5 Freq"
[0881] WS2.Cells(1, 18) = "#Group5 SpaceCount"
[0882] WS2.Cells(1, 19) = "#Group6 Freq"
[0883] WS2.Cells(1, 20) = "#Group6 SpaceCount"
[0884] WS2.Cells(1, 21) = "#Group7 Freq"
[0885] WS2.Cells(1, 22) = "#Group7 SpaceCount"
[0886] WS2.Cells(1, 23) = "#Group8 Freq"
[0887] WS2.Cells(1, 24) = "#Group8 SpaceCount"
[0888] WS2.Cells(1, 25) = "#Group9 Freq"
[0889] WS2.Cells(1, 26) = "#Group9 SpaceCount"
[0890] StrFile = Dir(ThisWorkbook.Path & "\FileLibrary\*" & "*.cha")
[0891] Do While Len(StrFile) > 0
[0892] MegaCorpus = MegaCorpus + TextFile_FindReplace_TAGGED_ALGORITHM(ThisWorkbook.Path _

```

```

[0893]     + "\FileLibrary\" + StrFile, StrFile, ThisWorkbook.Path _
[0894]     + "\FileLibrary\" + Replace(StrFile, ".", "", , , vbBinaryCompare) _
[0895]     + "-Output.txt", StatSheet, FileSuffix, _
[0896]     TotalTaggedProgressiveString, NumberOfWordsBetween, CurrentMode, LemmaList)
[0897]     StrFile = Dir
[0898]     Loop
[0899]     Call AddSecondSeriesForDividing5Groups(StatSheet)
[0900]     Call WriteTextBodyToFile("0000MegaCorpus-" + FileSuffix + ".txt", MegaCorpus)
[0901]     Call WriteTextBodyToFile("0000MegaCorpus-" + FileSuffix + "-TAGGED" + ".txt", _
[0902]     TotalTaggedProgressiveString)
[0903]
[0904]     MsgBox ("Done!")
[0905] End Sub
[0906]
[0907]
[0908] 'LINECEHCK 0909 (next line)
[0909] Private Sub CalculateProgressiveSub(StatSheet As String, VerbListSheet As String, _
[0910]     SubTotalsSheet As String, FileSuffix As String, _
[0911]     NumberOfWordsBetween As Integer, CurrentMode As Long, WordList As String)
[0912] Dim StrFile As String
[0913] Dim WS As Worksheet
[0914] Dim WS2 As Worksheet
[0915] Dim MegaCorpus As String
[0916]
[0917] MegaCorpus = ""
[0918] Set WS = Worksheets(SubTotalsSheet)
[0919] Call WS.Cells.Clear
[0920] Set WS = Nothing
[0921]
[0922]
[0923] Set WS = Worksheets(VerbListSheet)
[0924] Call WS.Cells.Clear
[0925] WS.Cells(1, 1) = "#Progressive form"
[0926] Set WS = Nothing
[0927]
[0928] Set WS2 = Worksheets(StatSheet)
[0929] Call WS2.Cells.Clear
[0930] WS2.Cells(1, 1) = "#Progressive Count"
[0931] WS2.Cells(1, 2) = "#Time Stamp"
[0932] WS2.Cells(1, 3) = "#Name"
[0933] WS2.Cells(1, 4) = "#SpaceCount"
[0934] WS2.Cells(1, 5) = "#Days"
[0935] WS2.Cells(1, 6) = "#Per 100000 Words"
[0936] WS2.Cells(1, 7) = "#Second serie"
[0937] WS2.Cells(1, 8) = "#Group No"
[0938] WS2.Cells(1, 9) = "#Group1 Freq"
[0939] WS2.Cells(1, 10) = "#Group1 SpaceCount"
[0940] WS2.Cells(1, 11) = "#Group2 Freq"
[0941] WS2.Cells(1, 12) = "#Group2 SpaceCount"
[0942] WS2.Cells(1, 13) = "#Group3 Freq"
[0943] WS2.Cells(1, 14) = "#Group3 SpaceCount"
[0944] WS2.Cells(1, 15) = "#Group4 Freq"
[0945] WS2.Cells(1, 16) = "#Group4 SpaceCount"
[0946] WS2.Cells(1, 17) = "#Group5 Freq"
[0947] WS2.Cells(1, 18) = "#Group5 SpaceCount"
[0948] WS2.Cells(1, 19) = "#Group6 Freq"
[0949] WS2.Cells(1, 20) = "#Group6 SpaceCount"
[0950] WS2.Cells(1, 21) = "#Group7 Freq"
[0951] WS2.Cells(1, 22) = "#Group7 SpaceCount"
[0952] WS2.Cells(1, 23) = "#Group8 Freq"
[0953] WS2.Cells(1, 24) = "#Group8 SpaceCount"
[0954] WS2.Cells(1, 25) = "#Group9 Freq"
[0955] WS2.Cells(1, 26) = "#Group9 SpaceCount"
[0956] StrFile = Dir(ThisWorkbook.Path & "\FileLibrary\*" & "*.cha")
[0957] Do While Len(StrFile) > 0
[0958] MegaCorpus = MegaCorpus + TextFile_FindReplace(ThisWorkbook.Path + "\FileLibrary\" _
[0959] + StrFile, StrFile, ThisWorkbook.Path + "\FileLibrary\" + _
[0960] Replace(StrFile, ".", "", , , vbBinaryCompare) + "-Output.txt", _
[0961] StatSheet, VerbListSheet, SubTotalsSheet, FileSuffix, False, NumberOfWordsBetween, CurrentMode, WordList)
[0962] StrFile = Dir

```

```

[0963] Loop
[0964]
[0965] StrFile = Dir(ThisWorkbook.Path & "\FileLibrary*" & "*.trn")
[0966] Do While Len(StrFile) > 0
[0967] MegaCorpus = MegaCorpus + TextFile_FindReplace(ThisWorkbook.Path + _
[0968] "\FileLibrary\" + StrFile, StrFile, ThisWorkbook.Path + "\FileLibrary\" _
[0969] + Replace(StrFile, ".", "", , , vbBinaryCompare) + "-Output.txt", StatSheet, _
[0970] VerbListSheet, SubTotalsSheet, FileSuffix, _
[0971] True, NumberOfWordsBetween, CurrentMode, WordList)
[0972] StrFile = Dir
[0973]
[0974] Loop
[0975]
[0976] Call AddSecondSeriesForDividing5Groups(StatSheet)
[0977] Call CreateSubTotals(VerbListSheet, SubTotalsSheet)
[0978] Call WriteTextBodyToFile("0000MegaCorpus-" + FileSuffix + ".txt", MegaCorpus)
[0979] MsgBox ("Done!")
[0980] End Sub
[0981]
[0982]
[0983]
[0984]
[0985]
[0986]
[0987]
[0988]
[0989]
[0990] Private Sub TryAddSheet(Sheetname As String)
[0991] Dim W As Worksheet
[0992] Dim found As Boolean
[0993] found = False
[0994]
[0995] For Each W In ThisWorkbook.Worksheets
[0996] If LCase(W.Name) = LCase(Sheetname) Then found = True
[0997] Next
[0998] If found = True Then Exit Sub
[0999] Set W = Worksheets.Add
[1000] W.Name = Sheetname
[1001] End Sub
[1002] 'LINECHECK 1003 (next line)
[1003] Private Sub CalculateProgressive_Click()
[1004]
[1005] Dim CurrentMode As Long
[1006] Dim WordList As String
[1007] Dim SheetPrefix As String
[1008] Dim Sheet1 As String
[1009] Dim Sheet2 As String
[1010] Dim Sheet3 As String
[1011] Dim Extension As String
[1012] Dim W As Worksheet
[1013]
[1014] If Me.MaxWordsBetween.Value <> 0 And Me.MaxWordsBetween.Value <> 1 Then
[1015] MsgBox ("Illegal value for Maximal numbers of words between.")
[1016] Exit Sub
[1017] End If
[1018]
[1019]
[1020] SheetPrefix = ""
[1021] WordList = "," + Replace(LCase(Me.xListOfLemmasOrProgressives), " ", "", , , vbBinaryCompare) + ","
[1022] CurrentMode = -1
[1023]
[1024] If Me.xProgressiveList = True Then CurrentMode = EnumProgressiveVerbList
[1025] If Me.xLemmaList = True Then CurrentMode = EnumLemmaList
[1026] If Me.xProgressiveAuto Or Me.xProgressiveTagBased = True Then CurrentMode = EnumRegular
[1027] If CurrentMode = EnumLemmaList Then SheetPrefix = SheetPrefix + "LEM-" Else SheetPrefix = SheetPrefix + "PRG-"
[1028]
[1029]
[1030] If Me.xProgressiveAuto Then Extension = "PRG-AUTO"
[1031] If Me.xProgressiveTagBased Then Extension = "PRG-TAG"
[1032] If Me.xProgressiveList Then Extension = "PRG-LIST"

```

```
[1033] If Me.xLemmaList Then Extension = "LEM-LIST"
[1034]
[1035] If CurrentMode = EnumLemmaList Or CurrentMode = EnumProgressiveVerbList Then SheetPrefix = SheetPrefix + "LIST-"
[1036] If Me.xProgressiveAuto Then SheetPrefix = SheetPrefix + "AUTO-" Else SheetPrefix = SheetPrefix + "TAG-"
[1037] If MaxWordsBetween = 1 Then SheetPrefix = SheetPrefix + "OneBTW-"
[1038] Sheet1 = SheetPrefix + "1"
[1039] Sheet2 = SheetPrefix + "2"
[1040] Sheet3 = SheetPrefix + "3"
[1041]
[1042]
[1043] If Me.xProgressiveAuto Or Me.xProgressiveList Then
[1044]     Call TryAddSheet(Sheet3)
[1045]     Call TryAddSheet(Sheet2)
[1046] End If
[1047] Call TryAddSheet(Sheet1)
[1048] If CurrentMode = -1 Then
[1049]     MsgBox ("Error code 384372618")
[1050]     Exit Sub
[1051] End If
[1052]
[1053] If Me.xProgressiveAuto.Value = True Or Me.xProgressiveList = True Then _
[1054]     Call CalculateProgressiveSub(Sheet1, Sheet2, Sheet3, Extension, _
[1055]     Me.MaxWordsBetween.Value, CurrentMode, WordList) Else _
[1056]     Call CalculateProgressiveSubTaggedAlgorithm(Sheet1, Extension, Me.MaxWordsBetween.Value, CurrentMode, WordList)
[1057] End Sub
```