Constructions emerging : a usage-based model of the acquisition of grammar

Beekhuizen, B.F.

CHAPTER 7

Production experiments

Having seen the behavior of the model (chapter 5) and its inner workings (chapter 6), we now turn to the last topic: the production of language. Desideratum D2 holds that a computational model of language acquisition not only has to account for comprehension, but also for production. In this chapter, we look at the capacity of the model to produce utterances on the basis of a situation, as well as how its behavior develops over time.

7.1 Global development of production

7.1.1 Evaluation

How do we evaluate the accuracy of the produced utterances? Recall that the input generation procedure of Alishahi & Stevenson (2010) generates utterance-situation pairs. In the first production experiment, we generate a test set of 100 utterance-situation pairs at random. Importantly, we are interested in SPL's grammatical behavior, and giving it situations it has seen before would result in simple 'recall' of the analysis of an utterance paired with that situation. For that reason, the 100 utterance-situation pairs in the test set are held out from the input generation procedure for the input items in the simulation as reported in chapter 5.[1]

[1] This works as follows: SPL first generates 100 unique utterance-situation pairs. When generating novel input items for the simulation, it checks for every input item if it can be found in this set of test items. If it is found, a new input item is generated. This procedure is repeated until the new input item is no longer one of the test items.

After every 100 input items, we give the model the situations, but not the utterances of the test set, and ask it to generate the most likely utterance on the basis of the situation (as defined in section 3.7). The resulting utterance U_{gen} can then be compared with the utterance Uwhich was generated by the input generation procedure.

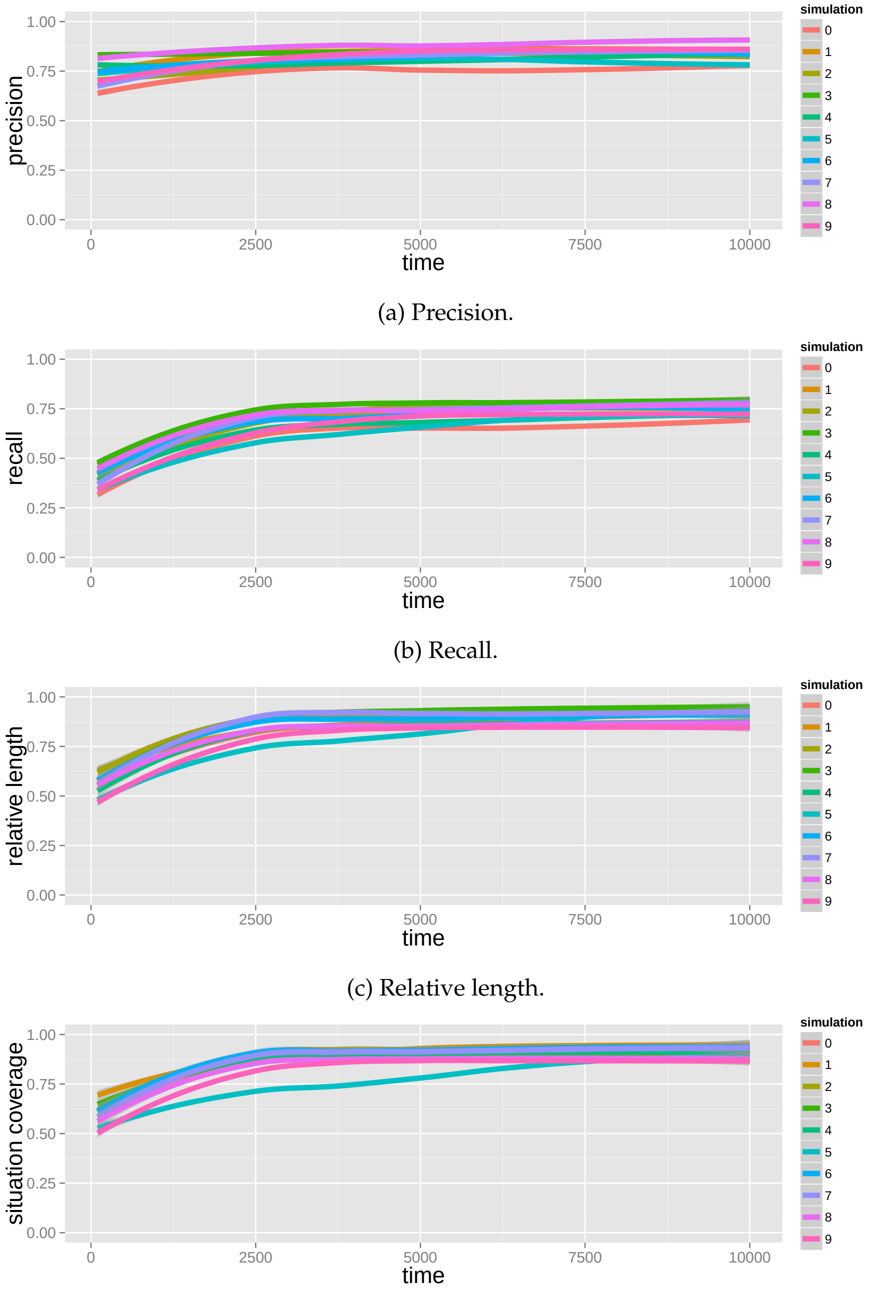
Two aspects of the comparison between U and U_{gen} are central to the evaluation. First, what proportion of U_{gen} is correct? That is: if we generate an utterance, does the model produce words that are part of U. If it produces different words, it has learned erroneous representations. Moreover, we want the model to produce the correct words *in the correct order*. When generating an utterance for the situation in which the father gets the ball, we do not want the model to generate *ball daddy get* or *ball get daddy*. To measure the proportion of words of U_{gen} being produced in the right order, we take the length of the maximal, potentially discontiguous substring shared between U and U_{gen} and divide it by the length of U_{gen}. We call this measure **precision** and errors on the precision correspond to errors of comission: SPL produces things that it should not produce. To give an example of the precision calculation: if the model produced *daddy give ball*, and U consists of the string *daddy give me ball*, the precision is $\frac{3}{3} = 1$ as all words in U_{gen} are found in U in the right order (but *me* is missing from U_{gen}). If the model, however, produced *give ball daddy*, the maximally shared discontiguous substring is *give ball*, and the **precision** is $\frac{2}{3} \approx 0.67$.

The complementary measure of evaluation is the **recall**. This measure captures what proportion of U is present in U_{gen}, again in the correct order. To calculate the recall, we again take the length of the maximal, potentially discontiguous substring shared between U and U_{gen}, but now divide it by the length of U rather than that of U_{gen}. **Recall** measures the amount of errors of omission: the score is penalized for words that are left out of U_{gen} but are present in U. For $U =$ *daddy give me ball* and $U_{\text{gen}} =$ *daddy give ball*, the maximal shared substring is *daddy give ball*, and the recall would be $\frac{3}{4} = 0.75$. For $U_{\text{gen}} =$ *give ball daddy*, the **recall** would be **recall** $= \frac{2}{4} = 0.5$.

Two other numbers are of interest. Besides **precision** and **recall**, it is insightful to see how long the productions in U_{gen} are, compared to the actual utterance U. This figure tells us whether produced utterances become longer over developmental time regardless of their correctness. **Relative length** is calculated by dividing the length of U_{gen} by the length of U. Finally, as with the comprehension experiment, we would like to know what parts of the situation the model expresses with its production. To this end, we calculate the **situation coverage** for the best analysis (see equation (5.2)).

7.1.2 Results

Figure 7.1 gives the values over time for the four measures. After $10,000$ input items, the **precision** scores for the ten simulations range between 0.75 and 0.9 (0.84 on average), whereas the **recall** scores at the end of the simulation range



(a) Precision.

(b) Recall.

(c) Relative length.

(d) Situation coverage.

Figure 7.1: Evaluation of production results.

between 0.7 and 0.8 (0.75 on average). Alhough far from perfect, the model does produce utterances that are relatively close to what an adult (i.c., the actual utterances from input generation procedure) would have said. I will analyze the errors the model makes in section 7.2.

Comparing **precision** and **recall**, it is remarkable to see how the **precision** starts out high, goes through a small dip in some simulations, and then goes up again, whereas **recall** starts low (0.2 to 0.35), and rises over the first 3000 input items to its final values (with the exception of simulation 5, the lowest line in **recall**, **relative length**, and **situation coverage**, to which we will return below). This observation is in line with the general observation that children's errors of commission are few, whereas they frequently make errors of omission.

Turning to the **relative length** now (figure 7.1c), we can see that the length of the produced utterances when compared to the actual utterances approaches its ceiling level after 4000 input items for most simulations (and some 6000 for simulation 5). The relative length at the end of the simulation is between 0.85 and 0.95, meaning that the utterances produced by the model are on average 0.85 to 0.95 times as long as the actual utterances.

Finally, the **situation coverage** of the model converges to an almost full expressivity relatively quickly, reaching values of around 0.90 and higher after some 2500 input items, again with simulation 5 lagging behind and reaching full expressivity after some 7000 input items.

Concluding: the model is relatively well able to produce utterances for novel situations, expressing the largest part of the situation. The **precision** and **recall** scores never reach, or even approach the full 1.00. We turn to the sources of this effect in the next section.

7.1.3 An example

Suppose you want to express a state of affairs in which an entity who can be categorized as a father enables the change of possession of a piece of gum. An adult speaker could say something like *father gives me gum* in such a case. Aften 900 input items, the model does so as well (example (52)), producing the utterance *father give me gum*. When we look at the best analysis leading to this utterance, we can see that SPL uses a maximally abstract ditransitive construction, combined with lexical constructions for every word.

The road to this production is one of a gradual build-up of the full utterance when looking at the utterances produced. As we can see in example (48) through (51), the model subsequently produces *give*, *me give*, and *father give me* before arriving at *father give me gum*. This is in line with the observation that over time more and more arguments of a verb are expressed (Tomasello 1992). When looking at the best analyses leading to these generated utterances, we find an interesting pattern. First, only a lexical construction leading to the word *give* is used, after which the model employs a maximally abstract intransitive construction to combine *me* with *give*. The intransitive construction only specifies that the first constituent fulfills a participant role in the event, and so the recipient *me* fits that slot. Combining this constellation with the lexical *give*-construction, the model arrives at a richer semantic interpretation: the role filled by *me* is now specified to be the RECIPIENT. It is interesting that the model makes a word-order error because of this: it takes the 'pre-verbal' slot to allow any semantic argument, and as such the model overextends a construction. Note that this kind of generation is allowed by the model in subsequent generation turns as well, but from $t = 300$ onwards, there are already analyses that are more likely and have a better coverage of the meaning. Overgeneralization does not go away, it is just outcompeted.

Looking at the generations at $t = 300$ and $t = 500$ (examples (50) and (51)), we can see that the model generates the string *father give me*, but does so with different means. In the former case, SPL uses a fully lexicalized construction, whereas in the latter, a verb-island construction [[PERSON] [GIVE / *give*] [ENTITY]] is used, combined with lexical constructions for *father* and *me*. This means that by 500 input items, the slightly more abstract construction has become reinforced to a greater extent than the fully lexicalized construction.

(48) [GIVE(GIVER,GIVEN,RECIPIENT) / *give*]

(49) [[PERSON]→[SPEAKER / *me*] [EVENT]→[GIVE(GIVER,GIVEN,RECIPIENT) / *give*] |
GIVE(GIVER,GIVEN,RECIPIENT(SPEAKER))

(50) [[FATHER / *father*] [GIVE / *give*] [SPEAKER / *me*]] |
GIVE(GIVER(FATHER),GIVEN,BENEFICIARY(SPEAKER))

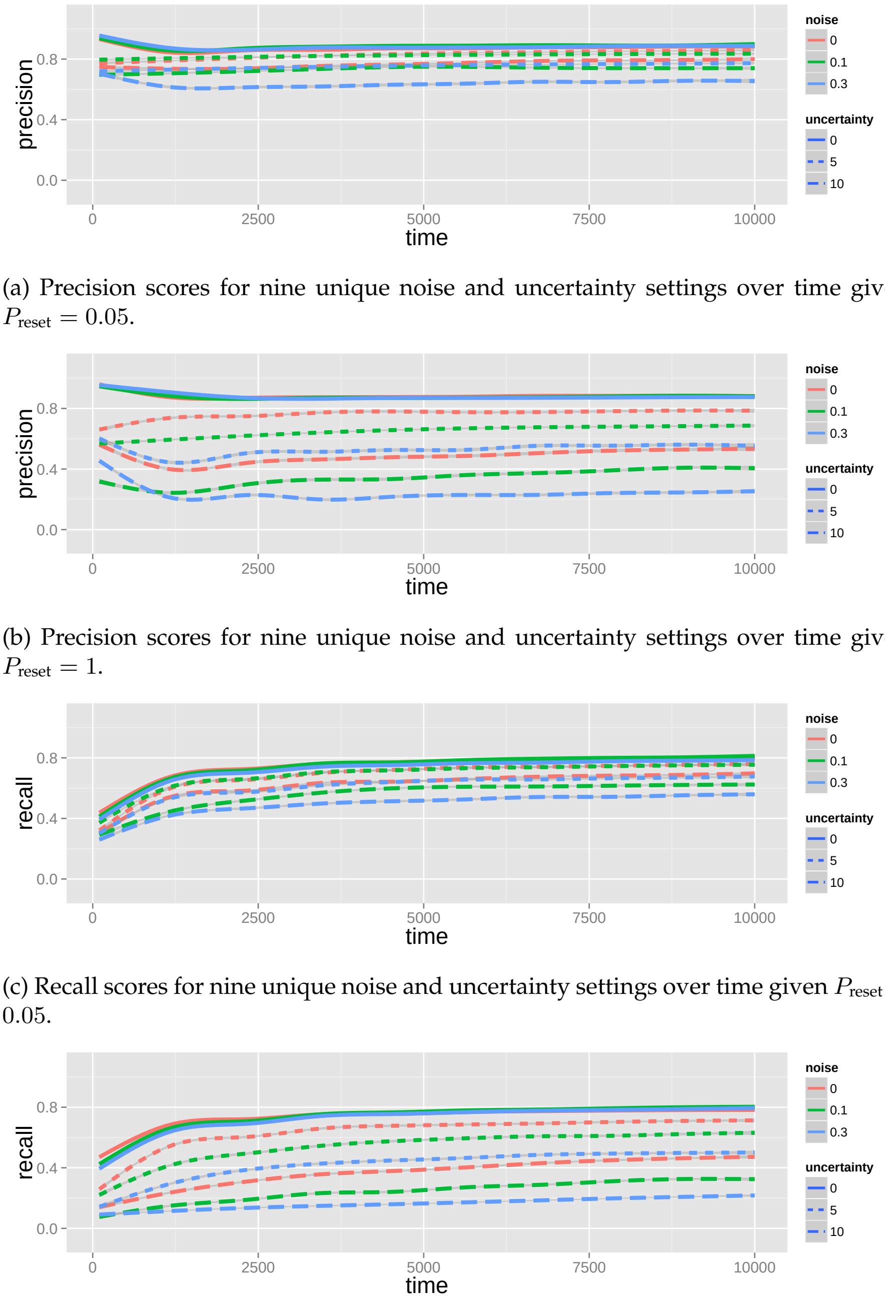
(51) [[PERSON]→[FATHER / *father*] [GIVE / *give*] [ENTITY]→[SPEAKER / *me*]] |
GIVE(GIVER(FATHER),AFFECTED-ROLE(SPEAKER))

(52) [[PERSON]→[FATHER / *father*] [CAUSE]→[GIVE / *give*] [OBJECT]→[SPEAKER / me] [ENTITY]→[GUM / *gum*]] |
GIVE(GIVER(FATHER),GIVEN(GUM),RECIPIENT(SPEAKER)

7.1.4 Robustness to uncertainty and noise

As in section 5.2.4, we can look at the model's performance given various settings for P_{noise}, $uncertainty$ and P_{reset}. If we make the conditions harder, does the model perform much worse on the generation task, or does its performance degrade gracefully? Again, we take values $noise = \{0.0, 0.1, 0.3\}$, $uncertainty = \{0, 5, 10\}$, $P_{\text{reset}} = \{0.05, 1\}$, and we run three simulations for every setting.

Looking at **precision** first, we can see that with $P_{\text{reset}} = 0.05$, increasing the levels of noise and uncertainty does not have a strong effect on the model's performance (figure 7.2a). Under the hardest condition, $uncertainty = 10$, $noise = 0.3$, the **precision** score after $10,000$ input items is 0.68, meaning that more than two thirds of the words the model produces are still correct. For



(a) Precision scores for nine unique noise and uncertainty settings over time given $P_{\text{reset}} = 0.05$.

(b) Precision scores for nine unique noise and uncertainty settings over time given $P_{\text{reset}} = 1$.

(c) Recall scores for nine unique noise and uncertainty settings over time given $P_{\text{reset}} = 0.05$.

(d) Recall scores for nine unique noise and uncertainty settings over time given $P_{\text{reset}} = 1$.

Figure 7.2: Precision and recall scores given various parameter settings.

all other settings, the PRECISION scores range between 0.75 and 0.85. That is: the model reasonably picks up the right representations from the noisy and uncertain sets of situations.

Setting P_{reset} to 1 makes SPL less robust to uncertainty and noise, as we have seen for the **identification** scores in section 5.2.4. Under the hardest conditions, the productions of the model are now only correct for some 20%, meaning that SPL has acquired many erroneous representations that, moreover, have been reinforced over time (figure 7.2b).

The variation between the various P_{noise} and $uncertainty$ settings is somewhat greater for the recall, meaning that, despite primarily producing utterances that are correct, they become less complete if the model faces higher levels of noise and uncertainty (figure 7.2c). The latter parameters seems to have a stronger effect than the former here: the lowest two scores after $10,000$ input items are for the setting $uncertainty = 10$. Again, the effect of setting the P_{reset} to 1 is dramatic (figure 7.2d): SPL acquires many erroneous representations, especially in the situation sets with high uncertainty, and subsequently fails to produce the correct target utterances.

Summarizing these findings, we could say that SPL is a robust learner given relatively high levels of noise and uncertainty (at least: higher levels than reported in other modeling experiments), but the chain of situations has to be 'coherent': if situations do not resemble each other, the robustness of the model fades away. However, I believe the uncertainty faced by actual learners is rather like the one given $P_{\text{reset}} = 0.05$ than $P_{\text{reset}} = 1$, as I argued in chapter 4. Asking the model to perform well given $P_{\text{reset}} = 1$ presents the experiential world of the child as an incoherent, haphazard sequence of events which we know it is not.

7.2 Error analysis

More interesting than the cases that are learned correctly are the ones where the model fails. Studying them provides us with more insight in the aspects of the model that cause this behavior, and thus constitute stepping stones towards even more comprehensive models. When the model omits words that are part of the actual utterance U or when it adds words that are not part of U, what are the kinds of errors the model makes? Some errors are more interesting than others: if the model simply has not acquired a lexical construction yet, and is hence unable to produce a certain word, it is simply a matter of time before the model encounters the word and (hopefully) acquires it. If we find errors in the grammatical patterns, for instance in the omission of arguments or displaying a different order, there is a more interesting story to be told. We will have a look at several cases in this section.

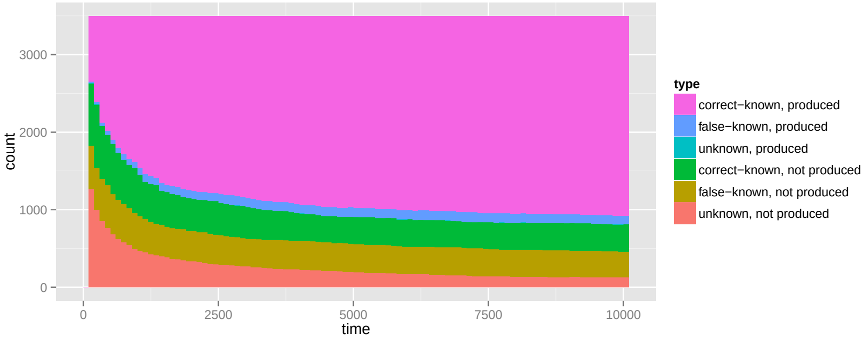


Figure 7.3: Lexical production over time, summed over 10 simulations.

7.2.1 Lexical errors

When a word in U is not produced in the generated utterance U_{gen}, there are several possibilities. First of all, SPL may simply not know the word, in which case it will either use another word or not express the meaning. Second, it may also be that the model has acquired the word. In that case, the acquisition may be correct (the word is learned with the right meaning) or incorrect (the word is learned with the wrong meaning). In the case of unknown words, and incorrectly acquired words, the story is relatively simple: SPL does not have the adequate representation, and hence does not produce the correct word. The case of correctly known, but not produced words is more interesting. Why would SPL not produce known and correct words when they are called for?

We can divide up the words of the various actual target utterances (the Us) in several groups: there are words that are produced, and words that are not produced (i.e., words that are or are not in U_{gen}. Both produced and non-produced words can be known as a word or not known as a word (i.e., at time t, there is a construction in Γ^t that has exactly one constituent with that word form as its phonological constraint). The known words can be further subdivided into correctly learned words and incorrectly learned ones (according to the input generation procedure). We count a word as correctly learned if there is at least one construction in Γ^t that has the meaning assigned to it in the input generation procedure as its meaning.

The counts of the six groups over time are given in figure 7.3. After $10,000$ input items, about 5 out of 7 words in all Us are correctly learned and produced in the generated utterance U_{gen}s. Initially, many words are simply not known and hence not produced, but the count of this group drops rapidly (recall that most word types have been seen after 1500 input items, cf. figure 5.7). Several words are simply acquired with the wrong meaning, and are therefore mostly not produced.

Outcompeted words

An interesting group are the cases in which the meaning has been acquired correctly, but that are still not produced (the green bin in figure 7.3). There are several reasons why cases like these exist. In the generation in example (53) below, the model tries to express the event in which a boy plays with a pen. It involves a semi-open construction involving the chunk *play with* and two open constituents for the participants. The two participant roles, however, are both filled with the word *worse* instead of *boy* and *pen*.

(53) U_{gen}: *worse play with worse*
U: *boy play with pen*

The expression of BOY with *worse* is easily explained: there are no lexical constructions involving *boy*. There are, however, several (erroneous) lexical constructions involving *worse* as the phonological specification. The abstraction over these, (i.e., the lowest common denominator) is the maximally abstract semantic feature ENTITY. The model now faces two choices: either not expressing BOY at all, or expressing it with the highly abstract [ENTITY / *worse*] construction. Because the model has acquired many grammatical constructions with the agent-role expressed as the first constituent and few without it, it will prefer the generation in which it can use a 'transitive'-like pattern combined with *worse* over a verb-patient construction without any word.

Roughly the same happens for the patient role. SPL has, at this point, acquired a [PEN / *pen*] construction, with a count of 1. Why does the model not combine this construction with the third constituent of the grammatical construction? The reason here is that *worse* has also been bootstrapped once (and erroneously) as meaning PEN. The count, however, is 0. There are, nonetheless, many lexical constructions with *worse* as their phonological form, and a meaning like ENTITY or ARTEFACT. These abstractions, as well as the [PEN / *worse*] 'gang up' (being equivalent derivations) and outweigh the [PEN / *pen*] construction.

This type of error can be considered to be a flaw in the design of the model, but resolving it on principled grounds is harder, and as such poses more of a theoretical challenge than an implementational issue. The problem is in the abstraction over lexical constructions: if a word is erroneously acquired and reinforced, and correctly learned and reinforced (e.g., [FATHER / *father*] and [PEN / *father*]), the lowest common denominator between the two is abstracted (e.g., [OBJECT / *father*]). We know this is unrealistic, but constraining the paradigmatization learning operation to apply in a more limited way would have to apply across the board. This is what, for instance Chang (2008) does in her model: the two constructions over which an abstraction is made, have to be sufficiently similar according to some metric. It is likely that this would work, but to what extent can it be justified as a cognitive operation? If abstraction is immanent, any shared structure is – in principle – an immanent abstraction. Restricting the amount of abstraction seems to me to impose an unprincipled constraint on immanence. If, however, such restrictions can be motivated, there is nothing barring us from implementing such a feature in a model.

Grammatical restrictions

The second case is constituted by words that are correctly learned, but not produced because there is no grammatical construction facilitating them or because the grammatical construction is less likely than another grammatical construction that does not facilitate that word.

In the former case, there simply is no grammatical construction to accomodate the production of the word. We can see an example of that in the best analysis of a situation in which Sarah puts a finger in her mouth (U = *Sarah put finger in mouth*), represented in (54) below.

(54) [[PERSON]→[SARAH / *sarah*] [EVENT]→[PUT / *put*] [OBJECT]→ [MOUTH / *mouth*]]

What happens in this case is that the best grammatical construction, the one that captures most of the situation and is most likely, is a transitive, and *Sarah* and *mouth* are expressed as the two arguments of that transitive. Nonetheless, at this point, the model does have two lexical constructions [IN / *in*] and [FINGER / *finger*], but it does not have the means to produce them under a single grammatical constellation.

These cases are interesting, because they are in line with the claim that errors of omission in early stages of language production do not depend on the vocabulary size, but that it is really a matter of grammar (Berk & Lillo-Martin 2012). Although Berk & Lillo-Martin (2012) argue for a different conception of grammar, their point can be easily transferred to a constructivist framework: all lexical constructions for producing a caused-motion pattern are present, it is just the caused-motion construction that is missing. This kind of analysis also provides a hint at a constructivist solution to Berk & Lillo-Martin's (2012) puzzle: if one-and-a-half-year-olds and six-year-olds that otherwise developed normally, go through the same phase of argument omission, the reason must be a grammatical one. A usage-based explanation of this phenomenon that, crucially, involves syntagmatization would be that the more abstact and longer grammatical patterns have not been 'constructed' yet.

The second case, where the grammatical pattern is available, but outcompeted, happens for an item in simulation 9 where the target utterance is *she play with toy* and the target situation PLAY(PLAYER(FEMALE-PERSON),TOOL-ROLE(TOY)). In the interval between 700 and 1400 input items, the model produces *she play with toy*, correctly, as the generated utterance, and does so on the basis of the analysis in example (55). This analysis involves a highly abstract transitive construction being combined with the chunk *play with* and the two participants. However, after 1400 input items, the model erroneously learns that *with* refers to the entity filling the TOOL-ROLE of the PLAY event, and acquires a construction given in example (56), in which the word *with* is taken to refer to the TOY. On the basis of an analysis combining this construction with the lexical construction [FEMALE-PERSON / *she*], the model produces the incomplete utterance *she play with*. This case is illustrative of a lexical error that is made despite the word being known: the model considers another construction 'better' for these purposes, despite even having a construction to express more aspects of the meaning.

(55) [[PERSON]→[FEMALE-PERSON / *she*] [EVENT]→[PLAY(PLAYER,TOOL-ROLE) / *play with*] [OBJECT]→[TOY / *toy*]]

(56) [[PERSON]→[FEMALE-PERSON / *she*] [PLAY / *play*] [OBJECT / *with*]]

7.2.2 Argument structure errors

Argument structure errors come in various sorts in the generations of the model. A first one is the case of a caused-motion event with a causer, and a object undergoing a falling action. The target utterance for such a sentence would be an intransitive utterance involving the undergoing object and the word *fall*, for instance *ball fall*. However, the meaning does steer towards a transitive expression. Note that the model does not have any alternative expressions available for expressing the causation of a falling event (e.g., the suppletive verb *drop* in *I dropped the ball* or a periphrastic causative like *I made the ball fall*. What happens in the model is that, after producing the sole word *fall* for a number of test moments, the model starts producing *fall* in the transitive frame, basically combining a maximally open transitive construction with the words for the causer and the undergoing object, and *fall*. This could be seen as a case of overgeneralization: the model wants to be expressive, but has no better means to do so than to use a transitive. However, the model never 'recovers' from this overgeneralization, as it has, as I mentioned, no alternative ways of expressing it and it has the built-in desire to trade off maximal expressivity with likelihood of the constructions.

The same pattern is found with caused motion events that involve, in the actual utterances, verbs like *go* and *come*, but are produced in a transitive frame (*you go it* for 'you made it move'). Here, again, there is no competing construction and the model relies on a highly general transitive construction despite never having heard *go* or *come* used in this frame. Here, however, it seems that the model does have a competing construction, viz. the caused-motion construction. However, both situations with *come* and *go* have a semantic feature COME and GO associated with them that clashes with the feature PUT associated with *put*, and hence the model is not able to use *put*. We will return to overgeneralizations in section 7.3.

7.2.3 Argument omission

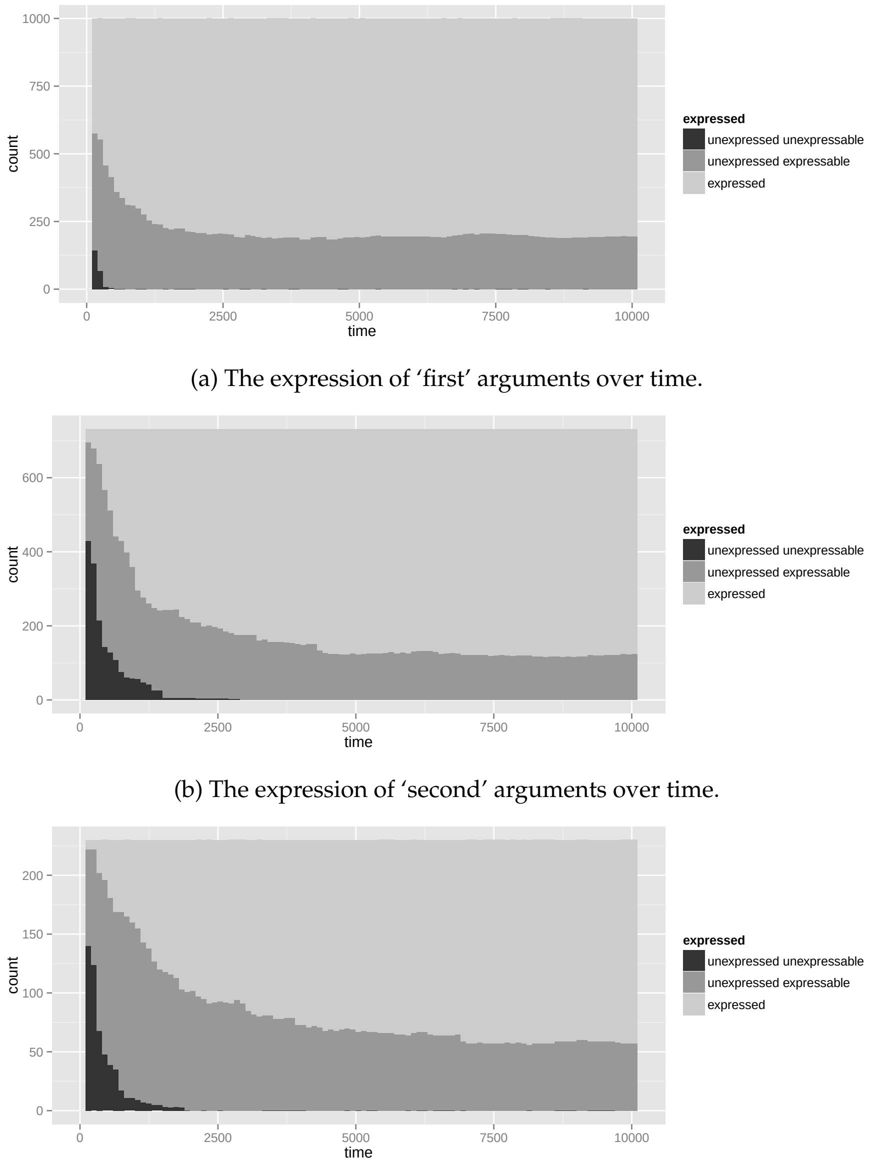
Recall that two of the explananda for a usage-based theory, E1 and E2, held that a computational model of language acquisition has to account for the increasing length of utterances, as well as explain why subject omission is more prevalent than the omission of other arguments. The data in figure 7.4 already suggests that the first explanandum is met: utterances become longer over time. The question, however, is whether this is actually an effect of more arguments being expressed or whether it is done for some other reason.

The three graphs in figure 7.4 show, over time, how often certain arguments are expressed. I grouped the arguments into three bins: 'first' arguments, such as agents and intransitive subjects, 'second' arguments, which are always undergoers, and 'third' arguments, encompassing recipients and locations. What we find, first, is that, in line with explanandum E1, more arguments are expressed over time.

This is not simply a factor of the growing vocabulary, as one may argue. The red bars in figure 7.4 display the 'unexpressed unexpressables', i.e., those meanings for which there is no construction in the grammar at that moment expressing them, whereas the green bars represent the 'unexpressed expressables' (i.e., those meanings that can be, but are not expressed). The former case is 'excusable': SPL simply has no means of expressing that concept. The latter group, the unexpressed expressables, is more interesting: here, SPL has a means of expressing that meaning, but cannot do so, because the grammatical constructions do not allow for it. As we can see for all three groups of arguments, the number of unexpressed unexpressables diminishes rapidly, whereas the number of unexpressed expressables diminishes more gradually. A main factor, according to this analysis, in early argument omission, is the availability of grammatical constructions for expressing arguments, in line with the findings of Berk & Lillo-Martin (2012), who excluded vocabulary size as a factor for the two-word phase (as discussed in chapter 2).

Turning to explanandum E2, the prevalence of subject omission, we can see that the model fares less well. For the first few hundreds of iterations, almost all second and third arguments are omitted, but only about half of the first arguments (i.e., subjects). One explanation for this could be that the model has no notion of information structure. As I discussed in chapter 2, Graf et al. (2015) found that children are more likely to omit old information. As subjects typically contain old information (Du Bois 1987), it is more likely that they are omitted. However, this explanation does not say how this is done representationally: are the subject arguments present in the grammatical representation and omitted, or are they simply not part of the linguistic construct? This is an issue that has been discussed extensively in various generative approaches (see chapter 2), but for which there is no clear answer yet within the usage-based framework.

A second explanation would be that learners have a right-edge bias in processing, in line with, for instance the MOSAIC model. If this is the case, it is



(a) The expression of 'first' arguments over time.

(b) The expression of 'second' arguments over time.

(c) The expression of 'third' arguments over time.

Figure 7.4: The expression of arguments over time, summed over 10 simulations.

likely that the model will start picking up [[EVENT] [ENTITY]] patterns earlier than [[ENTITY] [EVENT]] patterns, and that, hence, first arguments will be omitted more frequently. Similarly, one could imagine adding information structure to the comprehension: the more an argument is expected, the less salient it is, the less likely it is to be incorporated in the grammatical analysis, and hence the less likely it is to syntagmatize patterns involving the expected argument.

7.3 Overgeneralization

7.3.1 Motivation and Experimental set-up

In the previous section, we have seen that the model overgeneralizes the transitive construction to the verb *fall*, and does not overcome this overgeneralization. The reason it does not learn that *fall* is not to be used in a transitive frame, as adult speakers of English know, is that it has no alternative that prevents (or: pre-empts) this production. The existence of alternatives opens up the question under what conditions pre-emption takes place. The studies on overgeneralization by Ambridge and colleagues, as discussed in 2.4.3 present several factors involved in this process.

Statistical pre-emption, first, takes place when a competing form to the overgeneralization has been frequently encountered. Second, children seem to understand that if a verb is more frequently seen in a fixed set of constructions, their expectation of the occurrence of that verb in other argument-structure constructions becomes lower (entrenchment). Third, children are increasingly sensitive to the narrow verb classes for the various constructions: verbs of sound emission cannot be transitivized without a periphrastic causative (*I made him scream* vs. **I screamed him*) whereas verbs of manner of motion can be transitivized both with and without a periphrastic causative (*I rolled it* and *I made it roll*). Finally, Ambridge and colleagues suggest that the frequency of the various argument-structure constructions involved may have an effect as well: the more frequently an argument-structure construction occurs, irrespective of its relative frequency to the competing construction, the more entrenched it will be, and hence the more accessible.

All of these effects seem to follow from Alishahi & Stevenson's (2008) model. Can we, similarly, find them in the parsing approach taken with SPL? To investigate this, we adapt the input generation procedure slightly. The verb *fall* is part of the input generation procedure. It is produced either with a moving object as the first argument, in which case the situational event meaning is {EVENT,MOVE,FALL} and the underlying construction is [[ENTITY] [FALL / *fall*]] | FALL(MOVER(ENTITY)). The second construction in which *fall* has a moved object as the first argument, in which case the event meaning in the situation is {EVENT,CAUSE,MOVE,FALL} and the construction underlying it is [[ENTITY] [CAUSE-FALL / *fall*]] | CAUSE-FALL(MOVED(ENTITY)).

Recall that SPL overgeneralizes the transitive construction to generate cases like *you fall it* for the last type. Recall furthermore that the model does not overcome this overgeneralization for a lack of an alternative. For this experiment, I added another verb, *drop*, which has the same meaning as the second type of *fall* (viz. {EVENT,CAUSE,MOVE,FALL}), but also occurs in the transitive construction (i.e., [[ENTITY$_i$] [CAUSE-FALL / *drop*] [ENTITY$_j$]] | CAUSE-FALL(CAUSER(ENTITY$_i$),MOVER(ENTITY$_j$))). Will this alternative pre-empt the use of *fall* in the transitive construction?

Using this additional verb, we can manipulate the frequencies of the two verbs and the constructions they occur in to see if effects of entrenchment and pre-emption are found. The three frequencies we manipulate are:

1. The frequency of *fall* in the non-causative meaning. We expect that the higher the frequency of *fall* in this construction is, the more it will be entrenched, and the less likely it is that it will be extended to other argument frames. Within SPL, this expectation arises through the effect of independence, as discussed in chapter 6: the more a word will be seen in a particular construction, the more it will be associated with that construction, and the less autonomous it will be. We set the frequencies of *fall* in the non-causative frame to 750 (its original frequency) or 75.

2. The frequency of *fall* given a causative meaning. We expect that the higher the frequency of *fall* given this meaning, the more entrenched it is in the intransitive construction (but with a causative meaning), and the less frequent the overgeneralization will be.

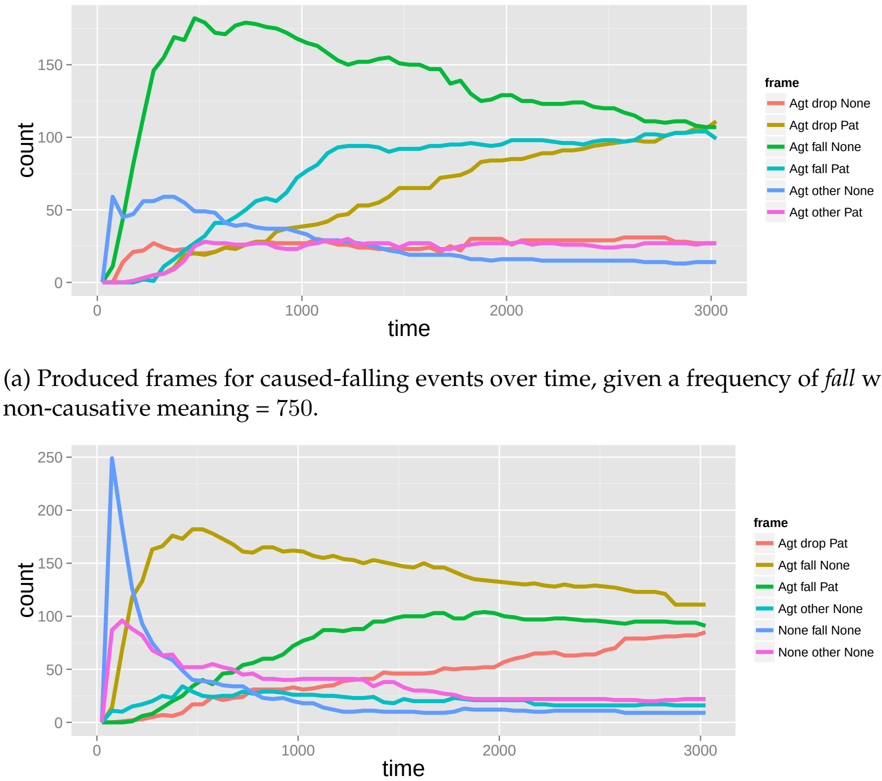
3. The frequency of *drop*. If *drop* is rare, its reinforcement will be weaker, and the chance of overgeneralizations will be higher. We set the frequencies of *drop* to 10 or 100.

I test these hypotheses by running 10 simulations of 3,000 input items for each of the 8 unique combinations of frequency settings. Every 50 input items, the model will receive 10 frames with a CAUSE-FALL event and two participants and is asked to generate utterances for each of them. I scored the produced generations as follows: the CAUSER-role can be expressed (Agt) or left unexpressed (None). The CAUSE-FALL event can be expressed with *drop*, *fall*, or another word, or left unexpressed. The MOVER-role, finally, can be expressed (Pat) or left unexpressed (None).

7.3.2 Results

Frequency of non-causative *fall*

Figure 7.5 displays the various types of generations for a CAUSE-FALL situation with two participants. For both frequency settings of non-causative *fall*, we can see that the majority of generations involves a causer and a



(a) Produced frames for caused-falling events over time, given a frequency of *fall* with non-causative meaning = 750.

(b) Produced frames for caused-falling events over time, given a frequency of *fall* with non-causative meaning = 1500.

Figure 7.5: Produced frames for caused-falling events over time with the frequency of *fall* with non-causative meaning as a dependent variable.

word expressing the event, represented as 'Agt fall' (e.g., *You fall* for CAUSE-FALL(CAUSER(HEARER),MOVED(BALL))). Over time, however, both the transitive use of *fall* ('Agt fall Pat', e.g., *you fall ball*) and the transitive use of *drop* ('Agt drop Pat', e.g., *you drop ball*) are on the rise.

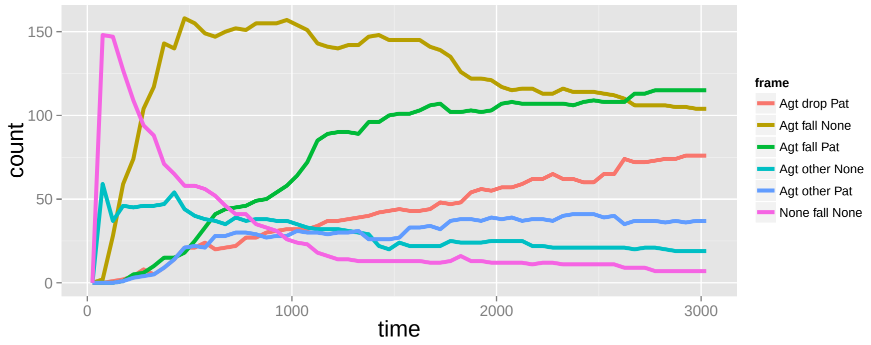
The difference between the two settings is that with the frequency of non-causative *fall* set to 750, the use of transitive *drop* surpasses that of both agentive-intransitive *fall* and transitive *fall* around 3,000 input items, whereas it remains lower than these two erroneous production types if we set the frequency of non-causative *fall* to 1500. This means that we do not find an entrenchment effect of *fall*: given the pure entrenchment hypothesis, we would expect that the more *fall* is seen in one grammatical construction, the less likely it would be to use it in other grammatical construction. Of course, this is an effect of the fact that SPL only positively reinforces verb-construction associations (with most-concrete constructions), but does not inhibit the non-occurrence of non-observed grammatical constructions.

Frequency of causative *fall*

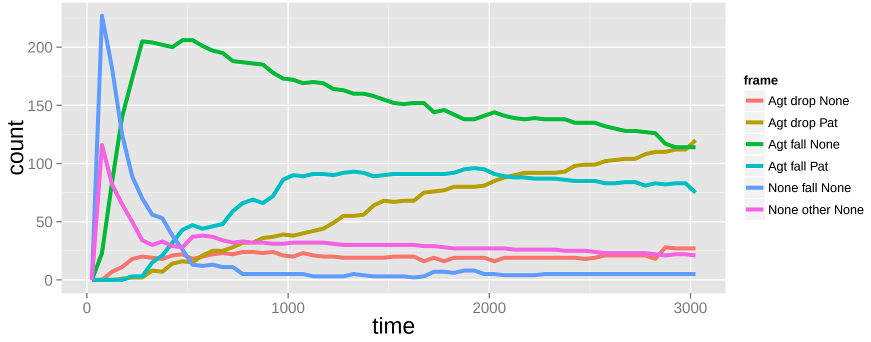
Interestingly, for the frequency of *fall* in the causative, but intransitive, frame, we do see an entrenchment effect (figure 7.6). Again, we find 'Agt drop None' being used most frequently early on, with 'Agt fall Pat' and 'Agt drop Pat' rising in frequency over time. However, here the higher frequency of *fall* given a causative meaning makes the correct use of *drop* being acquired faster, with its use surpassing that of 'Agt fall' and 'Agt fall Pat' at aroun 3000 input items (figure 7.6b). This means that we do find an entrenchment effect here: the more the model has seen *fall* with a causative meaning in the intransitive construction only, the quicker it arrives at productions with *drop* as a suppletive verb. SPL behaves like this because the representation of the constructions underlying the intransitive-*fall* utterances with a causative meaning are more reinforced, thus allowing the model to produce 'Pat fall' constructions. These constructions are, however, never produced, because the model finds the 'Agt fall Pat' and 'Agt drop Pat' patterns more expressive, and the 'Agt fall' pattern better entrenched and hence more likely.

Frequency of *drop*

The frequency setting for *drop* has the greatest effect. If we set the frequency of *drop* to 10, as in figure 7.7a, the verb is simply not reinforced enough to compete with *fall*, which has a frequency summed over both frames it occurs in of 775. Setting the frequency of *drop* to 100 remedies this and makes *drop* a viable competitor to the use of *fall*: the use of *drop* in a transitive construction surpasses both the 'Agt fall' and 'Agt fall Pat' patterns around 1800 input items, despite *drop* still being around 8 times as infrequent as *fall*.

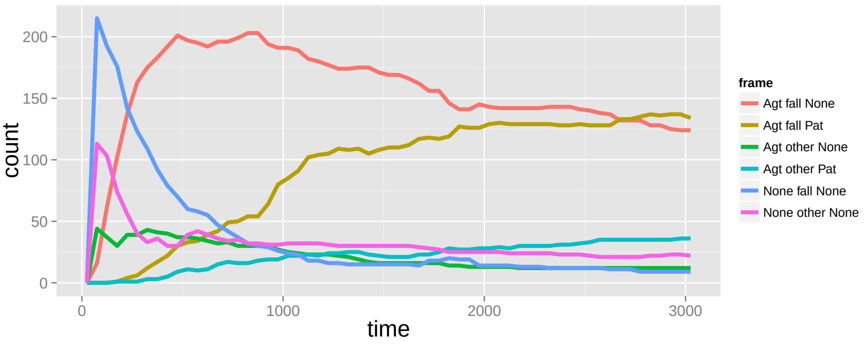


(a) Produced frames for caused-falling events over time, given a frequency of *fall* with causative meaning = 25.

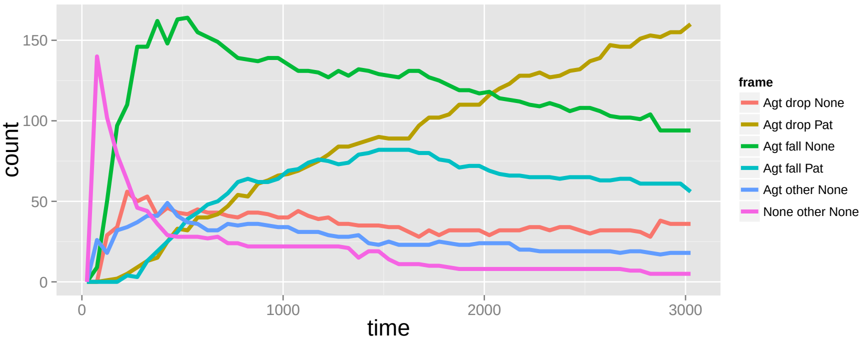


(b) Produced frames for caused-falling events over time, given a frequency of *fall* with non-causative meaning = 250.

Figure 7.6: Produced frames for caused-falling events over time with the frequency of *fall* with causative meaning as a dependent variable.



(a) Produced frames for caused-falling events over time, given a frequency of *drop* = 10.



(b) Produced frames for caused-falling events over time, given a frequency of *drop* = 100.

Figure 7.7: Produced frames for caused-falling events over time with the frequency of *drop* as a dependent variable.

7.3.3 Factors in the overgeneralization and retreat

As we saw in the inspection of the various settings, the model overgeneralizes *fall* to a transitive frame in all cases. This is not strange given the design of SPL: the model rewards expressiveness strongly, and if no alternative to a transitive construction with *fall* as the word expressing the EVENT is present, SPL will simply use that pattern. Alternatively, it uses the less expressive, but very well entrenched 'Agt fall' pattern, in which an intransitive is combined with the word *fall*. Overgeneralization is, as it were, the default state of the model: in its desire to be expressive, it will use whatever means it has available to express as much of the conceptualization of the situation as possible.

We have also seen that the model can overcome this overgeneralization, but that the alternative has to be frequent enough to outcompete *fall*. When *drop* is highly infrequent (77.5 times as infrequent as *fall*), it will not outcompete *fall*, but when it is less infrequent ('only' 7.8 times as infrequent), it will. This opens the interesting possibility that we can model the regularizations of linguistic systems through usage processes with SPL: as a diachronic model, SPL would predict that *drop* would fall out of use if its frequency were 10, and *fall* would become a transitive verb. If *drop* has a frequency of 100, however, it would remain stable in the language.

The other interesting effect is that of the frequency of *fall* with a causative meaning. If this pattern is seen often, the model is quicker to use *drop* as the expression of the causative meaning. This is remarkable, given that SPL does not negatively reinforce (or: inhibit) non-observed grammatical constructions for words. Why, then, does the frequency of causative-but-intransitive *fall* matter? It seems to me that the causative-but-intransitive *fall*-construction is acquired more readily given this setting. This construction prevents a more generic construction (with any role as the first constituent and *fall* as the second constituent) to be acquired. It is this latter, generic-intransitive-*fall* construction that causes the model to overgeneralize, and if it is 'latently pre-empted'[2] by the 'Pat fall' patterns, the 'Agt drop Pat' patterns have more of a chance of being produced.

The two factors involved in the retreat from overgeneralization show that SPL can account for explananda E4 and E5: the model overgeneralizes and retreats from it, and we can study how the frequencies of the various constructions play a role in this. A high frequency of *fall* with a causative meaning 'latently pre-empts' the use of transitive *fall*, and a high frequency of *drop* straightforwardly pre-empts the use of transitive *fall*. This suggests that pure entrenchment has no role to play and is a mere epiphenomenon. Given the various findings in experimental studies, I will leave this suggestion to future research.

The fact that SPL never produces 'Pat fall' patterns (i.e., patterns with the patient of a CAUSE-FALL event as the subject) may indicate that the expressiv-

[2] I say latently because the 'Pat fall' patterns are never produced – they do, however, take reinforcement mass away from the 'Any-Role fall' pattern.

ity constraint on generation is too strong: the model finds both the erroneous and correct patterns with two expressed arguments more likely in all cases, because they express more of the situation. It may be that taking the discourse salience of the participants into account remedies this.

7.4 Discussion

In the production experiment, SPL proves to perform reasonably well on the various tasks, making the model fully satisfy desideratum D2 (comprehensiveness) now. We have seen that the model omits increasingly less arguments over time (explanandum E1), but does not simulate the prevalence of subject omission (E2). I argued that this latter effect is due to either the model having no notion of discourse salience or its lack of a right-edge biased, a notion well established by models such as MOSAIC (Freudenthal et al. 2010).

One may wonder why I made such an effort at analyzing the errors the model makes. I believe it is in the things that the model does not do 'right', according to the target utterance, that we see how it works. The error analysis revealed the fact that lexical abstraction and grammatical abstraction seem to work differently; whereas it does not hurt to abstract any and all abstractions over grammatical constructions (they are pre-empted by more concrete ones anyway), abstracting over lexical constructions is problematic, because overly abstract word meanings emerge. This has theoretical consequences. Does it, for instance, mean that they are, despite the constructivist axiom of 'everything is a construction', different beasts? I would not be willing to draw that conclusion yet, but this is an issue that is definitely in want of further attention.

Similarly, I found that many overgeneralizations were not overcome given the set-up (maximally concrete features such as FALL and no suppletive cases for verbs like *fall*). The addition of the latter, when *drop* is defined as CAUSE-FALL, surely helps, as we have seen in section 7.3, but then the question remains: how do we implement a system in which the violation of some of the conceptual properties of the situation is allowed in a highly restricted way. Again, like the condition on expressivity, we could argue that the model has to be able to produce analyses for a situation that include features not present in the situation, at the cost of some penalty. This would allow the model to produce argument-structure patterns that match the situation better, but that also are overly specific in their features (and therefore penalized).