SCHEMA THEOREM IN LANGUAGE ACQUISITION

A Rags to Riches Story

Sean McLennan

Dr. Laurent Dekydtspotter and Dr. Steve Franks
L700: Language Acquisition
December 13th, 2000

1. INTRODUCTION

Learning a language is a Herculean task; one that children perform with relative easy. Exposed only to a language environment – one ripe with errors, incomplete utterances, and no "goodness" information – a child can quickly and expertly acquire a communication system that is symbolic, combinatorial, productive, and expressive. With appropriate awe, linguists have posited that such a feat could not be performed without a boot-strapping mechanism of some sort; since Chomsky (1965) we have assumed that the language stimuli to which a child is exposed is impoverished and that to compensate for poor or indeterminate quality and inadequate quantity of input, humans come with a stock set of discrete, symbolic features and parameters commonly known Universal Grammar.

Considering the state of academic thought in Linguistics, Psychology, Computer Science – all the areas that have contributed to the modern study of Cognitive Science – this claim was both warranted and understandable. However, while other disciplines have increasingly shunned logical, symbolic, and a temporal models of intelligence and behavior due to overwhelming evidence to the contrary, modern theoretical linguistics has remained committed to the same framework.

This, I believe, is due to three reasons: 1) Linguists remain convinced that language can be 100% dissociated from other cognitive abilities; 2) Linguists remain convinced that language can be "lifted out" of the physical bodies within which it is instantiated; and 3) in general other disciplines are linguistically naive and have little respect for what linguists do. Given these facts, it is unsurprising that Linguistics as a field would be swayed by research performed in other areas. I believe that the "Poverty of Stimulus" argument in particular has held up over time because no one has provided linguists with a differing view of what constitutes the "input" (i.e. the language data), nor what constitutes the resulting phenomena (i.e. language, itself). If neither of these well-defined entities are challenged, there is no reason to challenge the underlying intuitions that gave rise to "Poverty of Stimulus".

The purpose then of this paper is to present a differing view of one of those well-defined entities – the input. I maintain that what constitutes linguistic input differs dramatically from what we have previously thought and provide a new conception that is more parsimonious with current research in other cognitive disciplines. I argue that we can better understand this new notion of input and how it can give rise to a system as complicated as language through an abstracted form of Schema Theorem which has been used as a domain specific explanation of how Genetic Algorithms perform efficiently. Finally, I present a first step in trying to experimentally support to this new conception of input via a computational model of the acquisition of grammar.

2.Genetic Algorithms and Language Acquisition as Analogous

2.1. Genetic Algorithms

Originally developed as an tool to study natural evolution, genetic algorithms have proven to be an efficient tool in computation. The working assumption that fuels their functionality is that nature's principle of "survival of the fittest" is an effective problem solver. For an in depth introduction of the mechanics of GAs, the reader is referred to Mitchell's (1996) excellent discussion; however, it is necessary to outline a few basics.

GAs consist of three typical elements: 1) a population of chromosomes (usually a bit-string of ones and zeros), 2) a fitness function, and 3) a process of mating chromosomes to produce a new generation of "offspring". The fitness function is a characterization of a problem to be solved and each chromosome represents a possible solution. Based on the fitness function, chromosomes are assigned a "fitness" evaluation depending on how well they solved the problem. The more fit chromosomes are "mated" by a process a crossover and / or mutation that produces the next generation of chromosomes.

A simple example would be to search for the maximum value of a mathematical function, say, $f(x) = \sin(x)$. In this case the bit-string chromosomes represent real numbers. In the process of evaluation, the bit-string is converted to a decimal number, entered into the equation as x, and assigned a higher fitness the higher the resultant value. The chromosomes with the highest fitness ratings are mated producing the next generation. After several generations, a chromosome with 100% or near 100% fitness usually emerges.

There is a strong feeling that with GAs you are "getting something for nothing"; beginning with a relatively small, random sample of chromosomes, very quickly you can converge upon a good solution to your problem. For example, give a chromosome of length l = 20, there are 2^{20} = 1,048,576 possible solutions and yet it is usually possible to find a perfect solution by judging only a few hundred of those. It doesn't seem that the input to the system, those few hundred chromosomes, could possibly be sufficient. It seems that the input is impoverished.

2.2. Language Acquisition and Poverty of Stimulus

Before Chomsky in the 60's, language acquisition theory was primarily the domain of Skinner and the psychological school of behaviorism (Skinner, 1957; Chomsky, 1959) and Chomsky did us a great service by pointing out that it is not possible to account for the richness, nor universality of the phenomenon of "language" within the behaviorist framework.

However, by showing that language was not the product of reinforcement learning, Chomsky also demonstrated the complexities involved in acquiring language. A new framework to account for it was required, to which end Chomsky proposed the "Language Acquisition Device" (LAD) – an innate mechanism by which a generative grammar is produced based on input. (Later revisions make Universal Grammar (UG) synonymous with the LAD.) One of the primary motivations behind the LAD is the productive nature of language; i.e. that we can both understand and produce utterances that we have never seen before in such a huge variety that it may as well be infinite. Clearly, the linguistic input is finite.

This point has become known as the "Poverty of Stimulus" argument[1]. Cook & Newson (1996) provide a succinct characterization:

> **The poverty-of-the-stimulus argument, otherwise known as Plato's Problem, claims that the nature of language knowledge is such that it could not have been acquired from the actual samples of language available to the human child.**
>
> Cook & Newson (1996:86)

[1]Chomsky makes a formal distinction between "Poverty of Stimulus" and "Degeneracy of the Data" which refers to the fact that the input is not 100% grammatical; there are performance errors etc. that further complicate acquisition since errors are not labeled as such. For the sake of simplicity, I will not make the distinction since it is not a focus of this discussion.

Moreover, they simply summarize the necessary rational steps followed to conclude that some aspect of syntax is part of the LAD/UG and not learned from the input:

1.

Step A: A native speaker of a particular language knows a particular aspect of syntax. Ex. structure-dependency, Binding Principles, etc.
Step B: This aspect of syntax could not have been acquired from the language input typically available to children.
Step C: This aspect of syntax is not learnt from outside.
Step D: This aspect of syntax is built-in to the mind.

Cook & Newson (1996:86)

These steps can and have been extended to phonology, morphology, etc. and provide a general guide to how to *demonstrate* that a particular aspect of language is part of UG.

To counter many of the previously hypothesized methods of language acquisition (such as simple imitation, or correction through negative evidence), it was necessary to examine the nature of the linguistic input available to the L1 learner. Again, Cook & Newson (1996) provide a simple summary of Chomsky's observations:

2.

Requirements on the language evidence for the child

- **positive evidence requirement**: in principle children must be able to learn language simply from examples of language spoken by others (positive evidence), without correction, explanation, etc. (negative evidence).
- **occurrence requirement**: any type of evidence needed by the child must be shown to occur in normal language situations; for example correction does not normally occur.
- **uniformity requirement**: the type of evidence must be available uniformly to *all* children regardless of variations in culture, class, etc. (since all children acquire their L1)
- **take-up requirement**: children must be shown to make use of this type of evidence

Cook & Newson (1996:92)

I do not intend to counter any of the rationale or evidence behind (1) or (2) – in fact, I whole-heartedly embrace them as well-founded and entirely valid. The focus of my argument – what causes modern linguistic theory to be in conflict with other cognitive disciplines – is the underlying assumptions in Step B of (1) and the consequent methodologies that "demonstrate" that some aspect of language "could not be acquired from the language input typically available to children". To illustrate what has usually been considered sound evidence for Step B, let's take a closer look at the example that Cook & Newson provide concerning Binding Theory[2]:

3.
a) Helen said that Jane$_i$ voted for herself$_i$.
b)*Helen$_i$ said that Jane voted for herself$_i$.

Cook & Newson (1996:84)

These sentences exhibit the fact that the antecedent of reflexive pronouns must be within the same sentence or clause (local domain) and that otherwise possible antecedents outside the local domain cannot be considered as possible coreferents. After presenting the sentences, they go on to say: "Nothing would tell the children that they are wrong; no context could let them unerringly distinguish the binding of anaphors and of pronominals."

This implicitly assumes that the only source of information that the child has access to at this point is the utterance itself. It implicitly denies that the cumulative experience that the child has had interpreting anaphors and pronominals up to that point can be brought to bear in interpreting the sentence. Their experience with correctly interpreting binding relationships, considering that children only begin to produce sentences of this level of complexity at around the age of 5 years (O'Grady, 1997), has already been considerable. Furthermore, children demonstrate a preference for binding anaphors and pronominals to the closest possible antecedent from the outset (O'Grady, 1997) and it is really the following sentence that they must learn is ungrammatical:

4. *Helen said that Jane$_i$ voted for her$_i$.

(4) is something that children have a great deal of trouble with until about age 5 (O'Grady, 1997). Certainly, "bind anaphors to the closest antecedent" is obtainable from the input and if it is never contradicted in the input, it is unlikely that the child would produce anything different. If the correct interpretation of (4) requires a statistically significant number of utterances that contradict the incorrect interpretation (which there are likely to be many: in "John said Mary hit him", "Mary" is not a possible antecedent because of the gender discrepancy which by age 5 children do not have a

[2]Binding Theory governs the relationships between anaphors (ex. myself) and pronominals (ex. I, me) and their antecedents.

problem with) we would expect there to be a developmental delay in consistently, correctly interpreting (4). This is precisely what we see (O'Grady, 1997). This delay *has* been recognized as problematic for claiming that Binding Principles are innately part of UG; however, because Poverty of Stimulus and UG are assumptions of the framework within which the data is being analyzed, complicated explanations for why this occurs *despite* UG, are artificially constructed including a proposal that children mistakenly interpret pronominals (which in most other contexts they use perfectly) as reflexives (O'Grady, 1997). What is presented here does *not* constitute conclusive evidence that Binding Principles cannot be learned from the input; it seems quite possible that the opposite is true. Step B has not been rigorously demonstrated on the basis of (3).

Another example of this style of argumentation from O'Grady (1997) that is used as a counter example that analogy can be used as widely used inference method in language acquisition:

5.

a) It is likely that John will be delayed.
b) It is probable that John will be delayed.
c) John is likely to be delayed.
d)*John is probable to be delayed.

O'Grady (1997:246)

(5) is intended to show how analogy-making will lead to erroneous usages. I.e. we would expect that if analogy was used, (5d) would be perfectly grammatical based on the fact that grammaticality of (5a), (5b) and (5c). Such an argument denies that the child has access to any sophisticated information about word categories, syntactic structure, morphology, etc. which considering that a child must have adult-like competence to produce such sentences, seems *highly* unlikely. Examples such as this are typically considered enough to rule out analogy completely as a source of productivity in language despite the simple fact that since "probable" is an adjective, and "likely" is an adverb make it a very unlikely analogy to make. Analogy is very common among young children and is a well attested source of language change (Hock & Joseph, 1996). Removing the strong "logical inference and reasoning" connotations of the word "analogy" and defining it as it is now commonly used in Psychology and Cognitive Science[3], analogy becomes such a powerful concept that it has been even hypothesized to be the *core* of human cognition (Hofstadter, forthcoming). It cannot be concluded

[3] I.e. recognizing salient similarities between two entities such that in certain contexts they may stand in for one another.

on the basis of (5) that the difference in the usage of “probable” and “likely” is not learned from the input. Step B, in this case as well, has not been rigorously demonstrated to be true.

In the reality of research in linguistics, Step B, is not performed at all, let alone rigorously argued. Cross-linguistic data that a phenomenon is exhibited in more than one language has replaced “cannot be learned from the input” as the criteria for proposing innate principles and parameters of language. The Poverty of Stimulus argument behind Step B has become an a priori assumption that is self-propagating and no one has seriously challenged it from within Linguistics despite the fact that it has largely been thrown out in Psychology and Cognitive Science. Brooks (1991), Thelan & Smith (1994), Kelso (1995), Port & van Gelder (1995), Holland (1998), Elman et al. (1998), and Clark (1998) are typical of approaches taken to the study of all aspects of cognition, intelligence, and language that explicitly reject the arguments behind the Poverty of Stimulus.

Simply throwing out UG, does not solve our problem though. As with GA’s, without innate, symbolic knowledge of features, principles, and parameters of language, it appears that we are “getting something for nothing” – from a (somewhat) arbitrary starting state of neurons, based on limited input, a full language is acquired. It doesn’t seem that the input to the system could possibly be sufficient; it still seems to be impoverished. This intuition cannot be satisfied until there is a new understanding of the nature of the input.

2.3. Schema Theorem in GA’s

Unlike language, we have access to every detail of every generation of a GA. This fortunate characteristic of computational models makes a full and plausible analysis of their behavior tractable and is one of their most appealing properties. In order to explain how GAs can perform such an effective search of solution space, Holland (1975) introduces the formal notion of “schemata”. Schemata are essentially similarity templates that describe a set of chromosomes that share values in certain positions. To describe a schema we add the wildcard, “*”, to our string notation. Thus, the schema *0 describes a subset of 2 chromosomes: {10, 00}; 1** describes the subset {100, 101, 110, 111}, etc. Of course, schemata with no *’s describe sets of 1 element — i.e. the notion of schemata subsumes individual chromosomes. The total number of possible schemata given a chromosome length of l is 3^l since there are three possibilities at each position: 1, 0, or *.

We could replace the term “schema” with “category” to make the description more intuitive. What the above states is simply that each individual entity, say a particular dog that you encountered on your way home, is an instantiation of a very constrained category of only itself (perhaps temporally distinct experiences with the same dog: that dog today, that dog tomorrow). The category “Golden Retriever” includes not only that dog, but all dogs of the same breed. The category “dog” includes all Golden Retrievers and all other breeds of dog. So far, this is nothing remarkable.

The relationship between schema and the chromosomes they represent works both ways; the chromosomes, whose values are set, are instantiations of 2^l schemata since each position may take its actual value or the wildcard. To see this, let’s examine an example of a short chromosome length, $l = 3$, for which there are $3^3 = 27$ possible schemata. The chromosome 101 is an instantiation of $2^3 = 8$ of those 27 schemata: {***, 1**, *0*, **1, 10*, 1*1, *01, 101}. Of course, we can add more levels of categorization and schemata, themselves, can be thought of as instantiations of other schemata. For example, *1**1* is as much an instantiation of ***1* and *1*** as the chromosome 010010.

Returning to our dog on the street, it is an instantiation of the category of itself and is simultaneously an instantiation of “all Golden Retrievers”, “all dogs”, “all animals” – even more abstract categories like “all dogs that live on that street”, “all dogs owned by your neighbor”, “all dogs of the same color”, “all dogs of a similar size”, “all dogs named ‘Rover’”... the list could go on indefinitely along any dimension to any level of abstraction.

This is again unremarkable; the important insight is that when a single chromosome is being judged by the fitness function, the fitness of all the 2^l schemata it represents are *also* being judged – implicitly. By the same token, the fitness of an individual chromosome is also in a sense a function of the fitnesses of each schema it represents. Taken individually, this doesn’t provide us with much information. Crucially, chromosomes exist in a population and thus we can define the fitness of a particular schema as the average of the fitnesses of all instantiations of that schema in the population. It is important to note that this figure is never actually explicitly calculated – it is information that is *implicit* in the fitnesses of the individuals. A GA can make use of this implicit information because selection for the next generation is biased towards highly fit individuals. Since each chromosome’s fitness can be thought of as a function of the schemata it instantiates, the fitness function is also implicitly biased towards selecting highly fit schemata.

Implicit parallelism is the primary power of the GA. What we have here is a process that makes use of category information ***without explicit reference to the categories*** allowing a phenomenal reduction of computational load. Suddenly, taken in this light, a single generation of a GA explodes with rich information; the input contains orders of magnitude more information than it originally seemed, and it can no longer be thought of as impoverished. 200 chromosomes give rise to an enormous number of schemata and it is much clearer how it can effectively traverse solution space to the correct solution.

2.4. Schema Theorem in Language Acquisition

I intend to argue that Schema Theorem and this profound notion that processes can act on category information without explicit reference to the categories themselves, is generalizable and can be applied to other domains. Specifically, I argue that it can provide us with a profound insight into the nature of language acquisition that is consistent with both the facts observed in linguistics over the last 50 years and also the research movement towards non-innate, developmental embodied cognition in the other cognitive sciences.

The distilled, salient properties of GA's and Schema Theorem are presented in (6) below.

6.
   a. A single representation (ex. a chromosome) implicitly contains a huge amount of information about the categories to which it belongs.
   b. Processes that act on those representations (ex. selection) can implicitly make use of all that information in parallel.
   c. Valuable information (ex. a solution) can emerge from the repetition of the same process.

Roughly, the analogy that I will draw is shown in (7).

7.

Schema Theorem		**GAs**		**Acquisition**
representation	➢	chromosomes	➢	memories
process	➢	selection	➢	learning
product	➢	solution	➢	salience

Chromosomes ➢ Memories

Esther Thelen and Linda Smith (1994) in their arguments against innateness comment on the work of Newport (Johnson and Newport, 1989; Newport, 1990) and her studies in language acquisition.

> "Newport speculates that young children learn deep syntactic properties more readily than adults precisely because young children are cognitively 'deficient.' Newport suggests that when mature persons with all their cognitive resources try to learn a language, they attend to and remember all that they hear and the full range of meanings in context. Very young children are, however, cognitively deficient. They cannot hear, or remember, or think about it all. They only pick up bits and pieces of language."
>
> Thelen & Smith (1994: 33-34)

It is an important insight that children are "cognitively deficient"; however, Newport's claim that children "cannot hear, or remember, or think about it all — they only pick up bits and pieces of language" requires further qualification. It is clearly not the case that infants have primitive or impoverished perception and memory — in fact their senses are quite sophisticated and are capable of receiving and "storing" the same range of input that adults do, particularly with respect to linguistic tokens (Kuhl et al., 1992). Thelen's own work points this out; in discussing Rovee-Collier's (1991) experimental results of a task in which babies learn that kicking moves a mobile that is tethered to their leg, she states:

> "Over time, [the memory of the task] faded, although simply seeing the mobile would reactivate it. Most important is that this action memory was highly specific to the training situation. If Rovee-Collier changed the mobile, or even the designs on the pads that lined the cribs in which infants originally learned the task, infants forgot that kicking a lot makes the mobile move more. The action memory was highly tied to the learning context."
>
> Thelen (1995: 96)

It is clear from these results that even infants have quite detailed sensory memories — even the patterns on the crib liners could affect the learning of a task. We can conclude then that "cognitively deficient" does not mean that infants can not perceive and process complex stimuli in their environment; it is that they cannot perceive and process the salient, discrete, and symbolic aspects of their environment. In the task above, the infant did not comprehend that the crib liners were not a salient aspect of the task — the patterns remained 100% correlated to the task and thus are as likely a factor as any other stimuli until experience demonstrates otherwise. Indeed, Rovee-

Collier went on to show that if the child was trained on the task with several different pads, the same memory effects were not apparent (Thelen, 1995: 96).

What we can draw from Rovee-Collier's results is that infants, in learning a novel task, "record" the experience in its totality — all visual, auditory, tactile, olfactory information, etc. Being "cognitively deficient" they have no understanding of what information in that experience is salient; it is all relatively new, and thus an unparsable, meaningless blur. Adults on the other hand, readily extract the salient features of a scene, utterance — any sensory input — and only attend that which is particularly important. All other information that is irrelevant is immediately recognized as such and discarded. This skill of determining salient and non-salient information is precisely what the infants are in the process of learning.

Ultimately, this process will be definition result in higher-level organization in the brain. The fact that linguistic ability is highly correlated with the temporal and parietal lobes of the left hemisphere is often used as an argument for the uniqueness of language and by extension its innateness. However, the question of whether higher-level brain structures are specialized for linguistic tasks and whether those structures are genetically encoded for are two very different questions. Obviously we are genetically predisposed to *develop* particular neural structures, but it has been demonstrated that development is crucial in what structures eventually appear. For example, Sharma et al. (2000) dramatically showed that visual stimuli redirected into auditory cortex resulted in the organization of structures found in visual cortex. No one would deny that areas of the brain are specialized to deal with visual vs. auditory information, nor would they deny that genetics play a role in setting up the supplying the necessary conditions to form such areas. But it is also clear that, say, ocular dominance columns arise from development *not* from genetics.

In precisely the same way, it is likely that higher-level cognitive abilities correlated with higher-level neural structure are similarly self-organizing. As Rovee-Collier's experiments imply, newborn infants are exquisitely sensitive to changes (and consistencies) in their stimuli although they are not capable of parsing it all in an adult-like manner. The learning that occurs at this stage probably involves very low level, procedural and perceptual memory and is largely passive. It is also likely that every experience has strong effect on learning.

Although it is not possible to cleanly divide these experiences into discrete chunks, we do have a sense that they are delineated to some degree. The experiences relevant to this discussion are those that involve linguistic tokens which can be delineated roughly by periods of acoustic activity bracketed by periods of silence; other factors, such as timing phenomena or prosody, also play a role. For sake of convenience, I will refer to these roughly delineated linguistic experiences as "memories" and assume as per the discussion above that they include not just linguistic information but a totality of low-level sensory information.

Within the framework of Schema Theorem, I propose that these memories are to language acquisition, what chromosomes are to GAs. Both are a representation of information and both exist in populations. This latter claim needs to be clarified with regards to memory; all sensory input is processed at its lowest level by the same neurons upon which it has a direct, passive, Hebbian learning effect. In this way, memories are stored distributionally within the same physical substrate. Thus temporally discontinuous memories can be thought of coexisting together in a "population".

Already the view of what constitutes linguistic input has changed dramatically – it includes the entirety of sensory information being processed by the child that is concurrent with linguistic tokens in the environment. This does not presuppose that the child cannot "stream" information appropriately and that there is no distinction being made between modes of sensory information. Correlations are learned very quickly (cf. Rovee-Collier's experiments) and so the fact that infants distinguish between speech sounds and other sounds by 4 months (Kuhl et al, 1992) is unsurprising. By 4 months they have already been exposed to more than ample evidence (a strong correlation between speech and caregivers for example) that the two sources of sound are different in a meaningful way. The important point is that all modes of sensory experience in all their full, information rich, continuous, detail are available.

Referring back to (2) and Chomsky's requirements for language evidence, this enhanced form of linguistic input is a priori consistent with the first three: it is positive, it occurs, and it is uniform. The final requirement – the "take-up requirement" – has not been explicitly demonstrated. In part that would be a goal of further research in this direction. However, considering the nature the studies presented thus far and the promising results that are presented further on, it seems improbable that the this final requirement would not be met.

Selection ➢ Learning

"Learning" is a rather broad concept that can be defined on many levels. "Learning" here, is to be understood on a neuronal level and I will take a simple characterization made by Rumelhart (1997) for learning in connectionist systems:

> "Changing the processing of knowledge structure in a connectionist system involves modifying the patterns of interconnectivity. In principle this can involve three kinds of modification:
> (1) development of new connections;
> (2) loss of existing connections;
> (3) modification of strengths of connections that already exist.
> . . .
> "Virtually all learning rules for models of this type can be considered variants of the *Hebbian* learning rule, . . . if a unit u_i receives input from another unit u_j t a time when both units are highly active, then the weight w_{ij} to u_i from u_j should be *strengthened.*"

Rumelhart (1997: 213-214)

Since this definition is originally based on real brains (Hebb, 1949), it is at least analogous to learning that occurs in real neurons — that two highly active, connected cells develop a stronger excitatory relationship and thus repeated activations of the same neurons and groups of neurons result in gradually increasing strengths of activations over time. Additionally, associations between different areas of the brain can be learned through "convergence zones" — a neuronal grouping that directs the stimulation of anatomically distant regions (Damasio and Damasio, 1994).

Although unique episodic memories are stored (by high-level convergence-zones) "cumulative memories" are also generalized (by low-level convergence zones). It is these "cumulative memories" that correspond to our "population of chromosomes". As mentioned before, they are stored within the same physical substrate.

However, similar experiences are not *identical*. Areas of experiences that are most similar will cause the highest increase of activation — the highest degree of learning. To express this idea in a visual medium, let's pretend that it is possible to represent a sensory experience / memory — all its visual, auditory, tactile information etc — as a single waveform. Examine the graphs in below:

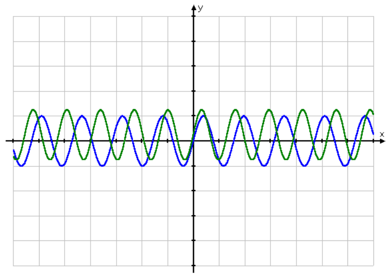


Figure 1.

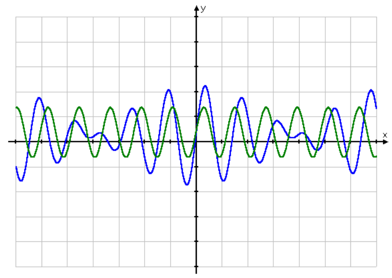


Figure 2.

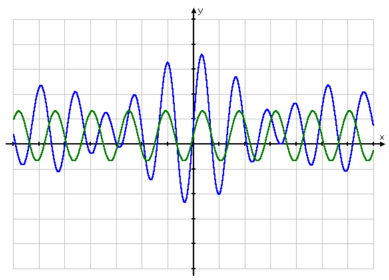


Figure 3.

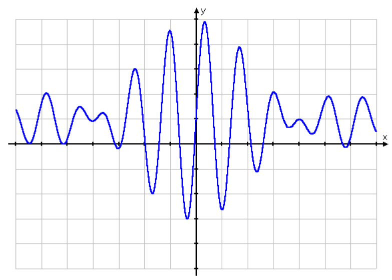


Figure 4.

Each of the first three graphs has two wave-forms: the blue one corresponds to the collective learning of past memories, and the green one corresponds to a new, similar experience. Adding the waveforms together to create more complex waveforms is analogous to learning. If peaks and troughs correspond to strength of activation, by (Fig. 2), we already see that certain areas are more highly activated than others. These areas correspond to the areas of the constituent waveforms (i.e. memories) that were "in phase" (i.e. most similar). Thus high areas of activation develop where experiences are most alike. This process is essentially the same as the development of feature detectors in connectionist networks described by Rumelhart and Zipser (1985) only on a much larger scale, and is reminiscent of the story told by Thelen (1995: 96-98).

Learning then, is a process that takes advantage of the similarities between individual memories. The dimensions along which experiences can vary implicitly define possible categories – possible schemata. Unlike GAs which are fundamentally discrete and finite, our experience with the world is continuous. Consequently, the realm of possible schemata is infinite, although constrained by the physical properties of our sensory-motor system. However, because the input to the child systematically makes use of particular dimensions, schemata – categories – are guaranteed to emerge. For example a phonetic feature such as [+nasal] that regularly and consistently appears *in context* in the input will be reinforced by learning until the feature emerges as a "fit" schemata – i.e. a dimension within acoustic space within which meaningful distinctions are made.

Although learning and selection are not comparable as processes – they are dramatically different both in nature and output – with regards to Schema Theorem, they play the same role. It is important to emphasize again that this process of learning which makes use of category information is entirely *implicit*. At no point does the process make explicit reference to the patterns of similarity in the environment. Take the following utterance which one can easily imagine a child hearing:

8. The doggie went away!

Ignore for the moment all of the sensory information that is correlated with hearing this utterance and focus solely on the linguistically relevant aspects of this utterance. As a "memory", in the sense presented here, this single utterance is an instantiation of a multitude of categories. It implicitly contains information about every level of linguistic detail from phonetics through phonology, phonotactics, syllable structure, intonation, morphology, syntactic categories, syntactic structure, possible English utterances, pragmatics, turn-taking, all the way to semantics. Just as with a single chromosome, taken individually, this fact doesn't mean much. However, because the memory exists in a population, it holds a tiny piece of the puzzle that in combination with other memory gives rise to a complete picture of the whole.

As with GAs, when looked at from this perspective, the input to system suddenly explodes with richness; orders of magnitude more information is packed into a single utterance than previously acknowledged, yet because of the implicit parallelism of the learning process, the computational load is minimized and we have an inkling of how a child can converge upon a generative language system.

Solution ➢ Salience

"Salience" is yet another broad term that is used in a multitude of contexts. As alluded to, I will use it as "the understanding that a particular dimension along which memories can be similar is relevant for making meaningful similarity judgements", where "understanding" and "judgements" should be understood as being a low-level perceptual phenomenon, below conscious access.

Concrete examples will help elucidate this definition; in the Rovee-Collier experiment, the infants learned, through repeated, similar experiences, that the motor action of kicking was salient, not the visual cue of the crib-liner, for causing the mobile to move. English speakers learn that the tense/lax distinction is salient in the categorization of vowels, whereas Japanese speakers learn instead that length is salient. English speakers learn that particular prosodic cues are salient for determining sarcasm. Children learn that the presence of a object noun phrase is salient for determining that a new verb is transitive. Children learn that a the presence of a known transitive verb is salient for determining that the following unheard noun phrase is the object of the verb.

Salience also provides the dimensions along which analogies can be drawn (or productive rules can be performed). For example, the fact that “the cat” and “the dog” are often used with similar verbs (“pet”, “run”, “feed” etc) and associated with classes of sensory stimuli implies that when the child hears “feed the llama” for the first time, those other verbs can be used in a similar manner.

Salience allows for categories to be constructed – it is clear that language often behaves like a formal system. Chomsky claims that it is symbols “all the way down”, despite the fact that that is in conflict with all we know about cognition and development. Salience provides a means by which a formal discrete symbol system can arise out of a continuous environment. It is also clear that language often does *not* behave like a formal system (Port & Dalby (1982) for example). While salience allows for categories to be built upon in a hierarchical manner, it also allows dimensions of categorization that leap across the formal levels linguistics describes (for example, the fact that intonation can override the semantic interpretation of an utterance based on the words it contains).

Salience is defined by the “good” – i.e. relevant – dimensions of “sensory-experience space” in an analogous way to how a solution is defined by the “good” – i.e. highly fit – points in “solution space”. Both salience and solutions are emergent properties of a process acting on representations. Neither, a priori exists within the system, although they do a priori exist in the world.

2.5. Implications

The picture of linguistic input presented so far is dramatically different from the one traditionally accepted and if its predictions, applications, and explanatory power were borne out, it would force a significant change of perspective in modern theoretical linguistics. However, it is not incompatible with most of the work that has gone on in the field – the major changes would be in focus and perspective.

For example, nothing presented here denies that there is a biological foundation for language. That would be analogous to saying that there is no biological basis for sight – that would be absurd. Nothing here is incompatible with a notion of UG as a set of universal constraints on natural language; Chomsky's Steps presented in (1) are not invalidated. However, our conception of the nature of UG might need to shift and we might add a requirement that proposed elements of UG have a biological grounding. Certainly, Step B would become a focus of attention instead of an assumption of the framework.

More concretely, the Schema Theorem picture of linguistic input makes certain predictions about the language acquisition that are not entirely consistent with the accepted paradigm. A sampling is presented below:

1) acquisition will proceed from the storing of "unparsable" tokens to partial parsability to full parsability and potentially back again to memorized "chunks" that can be parsed but might not regularly be
2) this is fundamentally a statistical approach to language acquisition; we would expect statistical frequency and statistical correlation to play an enormous role in the stages of acquisition
3) a better defined theory of "salience" will be required to gain a profound understanding of how it is employed in implicit category formation
4) because "salience" is an emergent property of statistical regularity, acquisition of very similar grammars / lexicons is probable, but the acquisition of identical grammars / lexicons is highly unlikely

3. Testing the Schema Theorem Picture of Input

Whether or not Chomsky is right and linguistic input is impoverished, will ultimately prove to be an empirical question. It will be a two step process: 1) it must be shown that statistically based systems can acquire, if not language itself, then qualitatively similar properties of language; 2) it must be shown that the acquisition of natural language makes use of qualitatively similar statistically based processes.

If it can be consistently shown that claims about the impotence of data-driven acquisition systems are false, we must accept the possibility that the "Poverty of Stimulus" and our notion of UG is, in fact, not well-founded and seriously examine our assumptions about the nature of language. The first step is already well underway (Elman, 1995; Chalmers, 1990; Vogt, P. (2000); Batali (1998); Colunga-Leal & Gasser, ms.; Kirby & Hurford, 1997; Redington et al., 1998; Redington & Chater, 1998) although typically through Cognitive Science, not mainstream Linguistics. Below I present my own model to specifically explore the following question:

Can a statistically based, data-driven algorithm infer a grammar that:

a) is productive

b) is grammatical in comparison with the target grammar

c) has emergent categories

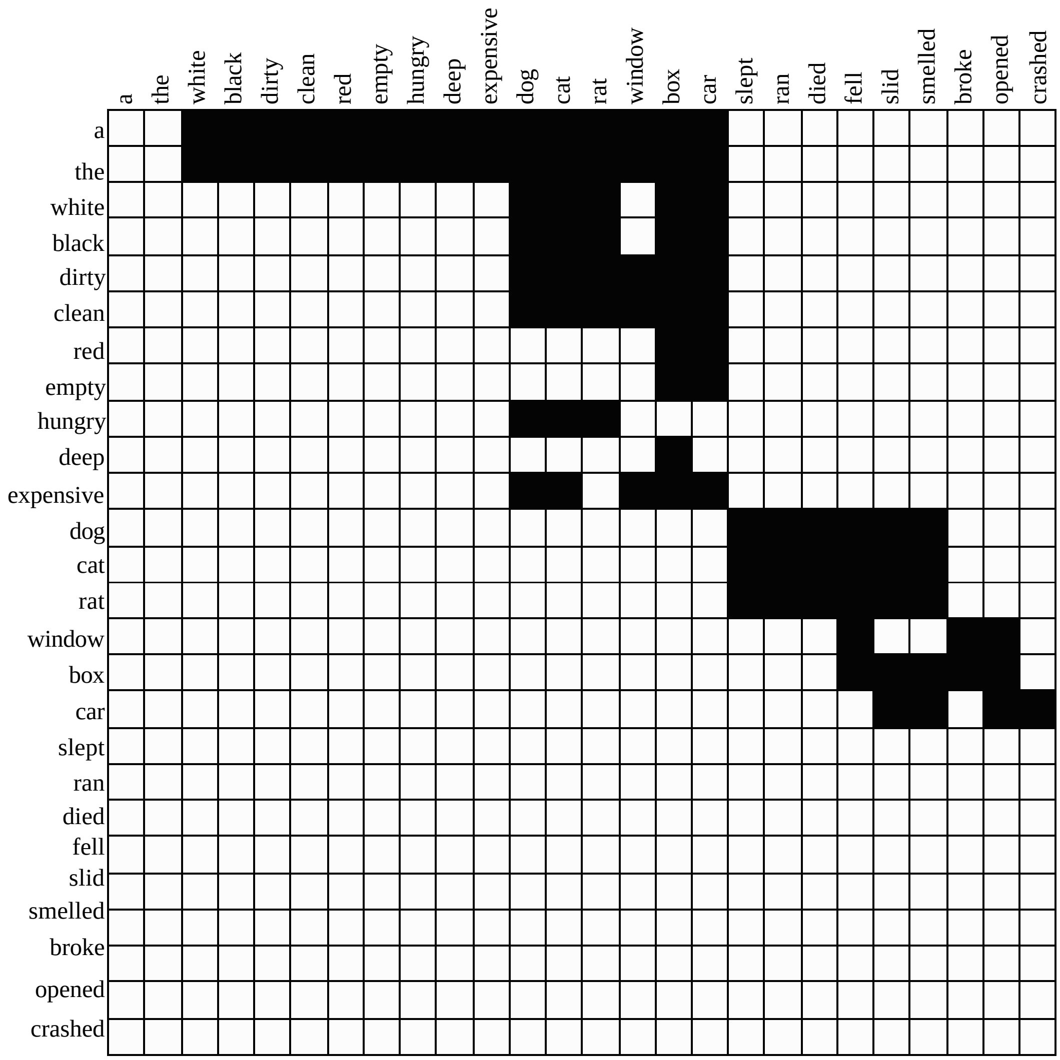
Unsurprisingly, its design does not perfectly conform to the ideal scenario presented in section 2. However, since section 2. is at the core of its inspiration, it does illustrate several of the ideas described.

3.1. INPUT

The input into the system is a number of utterances (varied for different trials) produced by the simple Markov chain in (9) represented as a transition matrix. I.e. given a word along the vertical dimension, black squares indicate that a following word along the horizontal axis is a grammatical transition. Words are used for the symbols for convenience, but since no semantics are represented in the system, they could have been any arbitrary symbol. Importantly, they fall into categories based on usage; there are four word categories (Det, A, N, V) and sub-categories determined by semantic relationships. It is also important to note that adjectives are optional creating some degree of grammatical complexity. The full list of 26 words is presented in (10).

The input utterances were of 3 types: full sentences (Det (A) N V); noun phrases (Det (A) N); and nouns alone (N). All of these are plausibly evident in child-directed speech. A typical sample of input utterances is presented in (11).

9



10.

Det	A	N	V
a	white	dog	slept
the	black	cat	ran
	dirty	rat	died
	clean	window	fell
	red	box	slid
	empty	car	smelled
	hungry		broke
	deep		opened
	expensive		crashed

11. <u>A typical Input Set - 25 Utterances</u>

```
the rat fell
window
cat
a hungry cat slid
box
cat
dog
the rat slept
the empty box smelled
a clean cat smelled
the box
the box
the empty box
box
a clean car
a red box
window
a clean car
the window
window
a clean car
window
the deep box
the deep box broke
the black rat slept
```

3.2. Learning

Being that this model was implemented on a serial computer using a discrete, symbolic programming language (Matlab), the implicit distributional aspects of learning could not be incorporated. However, there is no reason why similar processes could not be instantiated in, say a connectionist model.

That said, the system begins with zero knowledge and the construction of a new transition matrix proceeds through the following steps:

1. Each utterance is presented to the system as an unparsed chunk.
2. The system compares the chunk to the list of other chunks it already knows
   a. if it recognizes that a chunk it already knows is present within the new utterance, it will parse the utterance into: |before known chunk| |known chunk| |after known chunk| These new chunks are added to the "known chunk list".

b. if it recognizes no similarities between the new utterance and known chunks it simply stores the new utterance as an unparsed chunk.

3. Each time the system successfully parses a new utterance, it records that it saw a transition from one chunk to the following chunk by updating a matrix value with a simple formula: newval = oldval +.5 (1-oldval). This provides a measure of frequency that asymptotes to 1 as the frequency increases.
4. After all utterances have been processed, the system determines its smallest, unanalyzed chunks – atoms if you will – that comprise its vocabulary. Usually they are single words, but especially when the number of input utterances is small, there can by multi-word chunks present as atoms.

A new productive transition matrix is produced based on the observed transitions in the input

6. A comparison is made between the observed transition frequencies of each of the atoms and when two or more are found to be of statistically significant similarity, they are grouped together and their allowable transitions are merged. The dot product of the learned transition matrix and its inverse provides the measure of atom similarity and Z-scores that test the significance of those similarity ratings can be simply calculated.

Examples of the “Known Chunks” and “Atomic Level Words” based on the input in (11) are shown in (12).

12.

Known Chunks	Atomic Level Words
`the rat fell`	`window`
`window`	`cat`
`cat`	`a hungry`
`a hungry cat slid`	`slid`
`a hungry`	`box`
`slid`	`dog`
`box`	`smelled`
`dog`	`a clean`
`the rat slept`	`the`
`the empty box smelled`	`empty`
`the empty`	`car`
`smelled`	`a red`
`a clean cat smelled`	`deep`
`a clean cat`	`broke`
`a clean`	`black rat slept`
`the box`	
`the`	
`the empty box`	
`empty box`	
`empty`	
`a clean car`	
`car`	
`a red box`	
`a red`	
`the window`	
`the deep box`	
`the deep`	
`deep box`	
`deep`	
`the deep box broke`	
`broke`	
`box broke`	
`deep box broke`	
`the black rat slept`	
`black rat slept`	

As is immediately apparent, the processes employed by this model are painfully unsophisticated. It acts on a single source of information – transition frequency based on the input – and performs a similarity judgement based on a basic matrix operation and a simply calculated statistic. It is the stupidest possible (in ability, not significance) learning algorithm that could be imagined and all told is contained in less than 500 lines of code (see Appendix A), data analysis and all.

3.3. Output (Salience)

After it builds a new productive transition matrix, the system produces output utterances. For sake of simplicity, these utterances were restricted to ones built of atoms and were the equivalent of the new grammar's "full sentences", that is, utterance begins with an atom analyzed as a possible starting word and ends when a chosen atom has no allowable following transitions.

Additionally, as a byproduct of the similarity judgements category groupings are formed. Although, in this model, they are explicitly created and referred to, this is precisely the type of task that connectionist models are very good at. Regardless, the information used in creating categories is implicitly instantiated in the input set and so in a loose sense, the categories are an emergent property of the learning process, even in this model. The output utterances and categories produced by the trial we are exemplifying appear in (13).

13.

Category Groups

```
|slid| |smelled|
|cat| |box|
|smelled| |broke|
|window| |box| |empty| |deep| |black rat slept|
```

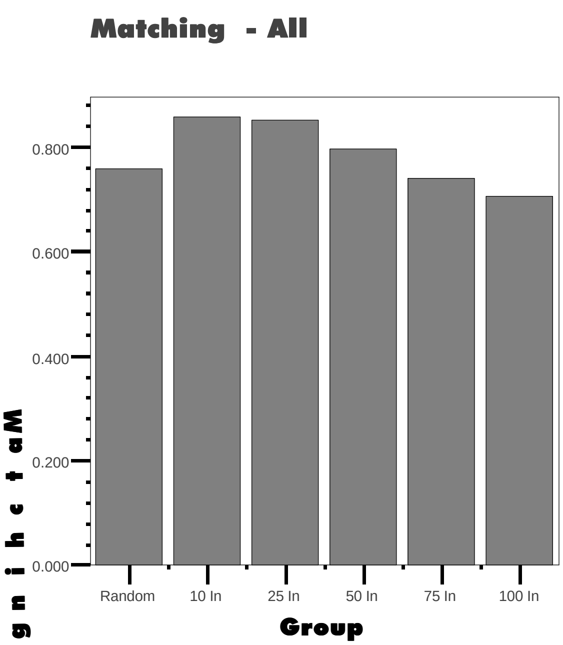
Output Utterances - ◆ indicates novel utterances

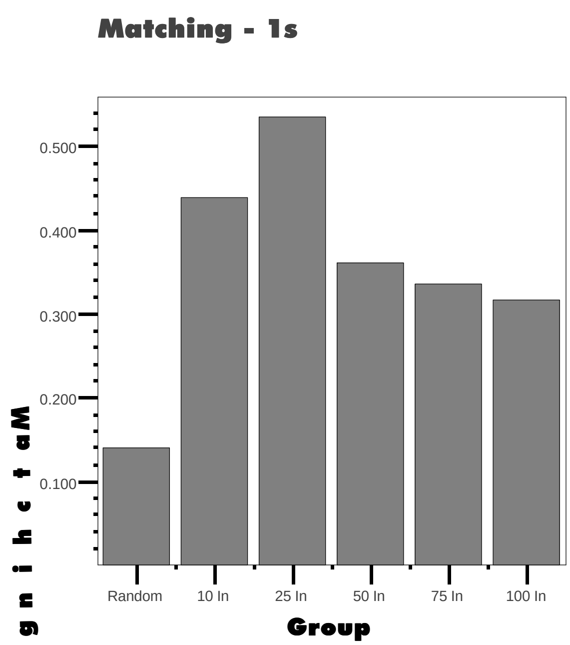
```
dog
a clean car
◆a hungry cat car
dog
◆a hungry a red empty
   window
◆a hungry a hungry slid
◆a red black rat slept
◆a clean dog
dog
◆a clean smelled
◆a hungry slid
◆a clean cat car
the window
◆a hungry the black rat
   slept
◆a clean box cat window
dog
the black rat slept
dog
◆the deep black rat slept
◆a clean black rat slept
◆a clean the box a red
   black rat slept
the window
◆a clean broke
dog
◆a clean a red empty deep
   empty window
```

3.4. RESULTS AND INTERPRETATIONS

In order to test the performance of the system a total of 600 trials were performed; 100 each of 6 groups: a randomly generated transition matrix[4], and trials on input sets of 10, 25, 50, 75, and 100 utterances. A total of 8 different measures were used to gauge the acquired grammars' performance

Matching - All **Matching - 1s**





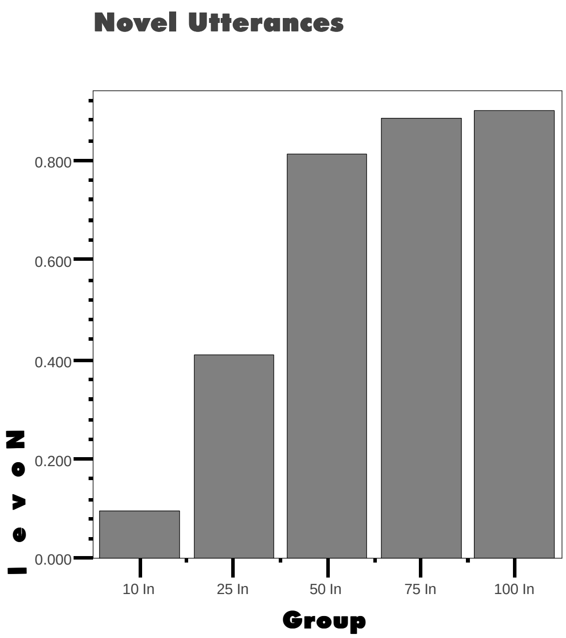
These two measures are direct comparisons of the original and acquired transition matrices. The first is a point-wise comparison of each matrix entry, the second only a comparison of matching "1s" between the original and acquired matrices. Despite the uniformity exhibited in the Matching-All measure all differences are strongly significant (even between the 10 and 25 groups) although in Matching-1s, the difference between groups 10 and 25 is the only non-significant difference. (The appropriate T-tests are included in Appendix B).

Interestingly, these results show an unintuitive decrease in accuracy of acquiring the target grammar that seems to plateau. However, is not necessarily a negative outcome – what is exhibited is an initial jump in accuracy as acquisition begins based on observed transitions, but then a decrease in accuracy caused by overgeneralization. I.e. the system is exhibiting stage-like behavior. Possibly,

[4]with a proportionate number of "on" or "1" entries

if the trials had continued, the trend would have continued and we would have seen a subsequent increase in accuracy after it bottomed-out.

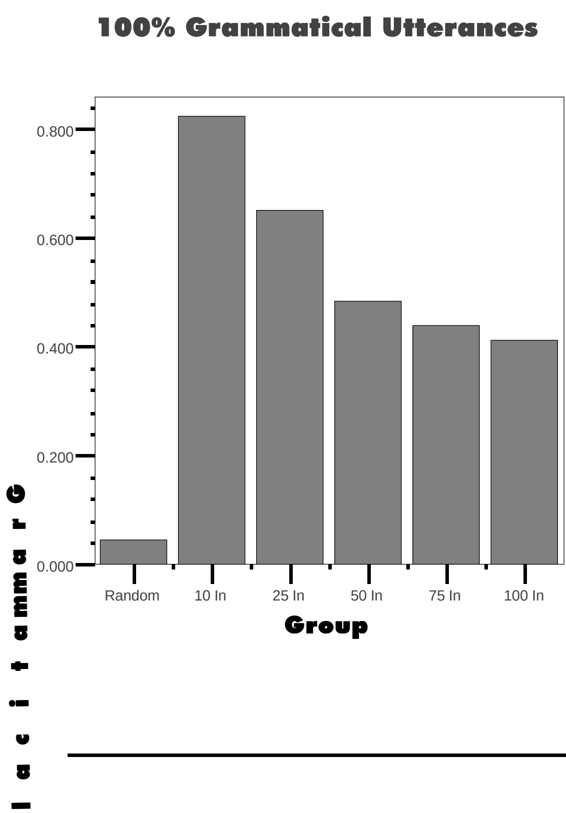
Novel Utterances



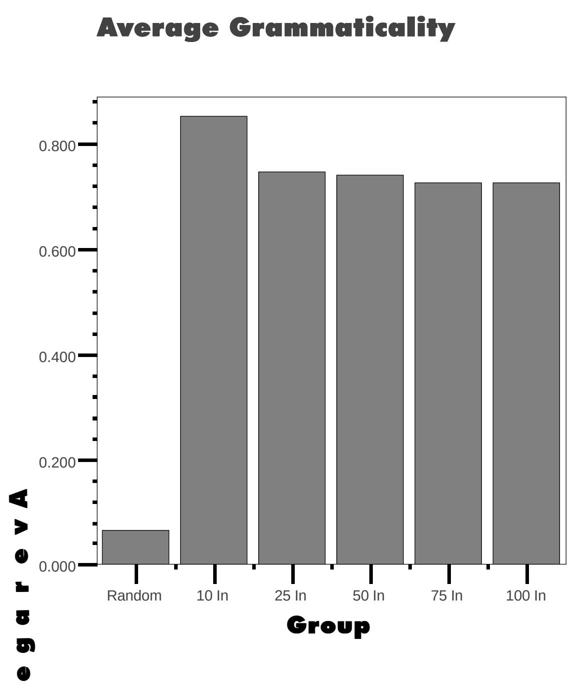
This measure is a simple tally of how many of the output utterances were not seen in the input – again each difference is significant with the exception of the final difference between the 75 and 100 groups.

Again, this is indicative of stage like behavior – after only 10 input utterances, most of the output is imitative, however productivity increases quickly and asymptotes after a critical point of around 75 input utterances. Even after only 50 input utterances, it is clear that the acquired grammar is highly productive.

Absolute Grammaticality

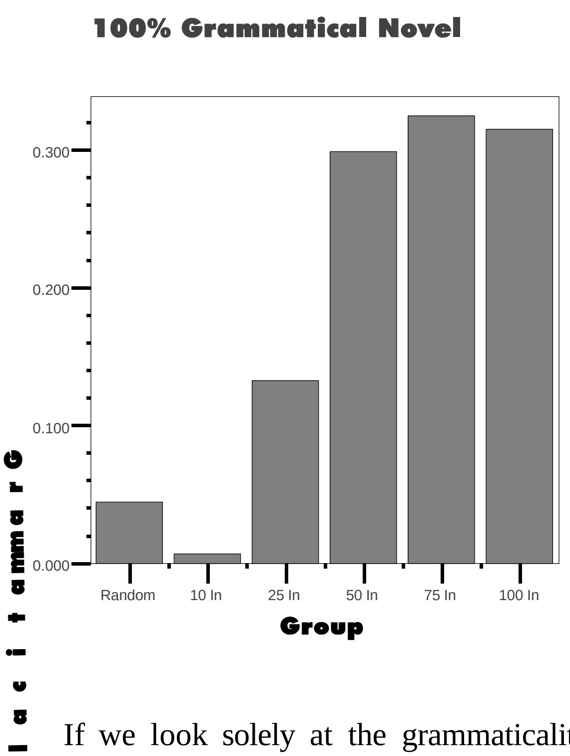


Average Grammaticality

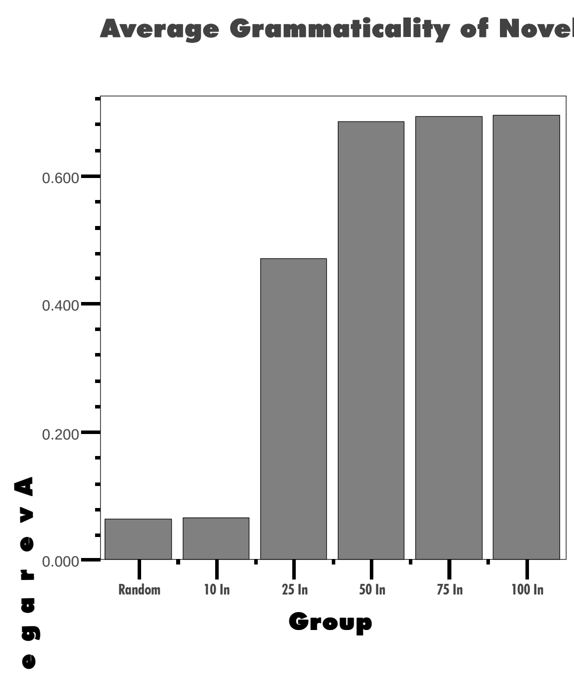


The grammaticality of output utterances was measured in two ways: 1) Absolute grammaticality - could the utterance have been produced by the target grammar? and 2) Average grammaticality - each utterance is given a grammaticality percentage based on the validity of its internal transitions. As with the accuracy of the transition matrix itself, we see a similar decrease in absolute grammaticality due to overgeneralization and the overall increase of novel utterances in the output. However, after an initial jump due to the "imitative" nature of the 10 group's output, the average grammaticality of output remains constant and impressively high – above 70% compared to 6.5% in the random group. This demonstrates that the average grammaticality of the output is seemingly independent of stages and changes occurring the grammar.

Absolute Grammaticality of Novel Utterances

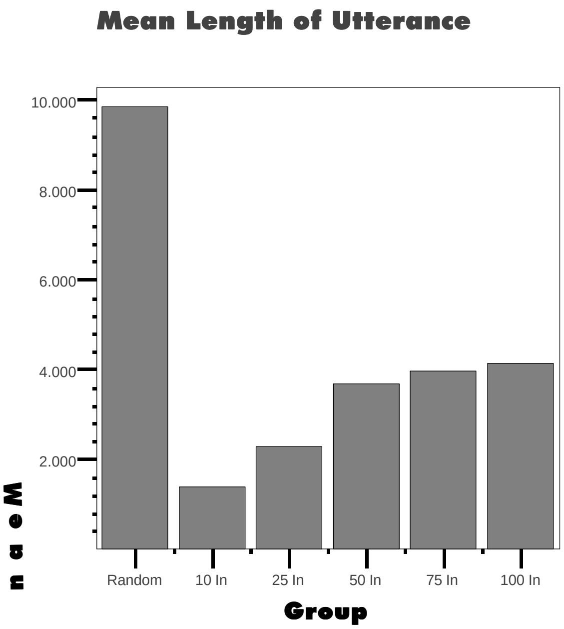


Average Grammaticality of Novel Utterances



If we look solely at the grammaticality measures of novel output utterances, there are some interesting differences. First, notice that although both measures drop greatly when imitative utterances are removed from the sample, even from 25 onward, the system performs dramatically better than random. In both cases, by 50 input utterances the differences in grammaticality have plateaued and become non-significant. We see for the first time a system performance that is not significantly better than random – the 10 group imitative stage – showing that the productivity and grammaticality exhibited by later stages is truly a function of input.

Mean Length Of Utterance



The final measure used to gauge performance was the classic MLU measurement (with respect to the acquired grammars' atomic elements). All the differences exhibited are strongly significant.

The asymptotic nature of the measurement with respect to the groups is strongly indicative of the system's ability to parse and correctly categorize words. The huge drop compared to the random group demonstrates that the structural properties of the input is genuinely represented in the acquired grammars, even if only in fact that produced utterances have definite starting and ending points; the random utterances were arbitrarily truncated at 11 words to prevent infinitely long utterances from being produced.

Other Observations

Examining the type of errors typically made by the acquired grammars is compelling. The constraints on grammaticality of utterances were fairly strict such that examples such as:

```
a hungry black dog slid
```

were considered ungrammatical. Although we can not make much of the fact that this turns out to be perfectly grammatical English, it is important to note that the errors being produced are themselves rule-governed and are not arbitrary and are somewhat innovative.

The categories that are typically produced correspond well to the gross word-level categories and moreover often reflect sub-categories that are reflective of the semantic properties that determined the original Markov chain despite the fact that the system has no direct access to semantic information.

3.5. Conclusions from the Model

Returning to the question originally proposed:

Can a statistically based, data-driven algorithm infer a grammar that:

a) is productive

b) is grammatical in comparison with the target grammar

c) has emergent categories

the answer is, “yes”. The acquired grammars are productive, have a high degree of grammaticality especially considering the algorithm’s simplistic one-dimensional nature, and produce categories that are very similar to ones used in creating the target grammar even though they are not explicitly represented a priori as part of knowledge given to the system.

4. Conclusion

What has been presented here is not enough to prove that children acquire language in the manner described; but then that was also not the intended goal. The point was to cast doubt on the “Poverty of Stimulus” argument that has long been held as a fundamental tenet of Linguistic Theory in light of growing evidence from other cognitive disciplines, that it may in fact be wrong. This has been accomplished – it has been demonstrated that more can be accomplished through statistically based learning based solely on input than has been previously accepted by linguists which raises the possibility that children, too, may employ more statistically based learning than we previously thought.

Appendix A: Matlab Code

```
function out = grammar(n);


load ('wordlist.mat');
load ('transmat.mat');

utterances = [];

for i=1:n

    type = round(rand*3+.5);

    if type == 1  % N
        utterances = [utterances; {round(rand*6+11.5)}];  %rand * (upper-lower+1) + (lower-.5)

    elseif type == 2  %NP
        start = round(rand * 2 + .5);
        finish = [12 13 14 15 16 17];
        utt = [start];
        l = length(utt);
        test = find(finish==utt(l));
        while isempty(test)==1
            possibles = find(transmat(utt(l),:));
            utt(l+1) = possibles(round(rand * (length(possibles)) + .5));
            l = length(utt);
            test = find(finish==utt(l));
        end
        utterances = [utterances; {utt}];

    elseif type == 3  %S
        start = round(rand * 2 + .5);
        utt = [start];
        l = length(utt);
        possibles = find(transmat(utt(l),:));
        while isempty(possibles)==0
            utt(l+1) = possibles(round(rand * (length(possibles)) + .5));
            l = length(utt);
            possibles = find(transmat(utt(l),:));
        end
        utterances = [utterances; {utt}];
    end
end

save('utterances.mat', 'utterances');
transutterances = transn2c(utterances, wordlist);

out = utterances;
```

```
function out = transn2c(utterances, wordlist);

out = [''];
brac = '|';

if iscellstr(utterances);
    temp = [];
    for i=1:length(utterances)
        temp = [temp; {str2num(utterances{i})}];
    end
    utterances = temp;
    brac = '';
end

if ~iscell(utterances)
    temp = [];
    [l w] = size(utterances);
    for i=1:l
        temp = [temp; {utterances(i,:)}];
    end
    utterances = temp;
end
```

```
for i=1:length(utterances)

    tobe = utterances{i};

    utt = [];
    [l w] = size(tobe);
    for j=1:w;
        utt = [utt, ' ', brac, wordlist{tobe(j)}, brac];
    end
    utt(1)=[];
    out = [out; {utt}];
end
```

```
function out = learn3(n, samples);

if nargin == 0;
    n = 50;
    samples = 1;
    output = 1;
else
    output = 0;
end

load('wordlist.mat');

if output == 0
    r1 = [];
    r2 = [];
    r3 = [];
    r4 = [];
    r5 = [];
    r6 = [];
    r7 = [];
    mlu = [];
end


for si = 1:samples

%***********************Learn stuff based on incoming data **********************

if output == 1
    load('utterances2.mat')
else
    utterances = grammar(n);
end

transutterances = transn2c(utterances, wordlist);
parsed = [];
learnedtransmat = [];
knownstufflist = [''];

for i=1:length(utterances)  %do this whole process for each utterance
    utt = utterances{i};
    l = length(utt);

    if isempty(strmatch(num2str(utt), knownstufflist, 'exact')) %if never seen full utterance before
add to memory
        knownstufflist = [knownstufflist; {num2str(utt)}];
        parsed(length(knownstufflist)) = 10;
    end

    if l == 2  %if utterance is only 2 words long
        p = strmatch(num2str(utt(1)), knownstufflist, 'exact');
        q = strmatch(num2str(utt(2)), knownstufflist, 'exact');
        if xor(isempty(p), isempty(q))
            if isempty(p)
                knownstufflist = [knownstufflist; {num2str(utt(1))}];
                p = strmatch(num2str(utt(1)), knownstufflist, 'exact');
            else
                knownstufflist = [knownstufflist; {num2str(utt(2))}];
```

```
            q = strmatch(num2str(utt(2)), knownstufflist, 'exact');
        end
    end
    if ~(isempty(p) & isempty(q))
        [sl sw] = size(learnedtransmat);
        if (sl<p | sw<q)
            con = 0;
        else
            con = learnedtransmat(p,q);
        end
        learnedtransmat(p,q) = con + (1-con)/2;
    end


elseif l > 2  %if utterance is longer than 2 words
    m = l-1;  % window size
    work = [];  % levels that are recognized
    while m >= 1  % for each window size
        for j = 1:(l-m) % starting in the first place, move window across utterace
            if l-m == 1  % if the window is so large that the utterance will only be divided into
two
                p = strmatch(num2str(utt(j:j+m-1)), knownstufflist, 'exact');
                q = strmatch(num2str(utt(j+m:l)), knownstufflist, 'exact');
                if xor(isempty(p), isempty(q))
                    if isempty(p)
                        knownstufflist = [knownstufflist; {num2str(utt(j:j+m-1))}];
                        p = strmatch(num2str(utt(j:j+m-1)), knownstufflist, 'exact');
                    else
                        knownstufflist = [knownstufflist; {num2str(utt(j+m:l))}];
                        q = strmatch(num2str(utt(j+m:l)), knownstufflist, 'exact');
                    end
                work = [work; {[p q]}];
                end
            elseif (j == 1) & (l-m > 1)  %if in first position, still only divided into two
(needed other wise index of 0)
                p = strmatch(num2str(utt(j:j+m-1)), knownstufflist, 'exact');
                q = strmatch(num2str(utt(j+m:l)), knownstufflist, 'exact');
                if xor(~isempty(p), ~isempty(q))
                    if isempty(p)
                        knownstufflist = [knownstufflist; {num2str(utt(j:j+m-1))}];
                        p = strmatch(num2str(utt(j:j+m-1)), knownstufflist, 'exact');
                    else
                        knownstufflist = [knownstufflist; {num2str(utt(j+m:l))}];
                        q = strmatch(num2str(utt(j+m:l)), knownstufflist, 'exact');
                    end
                work = [work; {[p q]}];
                end
            elseif (j > 1) & (l-m > 1)  %all other cases of window moving across space
                p = strmatch(num2str(utt(1:j-1)), knownstufflist, 'exact');
                q = strmatch(num2str(utt(j:j+m-1)), knownstufflist, 'exact');
                r = strmatch(num2str(utt(j+m:l)), knownstufflist, 'exact');
                if ~isempty(q)  %if windowed portion is recognized
                    if isempty(p)
                        knownstufflist = [knownstufflist; {num2str(utt(1:j-1))}];
                        p = strmatch(num2str(utt(1:j-1)), knownstufflist, 'exact');
                    end
                    if isempty(r)
                        knownstufflist = [knownstufflist; {num2str(utt(j+m:l))}];
                        r = strmatch(num2str(utt(j+m:l)), knownstufflist, 'exact');
                    end
                    work = [work; {[p q r]}];
                elseif isempty(q) & ~isempty(p) & ~isempty(r)   %if windowed portion is not
recognized, but the edges both are
                    knownstufflist = [knownstufflist; {num2str(utt(j:j+m-1))}];
                    q = strmatch(num2str(utt(j:j+m-1)), knownstufflist, 'exact');
                    work = [work; {[p q r]}];
                end
            end
        end

    worklength = length(work);
    for j=1:worklength
        if length(work{j}) == 2
            p = work{j}(1);
            q = work{j}(2);
```

```
                [sl sw] = size(learnedtransmat);
                if (sl<p | sw<q)
                    con = 0;
                else
                    con = learnedtransmat(p,q);
                end
                learnedtransmat(p,q) = con + (1-con)/2;
            elseif length(work{j}) == 3
                p = work{j}(1);
                q = work{j}(2);
                r = work{j}(3);
                [sl sw] = size(learnedtransmat);
                if (sl<p | sw<q)
                    con = 0;
                else
                    con = learnedtransmat(p,q);
                end
                learnedtransmat(p,q) = con + (1-con)/2;

                [sl sw] = size(learnedtransmat);
                if (sl<q | sw<r)
                    con = 0;
                else
                    con = learnedtransmat(q,r);
                end
                learnedtransmat(q,r) = con + (1-con)/2;
            end
        end

        m = m-1;

        end
        out = transn2c(work, transn2c(knownstufflist, wordlist));
    end
end

transknown = transn2c(knownstufflist, wordlist);

[ll lw] = size(learnedtransmat);
kl = length(knownstufflist);
learnedtransmat = [learnedtransmat; zeros(kl-ll, lw)];
learnedtransmat = [learnedtransmat, zeros(kl, kl-lw)];

l = length(learnedtransmat);

% find utterances that have not been parsed at all

if length(parsed) < length(knownstufflist)
    parsed = [parsed zeros(1, length(knownstufflist)-length(parsed))];
end
for i = 1:l
    totest = str2num(knownstufflist{i});
    if length(totest) > 1
        for j = 1:length(totest)
            p = strmatch(num2str(totest(j)), [knownstufflist(1:i-1); knownstufflist(i+1:l)], 'exact');
                if ~isempty(p)
                    parsed(i) = parsed(i) + 1;
                end
        end
    end
end

levelatoms = find(mod(parsed,10) == 0);

%calculates similarity based on what follows
similarityf = learnedtransmat * learnedtransmat';
%calculates similarity based on what precedes
similarityp = learnedtransmat' * learnedtransmat;
% combined similarity
similarity = (similarityp + similarityf)/2;


%normalize similarity
for i = 1:l
    similarity(i,i) = 0;
```

```
    similarityp(i,i) = 0;
    similarityf(i,i) = 0;
end

%similarity z-scores divided...
mp = mean(similarityp');
sigmap = std(similarityp');
normsimp = (similarityp - mp'*ones(1, length(mp)))./(sigmap'*ones(1, length(sigmap)));
mf = mean(similarityf');
sigmaf = std(similarityf');
normsimf = (similarityf - mf'*ones(1, length(mf)))./(sigmaf'*ones(1, length(sigmaf)));
normsim = (normsimp + normsimf)./2;

%build groups by pred and succ relationships
for i=1:l
    groups(1, i) = {levelatoms(find(learnedtransmat(i,levelatoms)))}; %group by common predecessor
    groups(2, i) = {levelatoms(find(learnedtransmat(levelatoms, i)))}; %grouped by common successor
end

%****************************Build New Transition Matrix**************************

prodwordlist = transknown(levelatoms);
prodmatrix = zeros(length(prodwordlist)+1);
[pl pw] = size(prodmatrix);


%possible starting words

constarts = [];
starters = intersect(find(sum(learnedtransmat) == 0), levelatoms);
for i = 1:length(starters)
    constarts(i) = find(levelatoms == starters(i));
end
prodmatrix(pl, constarts) = 1;

% enter possible followers straight from data
for i = 1:pl-1
    following = groups{1, levelatoms(i)};
    following = intersect(levelatoms, following);
    for j = 1:length(following);
        prodmatrix(i, find(levelatoms == following(j))) = 1;
    end
end

grouplist = [];
% make groups and add to new transition matrix
for i = 1:pl-1
    l2 = groups{1, levelatoms(i)};
    if ~isempty(l2) %only continue if there are a group of following words
        l3 = groups(1, l2);
        if ~isempty(l3{1}) %continue if there is one further level of following words
            l3b = [];
            for j = 1:length(l3)
                l3b = [l3b l3{j}];
            end
            if isempty(intersect(l2, l3b)) %build a group if the first and second groups do not share
elements
                l4 = groups(2, l3b);
                l4b = [];
                for j = 1:length(l4)
                    l4b = [l4b l4{j}];
                end
                if length(unique(l4b)) > 1  %don't bother making groups of size 1
                    l4 = [];
                    for j = 1:length(l4b)
                        l4(j) = find(levelatoms == l4b(j));
                    end
                    grouplist = [grouplist; {l4}];
                    prodmatrix(:, pw+1) = zeros(pl,1);
                    prodmatrix(i, pw+1) = length(l4);
                    [pl pw] = size(prodmatrix);
                end
            else %attempt to build groups in ambiguous cases, based on similarity
                l2 = unique(l2); % for first level down
                divisions = normsim(l2, l2);
```

```
                for j = 1:length(divisions)
                    l4 = l2(find(divisions(j,:) > 1.65));
                    if length(l4) > 0
                        l4b = [];
                        for k = 1:length(l4)
                            l4b(k) = find(levelatoms == l4(k));
                        end
                        grouplist = [grouplist; {[find(levelatoms == l2(j)) l4b]}];
                        prodmatrix(:, pw+1) = zeros(pl, 1);
                        prodmatrix(i, pw+1) = length(l4b)+1;
                        [pl pw] = size(prodmatrix);
                    end
                end
            end
        else %when end of utterance, make a group of first level
            l2b = [];
            for j = 1:length(l2)
                l2b(j) = find(levelatoms == l2(j));
            end
            if length(unique(l2b)) > 1 %don't bother making groups of size 1
                grouplist = [grouplist; {l2b}];
                prodmatrix(:, pw+1) = zeros(pl, 1);
                prodmatrix(i, pw+1) = length(l2b);
                [pl pw] = size(prodmatrix);
            end
        end
    end
end

%remove redundnant groups
cutlist = [];
for i = 1:length(grouplist);
    for j = i+1:length(grouplist)
        if length(unique(grouplist{i})) == length(unique(grouplist{j}))
            if unique(grouplist{i})==unique(grouplist{j})
                if length(grouplist{i}) >= length(grouplist{j})
                    prodmatrix(:, i+pl) = prodmatrix(:, i+pl) + prodmatrix(:, j+pl);
                    [pl pw] = size(prodmatrix);
                    cutlist = [cutlist; j];
                else
                    prodmatrix(:, j+pl) = prodmatrix(:, i+pl) + prodmatrix(:, j+pl);
                    [pl pw] = size(prodmatrix);
                    cutlist = [cutlist; i];
                end
            end
        end
    end
end
grouplist(cutlist) = [];
prodmatrix(:, cutlist+pl) = [];
transgrouplist = transn2c(grouplist, prodwordlist);

%Merge preds based on similarity
predsim = normsimf(levelatoms, levelatoms);
temp = zeros(size(prodmatrix));
for i = 1:length(predsim)
    threshold = 1;
    sims = find(predsim(i,:) > threshold);
    temp(i, :) = prodmatrix(i, :) + sum(prodmatrix(sims, :));
end
prodmatrix(1:pl-1, :) = temp(1:pl-1,:);

%**************** Merge Transmatrix for comparison *****************

%make correspondance list
correspond = zeros(length(prodwordlist), 2);
for i = 1:length(prodwordlist)
    temp = strmatch(prodwordlist(i), wordlist, 'exact');
    if ~isempty(temp)
        correspond(i,:) = [i temp];
    else
        correspond(i,:) = [i 0];
    end
end
```

```
%merge group info into the rest of the matrix
[pl pw] = size(prodmatrix);

prodmatrix2 = prodmatrix;

%[length(prodmatrix) length(prodmatrix2) length(grouplist)]
for i = 1:pl-1
    g = find(prodmatrix2(i, pl+2:pw) > 0);
    g = grouplist(g);
    h = [];
    for j = 1:length(g)
        h = [h g{j}];
    end
    prodmatrix2(i,unique(h)) = 1;
end
prodmatrix2(:,pl:pw) = [];
prodmatrix2(pl, :) = [];

%rearrange matrix
load('transmat.mat')
[tl tw] = size(transmat);
tempmatrix = zeros(tl, tw);
for i = 1:length(correspond)
    for j = 1:length(correspond)
        cx = correspond(i,2);
        cy = correspond(j,2);
%       [i j cx cy]
%       [size(tempmatrix) size(prodmatrix2)]
        if ~((correspond(i,2)==0)|(correspond(j,2)==0))
            tempmatrix(cx, cy) = prodmatrix2(i,j);
        end
    end
end

ind = find(tempmatrix > 0);
tempmatrix(ind) = 1;

indtransmat = find(transmat > 0);
rating1 = length(find((tempmatrix == transmat) > 0)) / 26^2;
if length(ind) > 0
    rating2 = length(intersect(ind, indtransmat))/length(ind);
else
    rating2 = 0;
end

%************************* Make New Stuff ****************************

startrow = length(prodwordlist)+1;
prodmatrix(:,startrow) = zeros(startrow,1);
%prodwordlist
%transgrouplist

newutterances = [];
for i=1:n
    start = find(prodmatrix(startrow, 1:startrow));
    starti = round(rand * length(start) + .5);
    start = start(starti);
    utt = [start];
    l = length(utt);
    possibles = find(prodmatrix(utt(l),:));
    while ~isempty(possibles)
        nexti = round(rand * (length(possibles)) + .5);
        if possibles(nexti) < startrow
            utt(l+1) = possibles(nexti);
            l = length(utt);
            possibles = find(prodmatrix(utt(l),:));
        elseif possibles(nexti) > startrow
            nextset = grouplist{(possibles(nexti) - startrow)};
            nextset = nextset(find(nextset <= length(prodwordlist)));
            utt(l+1) = nextset(round(rand * (length(nextset)) + .5));
            l = length(utt);
            possibles = find(prodmatrix(utt(l),:));
        elseif possibles(nexti) == startrow
            disp('uh-oh!');
        end
```

```
    end
    newutterances = [newutterances; {num2str(utt)}];
end

newnessind = zeros(length(newutterances),1);
diffnewutt = [];

transnewutt = transn2c(newutterances, prodwordlist);

for i=1:length(newutterances)
    utt = transnewutt(i);

    if isempty(strmatch(utt, transknown, 'exact'))
        newnessind(i) = 1;
        diffnewutt = [diffnewutt; utt];
    end
end

rating3 = length(diffnewutt)/length(newutterances);  % percentage of novel utterances

% ***************** Are utterances grammatical? *******************

gramrating = [];
for i = 1:n
    utt2chk = str2num(newutterances{i});
    ul = length(utt2chk);
    transchk = [];
    if ul > 1
        for j = 1:ul-1
            x = correspond(utt2chk(j),2);
            y = correspond(utt2chk(j+1),2);
            if ~((x == 0)|(y == 0))
%           [x y size(transmat)];
                if transmat(x, y) == 1
                    transchk = [transchk 1];
                else
                    transchk = [transchk 0];
                end
            else
                transchk = [transchk 0];
            end
        transchk = sum(transchk) / length(transchk);
        end
    else
        transchk = [1];
    end
    gramrating = [gramrating; transchk];
end

rating4 = length(find(gramrating == 1))/length(gramrating); %percentage of grammatical utterances
rating5 = mean(gramrating); %average grammaticality
rating6 = length(intersect(find(gramrating == 1), find(newnessind == 1))) / length(newnessind); %
percent of novel utterances that are grammatical
if find(newnessind) > 0
    rating7 = mean(gramrating(find(newnessind))); % average grammaticality of novel utterances
else
    rating7 = 0;
end


% calculate mean length of utterance
for i = 1:length(newutterances)
    lutt(i) = length(str2num(newutterances{i}));
end
meanlengthutt = sum(lutt)/length(lutt);


if output == 1
    disp(['percentage of matching entries: ', num2str(rating1)]);
    disp(['percentage of 1s that are correct: ', num2str(rating2)]);
    disp(['percentage of utterances that are novel: ' num2str(rating3)]);
    disp(['percentage of grammatical utterances: ' num2str(rating4)]);
    disp(['average grammaticality: ' num2str(rating5)]);
    disp(['percentage of grammatical, novel utterances: ' num2str(rating6)]);
    disp(['average grammaticality of novel utterances: ' num2str(rating7)]);
```

```
    disp(['mean length of utterance: ' num2str(meanlengthutt)]);
end

if output == 0
    r1 = [r1; rating1];
    r2 = [r2; rating2];
    r3 = [r3; rating3];
    r4 = [r4; rating4];
    r5 = [r5; rating5];
    r6 = [r6; rating6];
    r7 = [r7; rating7];
    mlu = [mlu; meanlengthutt];

    eval(['save(''data\nonrandom\run'   int2str(n)   int2str(si)   'data.mat'',   ''utterances'',
''transutterances'', ''prodwordlist'', ''transgrouplist'', ''knownstufflist'', ''learnedtransmat'',
''normsimp'', ''normsimf'', ''normsim'', ''prodmatrix'', ''tempmatrix'', ''rating1'', ''rating2'',
''rating3'',  ''rating4'',  ''rating5'',  ''rating6'',  ''rating7'',  ''mlu'',  ''newutterances'',
''transnewutt'', ''gramrating'', ''diffnewutt'');']);
end

si

end

if output == 0
    rdata = [r1 r2 r3 r4 r5 r6 r7 mlu];
    eval(['save(''data\nonrandom\r' int2str(n) 'data.txt'', ''rdata'', ''-ASCII'', ''-DOUBLE'', ''-
TABS'');']);
end

out = char(transgrouplist);
```

Appendix B: T-Tests

T-Test - 10 Input Utterances Group & 25 Input Utterances Group

Group Statistics

	Group	N	Mean	Std. Deviation	Std. Error Mean
Matching - All	1	100	.86028	2.3636E-03	2.364E-04
	2	100	.85262	1.5553E-02	1.555E-03
Matching - 1s	1	100	.43874	.44811	4.481E-02
	2	100	.53642	.23415	2.342E-02
Novel Utterances	1	100	9.700E-02	.19718	1.972E-02
	2	100	.40960	.23744	2.374E-02
Grammatical Utterances	1	100	.82600	.21678	2.168E-02
	2	100	.65240	.19386	1.939E-02
Average Grammaticality	1	100	.85372	.19053	1.905E-02
	2	100	.74814	.15585	1.559E-02
Grammatical Novel	1	100	8.000E-03	3.6735E-02	3.674E-03
	2	100	.13360	.12980	1.298E-02
Average Gramm. Novel	1	100	6.605E-02	.17829	1.783E-02
	2	100	.47073	.28417	2.842E-02
Mean Length of Utterance	1	100	1.40100	.52078	5.208E-02
	2	100	2.30560	.70913	7.091E-02

Independent Samples Test

		Levene's Test for Equality of Variances		t-test for Equality of Means						
									95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
Matching - All	Equal variances assumed	143.833	.000	4.871	198	.000	7.6627E-03	1.5731E-03	4.560E-03	1.076E-02
	Equal variances not assumed			4.871	103.571	.000	7.6627E-03	1.5731E-03	4.543E-03	1.078E-02
Matching - 1s	Equal variances assumed	114.038	.000	-1.932	198	.055	-9.768E-02	5.0560E-02	-.19738	2.029E-03
	Equal variances not assumed			-1.932	149.311	.055	-9.768E-02	5.0560E-02	-.19758	2.229E-03
Novel Utterances	Equal variances assumed	5.772	.017	-10.128	198	.000	-.31260	3.0863E-02	-.37346	-.25174
	Equal variances not assumed			-10.128	191.538	.000	-.31260	3.0863E-02	-.37348	-.25172
Grammatical Utterances	Equal variances assumed	1.860	.174	5.969	198	.000	.17360	2.9082E-02	.11625	.23095
	Equal variances not assumed			5.969	195.578	.000	.17360	2.9082E-02	.11625	.23095
Average Grammaticality	Equal variances assumed	4.943	.027	4.289	198	.000	.10558	2.4616E-02	5.703E-02	.15412
	Equal variances not assumed			4.289	190.513	.000	.10558	2.4616E-02	5.702E-02	.15413
Grammatical Novel	Equal variances assumed	146.642	.000	-9.311	198	.000	-.12560	1.3490E-02	-.15220	-9.90E-02
	Equal variances not assumed			-9.311	114.759	.000	-.12560	1.3490E-02	-.15232	-9.89E-02
Average Gramm. Novel	Equal variances assumed	36.858	.000	-12.063	198	.000	-.40469	3.3547E-02	-.47084	-.33853
	Equal variances not assumed			-12.063	166.484	.000	-.40469	3.3547E-02	-.47092	-.33845
Mean Length of Utterance	Equal variances assumed	10.130	.002	-10.282	198	.000	-.90460	8.7982E-02	-1.07810	-.73110
	Equal variances not assumed			-10.282	181.724	.000	-.90460	8.7982E-02	-1.07820	-.73100

T-Test - 25 Input Utterances Group & 50 Input Utterances Group

Group Statistics

	Group	N	Mean	Std. Deviation	Std. Error Mean
Matching - All	2	100	.85262	1.5553E-02	1.555E-03
	3	100	.79578	3.4995E-02	3.499E-03
Matching - 1s	2	100	.53642	.23415	2.342E-02
	3	100	.36193	5.2548E-02	5.255E-03
Novel Utterances	2	100	.40960	.23744	2.374E-02
	3	100	.81380	.13829	1.383E-02
Grammatical Utterances	2	100	.65240	.19386	1.939E-02
	3	100	.48420	.14902	1.490E-02
Average Grammaticality	2	100	.74814	.15585	1.559E-02
	3	100	.74130	.10707	1.071E-02
Grammatical Novel	2	100	.13360	.12980	1.298E-02
	3	100	.29880	.11064	1.106E-02
Average Gramm. Novel	2	100	.47073	.28417	2.842E-02
	3	100	.68591	.10942	1.094E-02
Mean Length of Utterance	2	100	2.30560	.70913	7.091E-02
	3	100	3.68740	.75819	7.582E-02

Independent Samples Test

		Levene's Test for Equality of Variances		t-test for Equality of Means						
									95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
Matching - All	Equal variances assumed	41.163	.000	14.841	198	.000	5.6834E-02	3.8295E-03	4.928E-02	6.439E-02
	Equal variances not assumed			14.841	136.641	.000	5.6834E-02	3.8295E-03	4.926E-02	6.441E-02
Matching - 1s	Equal variances assumed	100.733	.000	7.271	198	.000	.17449	2.3998E-02	.12717	.22181
	Equal variances not assumed			7.271	108.947	.000	.17449	2.3998E-02	.12693	.22205
Novel Utterances	Equal variances assumed	24.779	.000	-14.710	198	.000	-.40420	2.7477E-02	-.45839	-.35001
	Equal variances not assumed			-14.710	159.237	.000	-.40420	2.7477E-02	-.45847	-.34993
Grammatical Utterances	Equal variances assumed	4.739	.031	6.879	198	.000	.16820	2.4452E-02	.11998	.21642
	Equal variances not assumed			6.879	185.721	.000	.16820	2.4452E-02	.11996	.21644
Average Grammaticality	Equal variances assumed	10.942	.001	.362	198	.718	6.8445E-03	1.8909E-02	-3.04E-02	4.413E-02
	Equal variances not assumed			.362	175.427	.718	6.8445E-03	1.8909E-02	-3.05E-02	4.416E-02
Grammatical Novel	Equal variances assumed	3.966	.048	-9.686	198	.000	-.16520	1.7055E-02	-.19883	-.13157
	Equal variances not assumed			-9.686	193.156	.000	-.16520	1.7055E-02	-.19884	-.13156
Average Gramm. Novel	Equal variances assumed	79.512	.000	-7.066	198	.000	-.21518	3.0451E-02	-.27523	-.15513
	Equal variances not assumed			-7.066	127.724	.000	-.21518	3.0451E-02	-.27543	-.15492
Mean Length of Utterance	Equal variances assumed	.028	.867	-13.310	198	.000	-1.38180	.10381	-1.58652	-1.17708
	Equal variances not assumed			-13.310	197.120	.000	-1.38180	.10381	-1.58653	-1.17707

T-Test - 50 Input Utterances Group & 75 Input Utterances Group

Group Statistics

	Group	N	Mean	Std. Deviation	Std. Error Mean
Matching - All	3	100	.79578	3.4995E-02	3.499E-03
	4	100	.74189	4.5108E-02	4.511E-03
Matching - 1s	3	100	.36193	5.2548E-02	5.255E-03
	4	100	.33555	3.9509E-02	3.951E-03
Novel Utterances	3	100	.81380	.13829	1.383E-02
	4	100	.88507	7.5205E-02	7.520E-03
Grammatical Utterances	3	100	.48420	.14902	1.490E-02
	4	100	.44040	.11027	1.103E-02
Average Grammaticality	3	100	.74130	.10707	1.071E-02
	4	100	.72689	8.7531E-02	8.753E-03
Grammatical Novel	3	100	.29880	.11064	1.106E-02
	4	100	.32547	8.6764E-02	8.676E-03
Average Gramm. Novel	3	100	.68591	.10942	1.094E-02
	4	100	.69295	9.1474E-02	9.147E-03
Mean Length of Utterance	3	100	3.68740	.75819	7.582E-02
	4	100	3.97787	.52531	5.253E-02

Independent Samples Test

		Levene's Test for Equality of Variances		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference Lower	95% Confidence Interval of the Difference Upper
Matching - All	Equal variances assumed	4.719	.031	9.439	198	.000	5.3891E-02	5.7091E-03	4.263E-02	6.515E-02
	Equal variances not assumed			9.439	186.480	.000	5.3891E-02	5.7091E-03	4.263E-02	6.515E-02
Matching - 1s	Equal variances assumed	6.842	.010	4.013	198	.000	2.6382E-02	6.5744E-03	1.342E-02	3.935E-02
	Equal variances not assumed			4.013	183.823	.000	2.6382E-02	6.5744E-03	1.341E-02	3.935E-02
Novel Utterances	Equal variances assumed	36.440	.000	-4.527	198	.000	-7.127E-02	1.5742E-02	-.10231	-4.02E-02
	Equal variances not assumed			-4.527	152.845	.000	-7.127E-02	1.5742E-02	-.10237	-4.02E-02
Grammatical Utterances	Equal variances assumed	11.018	.001	2.363	198	.019	4.3800E-02	1.8538E-02	7.242E-03	8.036E-02
	Equal variances not assumed			2.363	182.403	.019	4.3800E-02	1.8538E-02	7.223E-03	8.038E-02
Average Grammaticality	Equal variances assumed	4.222	.041	1.042	198	.299	1.4406E-02	1.3830E-02	-1.29E-02	4.168E-02
	Equal variances not assumed			1.042	190.469	.299	1.4406E-02	1.3830E-02	-1.29E-02	4.169E-02
Grammatical Novel	Equal variances assumed	6.016	.015	-1.897	198	.059	-2.667E-02	1.4060E-02	-5.44E-02	1.060E-03
	Equal variances not assumed			-1.897	187.352	.059	-2.667E-02	1.4060E-02	-5.44E-02	1.070E-03
Average Gramm. Novel	Equal variances assumed	4.276	.040	-.494	198	.622	-7.039E-03	1.4262E-02	-3.52E-02	2.109E-02
	Equal variances not assumed			-.494	191.970	.622	-7.039E-03	1.4262E-02	-3.52E-02	2.109E-02
Mean Length of Utterance	Equal variances assumed	4.534	.034	-3.149	198	.002	-.29047	9.2240E-02	-.47236	-.10857
	Equal variances not assumed			-3.149	176.247	.002	-.29047	9.2240E-02	-.47250	-.10843

T-Test - 75 Input Utterances Group & 100 Input Utterances Group

Group Statistics

	Group	N	Mean	Std. Deviation	Std. Error Mean
Matching - All	4	100	.74189	4.5108E-02	4.511E-03
	5	100	.70652	4.3981E-02	4.398E-03
Matching - 1s	4	100	.33555	3.9509E-02	3.951E-03
	5	100	.31732	3.4917E-02	3.492E-03
Novel Utterances	4	100	.88507	7.5205E-02	7.520E-03
	5	100	.90220	4.7388E-02	4.739E-03
Grammatical Utterances	4	100	.44040	.11027	1.103E-02
	5	100	.41300	8.3985E-02	8.399E-03
Average Grammaticality	4	100	.72689	8.7531E-02	8.753E-03
	5	100	.72522	5.9244E-02	5.924E-03
Grammatical Novel	4	100	.32547	8.6764E-02	8.676E-03
	5	100	.31520	6.6583E-02	6.658E-03
Average Gramm. Novel	4	100	.69295	9.1474E-02	9.147E-03
	5	100	.69607	5.9103E-02	5.910E-03
Mean Length of Utterance	4	100	3.97787	.52531	5.253E-02
	5	100	4.16350	.42845	4.284E-02

Independent Samples Test

		Levene's Test for Equality of Variances		t-test for Equality of Means						
									95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
Matching - All	Equal variances assumed	.003	.960	5.614	198	.000	3.5370E-02	6.3000E-03	2.295E-02	4.779E-02
	Equal variances not assumed			5.614	197.873	.000	3.5370E-02	6.3000E-03	2.295E-02	4.779E-02
Matching - 1s	Equal variances assumed	.864	.354	3.456	198	.001	1.8224E-02	5.2727E-03	7.826E-03	2.862E-02
	Equal variances not assumed			3.456	195.053	.001	1.8224E-02	5.2727E-03	7.825E-03	2.862E-02
Novel Utterances	Equal variances assumed	7.747	.006	-1.927	198	.055	-1.713E-02	8.8890E-03	-3.47E-02	3.959E-04
	Equal variances not assumed			-1.927	166.910	.056	-1.713E-02	8.8890E-03	-3.47E-02	4.160E-04
Grammatical Utterances	Equal variances assumed	8.412	.004	1.977	198	.049	2.7400E-02	1.3861E-02	6.618E-05	5.473E-02
	Equal variances not assumed			1.977	184.941	.050	2.7400E-02	1.3861E-02	5.430E-05	5.475E-02
Average Grammaticality	Equal variances assumed	12.362	.001	.158	198	.874	1.6748E-03	1.0570E-02	-1.92E-02	2.252E-02
	Equal variances not assumed			.158	173.972	.874	1.6748E-03	1.0570E-02	-1.92E-02	2.254E-02
Grammatical Novel	Equal variances assumed	8.363	.004	.939	198	.349	1.0267E-02	1.0937E-02	-1.13E-02	3.183E-02
	Equal variances not assumed			.939	185.577	.349	1.0267E-02	1.0937E-02	-1.13E-02	3.184E-02
Average Gramm. Novel	Equal variances assumed	13.438	.000	-.286	198	.775	-3.119E-03	1.0891E-02	-2.46E-02	1.836E-02
	Equal variances not assumed			-.286	169.391	.775	-3.119E-03	1.0891E-02	-2.46E-02	1.838E-02
Mean Length of Utterance	Equal variances assumed	4.068	.045	-2.738	198	.007	-.18563	6.7788E-02	-.31931	-5.20E-02
	Equal variances not assumed			-2.738	190.307	.007	-.18563	6.7788E-02	-.31935	-5.19E-02

T-Test - Random Group & 10 Input Utterances Group

Group Statistics

	Group	N	Mean	Std. Deviation	Std. Error Mean
Matching - All	0	100	.75867	1.3169E-02	1.317E-03
	1	100	.86028	2.3636E-03	2.364E-04
Matching - 1s	0	100	.14043	3.1867E-02	3.187E-03
	1	100	.43874	.44811	4.481E-02
Grammatical Utterances	0	100	4.580E-02	7.3926E-02	7.393E-03
	1	100	.82600	.21678	2.168E-02
Average Grammaticality	0	100	6.504E-02	9.8101E-02	9.810E-03
	1	100	.85372	.19053	1.905E-02
Grammatical Novel	0	100	4.580E-02	7.3926E-02	7.393E-03
	1	100	8.000E-03	3.6735E-02	3.674E-03
Average Gramm. Novel	0	100	6.504E-02	9.8101E-02	9.810E-03
	1	100	6.605E-02	.17829	1.783E-02
Mean Length of Utterance	0	100	9.88208	1.76811	.17681
	1	100	1.40100	.52078	5.208E-02

Independent Samples Test

		Levene's Test for Equality of Variances		t-test for Equality of Means						
									95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
Matching - All	Equal variances assumed	124.781	.000	-75.945	198	.000	-.10161	1.3380E-03	-.10425	-9.90E-02
	Equal variances not assumed			-75.945	105.372	.000	-.10161	1.3380E-03	-.10427	-9.90E-02
Matching - 1s	Equal variances assumed	568.720	.000	-6.640	198	.000	-.29831	4.4924E-02	-.38691	-.20972
	Equal variances not assumed			-6.640	100.001	.000	-.29831	4.4924E-02	-.38744	-.20919
Grammatical Utterances	Equal variances assumed	85.880	.000	-34.064	198	.000	-.78020	2.2904E-02	-.82537	-.73503
	Equal variances not assumed			-34.064	121.719	.000	-.78020	2.2904E-02	-.82554	-.73486
Average Grammaticality	Equal variances assumed	39.356	.000	-36.801	198	.000	-.78868	2.1431E-02	-.83094	-.74641
	Equal variances not assumed			-36.801	148.043	.000	-.78868	2.1431E-02	-.83102	-.74633
Grammatical Novel	Equal variances assumed	47.717	.000	4.579	198	.000	3.7800E-02	8.2550E-03	2.152E-02	5.408E-02
	Equal variances not assumed			4.579	145.083	.000	3.7800E-02	8.2550E-03	2.148E-02	5.412E-02
Average Gramm. Novel	Equal variances assumed	5.083	.025	-.049	198	.961	-1.004E-03	2.0350E-02	-4.11E-02	3.913E-02
	Equal variances not assumed			-.049	153.912	.961	-1.004E-03	2.0350E-02	-4.12E-02	3.920E-02
Mean Length of Utterance	Equal variances assumed	68.779	.000	46.012	198	.000	8.48108	.18432	8.11759	8.84456
	Equal variances not assumed			46.012	116.049	.000	8.48108	.18432	8.11601	8.84615

T-Test - Random Group & 25 Input Utterances Group

Group Statistics

	Group	N	Mean	Std. Deviation	Std. Error Mean
Matching - All	0	100	.75867	1.3169E-02	1.317E-03
	2	100	.85262	1.5553E-02	1.555E-03
Matching - 1s	0	100	.14043	3.1867E-02	3.187E-03
	2	100	.53642	.23415	2.342E-02
Grammatical Utterances	0	100	4.580E-02	7.3926E-02	7.393E-03
	2	100	.65240	.19386	1.939E-02
Average Grammaticality	0	100	6.504E-02	9.8101E-02	9.810E-03
	2	100	.74814	.15585	1.559E-02
Grammatical Novel	0	100	4.580E-02	7.3926E-02	7.393E-03
	2	100	.13360	.12980	1.298E-02
Average Gramm. Novel	0	100	6.504E-02	9.8101E-02	9.810E-03
	2	100	.47073	.28417	2.842E-02
Mean Length of Utterance	0	100	9.88208	1.76811	.17681
	2	100	2.30560	.70913	7.091E-02

Independent Samples Test

		Levene's Test for Equality of Variances		t-test for Equality of Means						
									95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
Matching - All	Equal variances assumed	3.013	.084	-46.100	198	.000	-9.395E-02	2.0379E-03	-9.80E-02	-8.99E-02
	Equal variances not assumed			-46.100	192.762	.000	-9.395E-02	2.0379E-03	-9.80E-02	-8.99E-02
Matching - 1s	Equal variances assumed	126.124	.000	-16.757	198	.000	-.39599	2.3631E-02	-.44259	-.34939
	Equal variances not assumed			-16.757	102.666	.000	-.39599	2.3631E-02	-.44286	-.34912
Grammatical Utterances	Equal variances assumed	62.697	.000	-29.237	198	.000	-.60660	2.0748E-02	-.64752	-.56568
	Equal variances not assumed			-29.237	127.196	.000	-.60660	2.0748E-02	-.64766	-.56554
Average Grammaticality	Equal variances assumed	16.882	.000	-37.093	198	.000	-.68310	1.8416E-02	-.71942	-.64678
	Equal variances not assumed			-37.093	166.804	.000	-.68310	1.8416E-02	-.71946	-.64674
Grammatical Novel	Equal variances assumed	39.209	.000	-5.878	198	.000	-8.780E-02	1.4937E-02	-.11726	-5.83E-02
	Equal variances not assumed			-5.878	157.113	.000	-8.780E-02	1.4937E-02	-.11730	-5.83E-02
Average Gramm. Novel	Equal variances assumed	93.375	.000	-13.495	198	.000	-.40569	3.0063E-02	-.46497	-.34641
	Equal variances not assumed			-13.495	122.267	.000	-.40569	3.0063E-02	-.46520	-.34618
Mean Length of Utterance	Equal variances assumed	44.093	.000	39.771	198	.000	7.57648	.19050	7.20081	7.95215
	Equal variances not assumed			39.771	130.046	.000	7.57648	.19050	7.19960	7.95336

T-Test - Random Group & 50 Input Utterances Group

Group Statistics

	Group	N	Mean	Std. Deviation	Std. Error Mean
Matching - All	0	100	.75867	1.3169E-02	1.317E-03
	3	100	.79578	3.4995E-02	3.499E-03
Matching - 1s	0	100	.14043	3.1867E-02	3.187E-03
	3	100	.36193	5.2548E-02	5.255E-03
Grammatical Utterances	0	100	4.580E-02	7.3926E-02	7.393E-03
	3	100	.48420	.14902	1.490E-02
Average Grammaticality	0	100	6.504E-02	9.8101E-02	9.810E-03
	3	100	.74130	.10707	1.071E-02
Grammatical Novel	0	100	4.580E-02	7.3926E-02	7.393E-03
	3	100	.29880	.11064	1.106E-02
Average Gramm. Novel	0	100	6.504E-02	9.8101E-02	9.810E-03
	3	100	.68591	.10942	1.094E-02
Mean Length of Utterance	0	100	9.88208	1.76811	.17681
	3	100	3.68740	.75819	7.582E-02

Independent Samples Test

		Levene's Test for Equality of Variances		t-test for Equality of Means						
									95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
Matching - All	Equal variances assumed	55.420	.000	-9.926	198	.000	-3.712E-02	3.7391E-03	-4.45E-02	-2.97E-02
	Equal variances not assumed			-9.926	126.489	.000	-3.712E-02	3.7391E-03	-4.45E-02	-2.97E-02
Matching - 1s	Equal variances assumed	14.046	.000	-36.042	198	.000	-.22150	6.1456E-03	-.23362	-.20938
	Equal variances not assumed			-36.042	163.144	.000	-.22150	6.1456E-03	-.23364	-.20937
Grammatical Utterances	Equal variances assumed	51.977	.000	-26.354	198	.000	-.43840	1.6635E-02	-.47121	-.40559
	Equal variances not assumed			-26.354	144.941	.000	-.43840	1.6635E-02	-.47128	-.40552
Average Grammaticality	Equal variances assumed	.884	.348	-46.568	198	.000	-.67625	1.4522E-02	-.70489	-.64762
	Equal variances not assumed			-46.568	196.503	.000	-.67625	1.4522E-02	-.70489	-.64762
Grammatical Novel	Equal variances assumed	18.482	.000	-19.013	198	.000	-.25300	1.3306E-02	-.27924	-.22676
	Equal variances not assumed			-19.013	172.706	.000	-.25300	1.3306E-02	-.27926	-.22674
Average Gramm. Novel	Equal variances assumed	1.538	.216	-42.248	198	.000	-.62087	1.4696E-02	-.64985	-.59189
	Equal variances not assumed			-42.248	195.686	.000	-.62087	1.4696E-02	-.64985	-.59188
Mean Length of Utterance	Equal variances assumed	42.848	.000	32.200	198	.000	6.19468	.19238	5.81530	6.57406
	Equal variances not assumed			32.200	134.218	.000	6.19468	.19238	5.81419	6.57517

T-Test - Random Group & 75 Input Utterances Group

Group Statistics

	Group	N	Mean	Std. Deviation	Std. Error Mean
Matching - All	0	100	.75867	1.3169E-02	1.317E-03
	4	100	.74189	4.5108E-02	4.511E-03
Matching - 1s	0	100	.14043	3.1867E-02	3.187E-03
	4	100	.33555	3.9509E-02	3.951E-03
Grammatical Utterances	0	100	4.580E-02	7.3926E-02	7.393E-03
	4	100	.44040	.11027	1.103E-02
Average Grammaticality	0	100	6.504E-02	9.8101E-02	9.810E-03
	4	100	.72689	8.7531E-02	8.753E-03
Grammatical Novel	0	100	4.580E-02	7.3926E-02	7.393E-03
	4	100	.32547	8.6764E-02	8.676E-03
Average Gramm. Novel	0	100	6.504E-02	9.8101E-02	9.810E-03
	4	100	.69295	9.1474E-02	9.147E-03
Mean Length of Utterance	0	100	9.88208	1.76811	.17681
	4	100	3.97787	.52531	5.253E-02

Independent Samples Test

		Levene's Test for Equality of Variances		t-test for Equality of Means						
									95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
Matching - All	Equal variances assumed	71.813	.000	3.570	198	.000	1.6775E-02	4.6991E-03	7.508E-03	2.604E-02
	Equal variances not assumed			3.570	115.755	.001	1.6775E-02	4.6991E-03	7.468E-03	2.608E-02
Matching - 1s	Equal variances assumed	.930	.336	-38.440	198	.000	-.19512	5.0759E-03	-.20513	-.18511
	Equal variances not assumed			-38.440	189.507	.000	-.19512	5.0759E-03	-.20513	-.18511
Grammatical Utterances	Equal variances assumed	18.255	.000	-29.724	198	.000	-.39460	1.3275E-02	-.42078	-.36842
	Equal variances not assumed			-29.724	173.038	.000	-.39460	1.3275E-02	-.42080	-.36840
Average Grammaticality	Equal variances assumed	1.236	.268	-50.341	198	.000	-.66185	1.3147E-02	-.68778	-.63592
	Equal variances not assumed			-50.341	195.481	.000	-.66185	1.3147E-02	-.68778	-.63592
Grammatical Novel	Equal variances assumed	4.230	.041	-24.535	198	.000	-.27967	1.1399E-02	-.30215	-.25719
	Equal variances not assumed			-24.535	193.131	.000	-.27967	1.1399E-02	-.30215	-.25718
Average Gramm. Novel	Equal variances assumed	.706	.402	-46.813	198	.000	-.62791	1.3413E-02	-.65436	-.60145
	Equal variances not assumed			-46.813	197.039	.000	-.62791	1.3413E-02	-.65436	-.60145
Mean Length of Utterance	Equal variances assumed	64.930	.000	32.010	198	.000	5.90421	.18445	5.54047	6.26795
	Equal variances not assumed			32.010	116.343	.000	5.90421	.18445	5.53890	6.26953

T-Test - Random Group & 100 Input Utterances Group

Group Statistics

	Group	N	Mean	Std. Deviation	Std. Error Mean
Matching - All	0	100	.75867	1.3169E-02	1.317E-03
	5	100	.70652	4.3981E-02	4.398E-03
Matching - 1s	0	100	.14043	3.1867E-02	3.187E-03
	5	100	.31732	3.4917E-02	3.492E-03
Grammatical Utterances	0	100	4.580E-02	7.3926E-02	7.393E-03
	5	100	.41300	8.3985E-02	8.399E-03
Average Grammaticality	0	100	6.504E-02	9.8101E-02	9.810E-03
	5	100	.72522	5.9244E-02	5.924E-03
Grammatical Novel	0	100	4.580E-02	7.3926E-02	7.393E-03
	5	100	.31520	6.6583E-02	6.658E-03
Average Gramm. Novel	0	100	6.504E-02	9.8101E-02	9.810E-03
	5	100	.69607	5.9103E-02	5.910E-03
Mean Length of Utterance	0	100	9.88208	1.76811	.17681
	5	100	4.16350	.42845	4.284E-02

Independent Samples Test

		Levene's Test for Equality of Variances		t-test for Equality of Means						
									95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
Matching - All	Equal variances assumed	84.343	.000	11.358	198	.000	5.2145E-02	4.5910E-03	4.309E-02	6.120E-02
	Equal variances not assumed			11.358	116.611	.000	5.2145E-02	4.5910E-03	4.305E-02	6.124E-02
Matching - 1s	Equal variances assumed	.004	.951	-37.420	198	.000	-.17689	4.7273E-03	-.18622	-.16757
	Equal variances not assumed			-37.420	196.369	.000	-.17689	4.7273E-03	-.18622	-.16757
Grammatical Utterances	Equal variances assumed	2.163	.143	-32.819	198	.000	-.36720	1.1189E-02	-.38926	-.34514
	Equal variances not assumed			-32.819	194.862	.000	-.36720	1.1189E-02	-.38927	-.34513
Average Grammaticality	Equal variances assumed	20.742	.000	-57.606	198	.000	-.66017	1.1460E-02	-.68277	-.63757
	Equal variances not assumed			-57.606	162.734	.000	-.66017	1.1460E-02	-.68280	-.63754
Grammatical Novel	Equal variances assumed	.542	.462	-27.078	198	.000	-.26940	9.9490E-03	-.28902	-.24978
	Equal variances not assumed			-27.078	195.872	.000	-.26940	9.9490E-03	-.28902	-.24978
Average Gramm. Novel	Equal variances assumed	20.916	.000	-55.097	198	.000	-.63102	1.1453E-02	-.65361	-.60844
	Equal variances not assumed			-55.097	162.501	.000	-.63102	1.1453E-02	-.65364	-.60841
Mean Length of Utterance	Equal variances assumed	78.523	.000	31.433	198	.000	5.71858	.18193	5.35981	6.07735
	Equal variances not assumed			31.433	110.586	.000	5.71858	.18193	5.35806	6.07910

Appendix C: Sample Data - Poor, Average, and Good

Random Data - Run 26 - Output Utterances

fell expensive ran slept dirty
dog crashed smelled
red dirty
died hungry black expensive the
black expensive ran slept dirty
clean died hungry white the
rat rat slept dog crashed black expensive slept dog crashed black
fell expensive slept dirty
white car deep black expensive slept dog crashed car empty died
the
black expensive dog the
hungry box window broke rat rat rat rat slept dog the
dog crashed smelled
rat slept dog crashed deep slid car car a fell a
rat slept dirty
a fell expensive ran slept hungry opened red dirty
dog the
slid box window broke expensive dog crashed deep slid car empty
white cat opened red dog the
slept hungry black expensive the
smelled
slept dog the
the
crashed smelled
ran red clean died hungry white the
clean crashed car dog the
rat rat rat slept dog crashed rat rat rat rat rat
empty died hungry white cat hungry box broke smelled
slept dirty
slept dog crashed deep opened red rat slept dirty
a fell box window slept dog the
smelled
expensive ran clean died hungry black expensive the
cat slept dirty
smelled
broke rat slept dirty
dog crashed rat rat rat slept dirty
broke smelled
deep deep slid died hungry opened red slept hungry white the
hungry white the
clean crashed car car deep deep opened opened opened opened opened
box clean died hungry opened opened opened red rat rat slept
slid car car a fell box cat hungry opened red rat
died hungry black expensive the
car empty died hungry opened red dirty
red dog the
car a fell died hungry box cat slid died hungry white
expensive the
clean crashed black expensive the
crashed smelled

Random Data - Run 49 - Output Utterances

a cat window cat opened empty a broke expensive fell died
the hungry
white expensive smelled crashed dog fell white a rat slept fell
the dog fell rat died opened empty empty a empty slid
smelled crashed dog fell clean deep fell rat died car hungry
expensive empty dirty
black the dog opened empty dog fell died black the dog
a smelled crashed dog opened empty slid dirty
black broke expensive smelled crashed dirty
the dog fell died white a window cat slept box opened
dirty
died black cat window hungry
box opened empty slid dog opened empty dog fell opened empty
ran car expensive smelled crashed dirty
clean black broke window cat slept died white a ran car
hungry
the hungry
dirty
the died opened empty slid white deep fell died white deep
slept fell rat broke expensive smelled crashed ran car window cat
window cat opened empty cat window hungry
dirty
car hungry
slept a window cat opened empty slid deep rat broke window
dirty
broke expensive empty slid white black broke expensive fell opened empty
opened empty a the hungry
the died black broke window hungry
expensive fell white white deep empty slid dog fell died white
dog opened empty a window cat opened empty empty a ran
dog fell clean deep rat slept died car hungry
smelled crashed dog opened empty slid white white a window cat
window cat opened empty empty cat opened empty empty slid dirty
a ran car window hungry
ran car hungry
clean smelled smelled crashed dog fell clean slid white white
rat broke window hungry
black clean deep clean black crashed dirty
cat window hungry
the dirty
the broke window cat window cat opened empty dog fell died
deep clean smelled crashed ran car hungry
car hungry
smelled crashed dog fell white expensive fell opened empty dirty
fell died car expensive fell opened empty cat window cat slept
crashed dog opened empty dog fell white a broke window hungry
broke expensive empty empty slid white black white deep clean black
dog opened empty empty cat slept deep clean deep clean black
slid dirty
dirty

Random Data - Run 93 - Output Utterances

window fell ran smelled clean a black rat a empty the
dog fell ran slept dirty the fell ran slept empty the
car ran smelled clean cat car ran slept fell smelled clean
slid box slid box fell ran slept white broke dirty deep
slid box slid box slid box opened rat a window dog
white ran smelled cat slept dirty deep dog fell ran slept
red empty white broke dirty black slid box fell smelled clean
crashed black clean cat cat deep dog fell smelled clean cat
rat a empty the window dog fell smelled empty white broke
opened a black clean cat slept fell smelled cat car slid
car ran smelled clean cat cat cat cat box fell ran
black the fell ran slept a black rat a hungry crashed
slid box slid box opened rat a empty the empty the
black the empty white ran slept fell smelled clean a window
red empty the window dog fell ran slept white broke dirty
crashed black crashed the empty the empty white ran slept a
rat a hungry opened a window dog fell ran slept dirty
rat a window empty the fell smelled cat deep expensive ran
empty the window dog fell smelled cat expensive expensive expensive
slept a hungry opened rat a black the fell smelled cat
dog fell smelled empty white ran slept a hungry opened a
window empty white broke slid box fell ran slept a hungry
cat box fell smelled clean a black clean cat slept empty
broke dirty the empty the window empty the fell smelled empty
fell ran smelled clean a black clean cat slept white broke
deep dog fell smelled empty white ran slept empty the empty
expensive box slid box slid box opened rat a empty white

window dog fell ran slept a window fell smelled cat cat
died red empty the fell smelled clean a black the fell
rat a black rat a empty the window fell smelled cat
cat slept empty the empty white ran slept fell smelled clean
slid box opened rat a black clean cat car dirty deep
car ran slept a window fell ran smelled cat box opened
window dog fell ran smelled clean cat cat car ran smelled
dog fell smelled clean cat slept a window dog fell ran
empty white ran smelled clean cat cat box slid box fell
slid box slid box opened rat a hungry crashed black crashed
car dirty deep cat cat car dirty the empty white broke
black slid box fell smelled cat cat cat cat cat box
deep cat deep opened rat a window fell ran slept fell
red dirty the window dog fell smelled clean a hungry red
black the fell smelled cat slept dirty deep expensive box slid
expensive ran smelled cat deep cat slept white broke slid box
smelled empty the window dog fell ran smelled empty white broke
crashed ran slept dirty the window dog fell ran slept a
deep cat cat cat expensive expensive box fell smelled cat slept
rat a black crashed black slid box fell smelled cat deep
deep dog fell smelled cat deep rat a black rat a
dog fell ran smelled clean cat slept empty the fell smelled
white ran slept fell smelled cat cat box fell smelled empty

Non-random 10 Data - Run 14
Input Utterances

|a| |cat|
|box|
|cat|
|the| |clean| |cat| |died|
|a| |hungry| |rat|
|the| |box| |opened|
|rat|
|a| |rat| |slept|
|window|
|the| |deep| |box|

Known Chunks

a cat
box
cat
the clean cat died
the clean
died
a hungry rat
the box opened
the
opened
rat
a rat slept
a
slept
window
the deep box
the deep
deep box
deep

Atomic Level Words

box
cat
died
the
opened
rat
a
slept
window
deep

Category Groups

Output Utterances

window
the deep window
window
window
window
a rat slept
the window
the died
window
a rat slept

Novel Output Utterances

the deep window
the window
the died

Matching - All: 0.853550
Matching - 1s: 0.416667
Novel Utterances: 0.300000
Grammatical Utterances: 0.800000
Average Grammaticality: 0.850000
Novel Grammatical Utterances: 0.100000
Average Novel Grammaticality: 0.500000

Non-random 10 Data - Run 22
Input Utterances

|cat|
|the| |dirty| |car| |opened|
|dog|
|a| |window| |opened|
|the| |cat| |smelled|
|the| |expensive| |dog| |fell|
|the| |red| |box| |slid|
|the| |white| |car|
|a| |white| |car| |smelled|
|the| |window| |opened|

Known Chunks

cat
the dirty car opened
dog
a window opened
the cat smelled
the
smelled
the expensive dog fell
expensive dog fell
the expensive
fell
the red box slid
red box slid
the white car
white car
a white car smelled
a white car
a
white car smelled
the window opened
window opened

Atomic Level Words

cat
dog
the
smelled
fell
red box slid
white car
a
window opened

Category Groups

|cat| |white car| |cat| |white car|

Output Utterances

the smelled
a red box slid
a smelled
a the dog fell
a smelled
a cat the white car window opened
a cat a dog fell
a white car window opened
a window opened
the white car fell

Novel Output Utterances

the smelled
a red box slid
a smelled
a the dog fell
a smelled
a cat the white car window opened
a cat a dog fell
a white car window opened
the white car fell

Matching - All: 0.852071
Matching - 1s: 0.368421
Novel Utterances: 0.900000
Grammatical Utterances: 0.000000
Average Grammaticality: 0.175000
Novel Grammatical Utterances: 0.000000
Average Novel Grammaticality: 0.194444

Non-random 10 Data - Run 7
Input Utterances

|a| |black| |car| |smelled|
|window|
|rat|
|box|
|a| |car| |smelled|
|rat|

|the| |black| |cat|
|a| |cat|
|the| |car|
|a| |red| |car| |crashed|

Known Chunks

a black car smelled
window
rat
box
a car smelled
the black cat
a cat
the car
a red car crashed

Atomic Level Words

a black car smelled
window
rat
box
a car smelled
the black cat
a cat
the car
a red car crashed

Category Groups

Output Utterances

rat
a black car smelled
the black cat
a black car smelled
window
a cat
a red car crashed
rat
a red car crashed
window

Novel Output Utterances

Matching - All: 0.859467
Matching - 1s: 0.000000
Novel Utterances: 0.000000
Grammatical Utterances: 1.000000
Average Grammaticality: 1.000000
Novel Grammatical Utterances: 0.000000
Average Novel Grammaticality: 0.000000

Non-random 25 Data - Run 17
Input Utterances

|the| |black| |rat| |died|
|dog|
|the| |white| |rat|
|window|
|a| |hungry| |rat| |fell|
|the| |box|
|the| |red| |car| |opened|
|the| |empty| |car|
|cat|
|the| |black| |rat| |fell|
|a| |clean| |rat|
|the| |hungry| |rat| |ran|
|a| |cat|
|the| |box|
|a| |expensive| |dog|
|a| |dirty| |rat| |slept|
|the| |white| |dog| |died|
|the| |dirty| |car|
|the| |car| |smelled|
|a| |black| |car| |crashed|
|car|
|a| |deep| |box| |slid|
|a| |empty| |car|
|a| |red| |box|
|the| |black| |box| |opened|

Known Chunks

the black rat died
dog
the white rat
window
a hungry rat fell
the box
the red car opened
the empty car
cat
the black rat fell
a clean rat
the hungry rat ran
a cat
a
a expensive dog
a expensive
expensive dog
expensive
a dirty rat slept
dirty rat slept
the white dog died
the white
died
the dirty car
the car smelled
a black car crashed
black car crashed
car
a deep box slid
deep box slid
a empty car
a empty
empty car
empty
a red box
red box
the black box opened

Atomic Level Words

dog
the white rat
window
the box
cat
the black rat fell
the hungry rat ran
a
expensive
dirty rat slept
the white
died
car
deep box slid
empty
red box
the black box opened

Category Groups

|cat| |expensive| |dirty rat slept| |deep box slid| |empty| |red box|

Output Utterances

a cat
the white rat
a red box
the hungry rat ran
the black rat fell
a dirty rat slept
the white rat
the hungry rat ran
the black box opened
a expensive the hungry rat ran
the black box opened
the black rat fell
the black rat fell
the box
the white rat
the black rat fell
the box
the white the black rat fell
the white rat
the black box opened
the black rat fell
a red box
the black box opened
a deep box slid
the white the hungry rat ran

Novel Output Utterances

a expensive the hungry rat ran
the white the black rat fell
the white the hungry rat ran

Matching - All: 0.866864
Matching - 1s: 0.692308
Novel Utterances: 0.120000
Grammatical Utterances: 0.720000
Average Grammaticality: 0.740000
Novel Grammatical Utterances: 0.000000
Average Novel Grammaticality: 0.166667

Non-random 25 Data - Run 67
Input Utterances

|the| |rat| |fell|
|window|
|cat|
|a| |hungry| |cat| |slid|
|box|
|cat|
|dog|
|the| |rat| |slept|
|the| |empty| |box| |smelled|
|a| |clean| |cat| |smelled|
|the| |box|
|the| |box|
|the| |empty| |box|
|box|
|a| |clean| |car|
|a| |red| |box|
|window|
|a| |clean| |car|
|the| |window|
|window|
|a| |clean| |car|
|window|
|the| |deep| |box|
|the| |deep| |box| |broke|
|the| |black| |rat| |slept|

Known Chunks

the rat fell
window
cat
a hungry cat slid

a hungry
slid
box
dog
the rat slept
the empty box smelled
the empty
smelled
a clean cat smelled
a clean cat
a clean
the box
the
the empty box
empty box
empty
a clean car
car
a red box
a red
the window
the deep box
the deep
deep box
deep
the deep box broke
broke
box broke
deep box broke
the black rat slept
black rat slept

Atomic Level Words

window
cat
a hungry
slid
box
dog
smelled
a clean
the
empty
car
a red
deep
broke
black rat slept

Category Groups

|slid| |smelled|
|cat| |cat| |box|
|smelled| |broke|
|window| |box| |empty| |deep| |black rat slept|

Output Utterances

dog
a clean car
a hungry cat car
dog
a hungry a red empty window
a hungry a hungry slid
a red black rat slept
a clean dog
dog
a clean smelled
a hungry slid
a clean cat car
the window
a hungry the black rat slept
a clean box cat window
dog
the black rat slept
dog
the deep black rat slept
a clean black rat slept
a clean the box a red black rat slept
the window
a clean broke
dog
a clean a red empty deep empty window

Novel Output Utterances

a hungry cat car
a hungry a red empty window
a hungry a hungry slid
a red black rat slept
a clean dog
a clean smelled
a hungry slid
a clean cat car
a hungry the black rat slept
a clean box cat window
the deep black rat slept
a clean black rat slept
a clean the box a red black rat slept
a clean broke
a clean a red empty deep empty window

Matching - All: 0.828402
Matching - 1s: 0.255814
Novel Utterances: 0.600000
Grammatical Utterances: 0.320000
Average Grammaticality: 0.345000
Novel Grammatical Utterances: 0.000000
Average Novel Grammaticality: 0.041667

Non-random 25 Data - Run 8
Input Utterances

|cat|
|a| |expensive| |cat| |slept|
|car|
|box|
|car|
|window|
|rat|
|the| |car|
|a| |red| |box|
|cat|
|car|
|box|
|window|
|window|
|the| |cat|
|the| |clean| |box| |smelled|
|cat|
|a| |empty| |box|
|cat|
|a| |deep| |box|
|rat|
|cat|
|the| |window| |opened|
|the| |cat| |slid|
|a| |box|

Known Chunks

cat
a expensive cat slept
a expensive
slept
car
box
window
rat
the car
the
a red box
a red
the cat
the clean box smelled
clean box smelled
the clean
smelled
a empty box
a empty
a deep box
a deep
the window opened
window opened
opened
the cat slid
slid
cat slid
a box
a

Atomic Level Words

cat
slept
car
box
window
rat
the
smelled
opened
slid
a

Category Groups

|slept| |slid|
|cat| |cat| |window|

Output Utterances

rat
rat
the car
the window opened
a box smelled
a box smelled
a box smelled
the cat slept
rat
a box smelled
rat
the window opened
rat
rat
a box smelled
rat
a box smelled
the window opened
a box smelled
the window opened
the cat slid
rat
a box smelled
rat
rat

Novel Output Utterances

a box smelled
a box smelled
a box smelled
the cat slept
a box smelled
a box smelled
a box smelled
a box smelled
a box smelled

Matching - All: 0.868343
Matching - 1s: 0.800000

Novel Utterances: 0.360000
Grammatical Utterances: 1.000000
Average Grammaticality: 1.000000
Novel Grammatical Utterances: 0.360000
Average Novel Grammaticality: 1.000000

Non-random 50 Data - Run 10
Input Utterances

|rat|
a| |car| |opened|
a| |empty| |car|
a| |box|
window|
car|
the| |deep| |box|
rat|
window|
the| |dirty| |rat|
a| |black| |cat| |ran|
window|
the| |dog| |slid|
a| |red| |box|
the| |black| |dog|
cat|
car|
a| |dirty| |box|
a| |white| |dog| |slept|
rat|
a| |window| |opened|
the| |rat| |slid|
a| |cat|
the| |clean| |rat|
the| |hungry| |rat|
a| |black| |rat|
the| |white| |car|
a| |dog| |slept|
car|
the| |dog|
cat|
a| |hungry| |cat| |ran|
rat|
a| |deep| |box| |slid|
the| |dirty| |cat|
box|
the| |black| |box| |broke|
a| |box|
box|
car|
cat|
the| |black| |cat| |fell|
the| |hungry| |dog| |ran|
window|
the| |deep| |box| |fell|
a| |dog| |slid|
a| |window|
box|
window|
|the| |rat|

Known Chunks

rat
a car opened
a empty car
a box
window
car
the deep box
the dirty rat
the dirty
a black cat ran
the dog slid
a red box
the black dog
cat
a dirty box
a white dog slept
a window opened
a
opened
the rat slid
the
slid
a cat
the clean rat
the clean
clean rat
clean
the hungry rat
the hungry
hungry rat
hungry
a black rat
a black
black rat
black
the white car
the white
white car
white
a dog slept
dog slept
the dog
dog
a hungry cat ran
hungry cat ran
cat ran
a hungry
ran
a deep box slid
a deep box
deep box
deep box slid
the dirty cat
dirty cat
dirty
box
the black box broke
black box broke
box broke
the black
broke
the black cat fell
cat fell
black cat fell
fell
the hungry dog ran
the hungry dog
dog ran
hungry dog
hungry dog ran
the deep box fell
deep box fell
the deep
a dog slid
a dog
dog slid
a window
the rat

Atomic Level Words

rat
window
car
cat
a
opened
the
slid
clean
hungry
black
white
dog
ran
dirty
box
broke
fell

Category Groups

|ran| |fell|
|window| |cat| |dog| |cat| |box| |the| |clean| |hungry| |black| |the| |clean| |hungry| |black| |rat| |dog| |cat| |dog| |box| |cat| |box|
|rat| |dog|
|clean| |hungry| |black|
|hungry| |black|
|slid| |ran|
|cat| |dog| |cat| |box|
|broke| |fell|

Output Utterances

a box black black black dog fell
a hungry black black clean clean black black clean rat box white car
a rat opened
a clean black dog slid
the black dirty black black white car
a cat broke
a black dog ran
a dirty clean dirty dirty hungry clean dog slid
a box dog slid
a hungry dirty white car
a black white car
the white car
the hungry hungry black rat black rat hungry black black hungry hungry black clean black hungry rat ran
a rat cat fell
the dog slid
a hungry clean dirty slid
the dirty slid
a white car
the black dirty dirty cat fell
a dog ran
a clean clean black white car
the box broke
a box slid
the clean black black dog ran
the rat ran
the clean black rat broke
the dirty fell
a hungry black dog slid
a rat fell
the clean black hungry black hungry hungry black hungry rat cat slid
the dirty hungry dog ran
the black hungry rat a black hungry black hungry white car
the dirty ran
the dog slid
the dog fell
the window opened
a clean hungry dirty clean hungry dog slid
a black dirty black dirty ran
the black hungry clean clean rat clean dog fell
a dirty clean clean dog fell
the box the window opened
a box box rat car
the cat ran
the black dog fell
the black black clean dog slid
a black black dog ran
the box fell
the black dirty fell
the black dirty dog slid

a black hungry white car

Novel Output Utterances

a box black black black dog fell
a hungry black black clean clean black black clean rat box white car
a rat opened
a clean black dog slid
the black dirty black black white car
a cat broke
a black dog ran
a dirty clean dirty dirty hungry clean dog slid
a box dog slid
a hungry dirty white car
a black white car
the hungry hungry black rat black rat hungry black black hungry hungry black clean black hungry rat ran
a rat cat fell
a hungry clean dirty slid
the dirty slid
a white car
the black dirty dirty cat fell
a dog ran
a clean clean black white car
the box broke
a box slid
the clean black black dog ran
the rat ran
the clean black rat broke
the dirty fell
a hungry black dog slid
a rat fell
the clean black hungry black hungry hungry black hungry rat cat slid
the dirty hungry dog ran
the black hungry rat a black hungry black hungry white car
the dirty ran
the dog fell
the window opened
a clean hungry dirty clean hungry dog slid
a black dirty black dirty ran
the black hungry clean clean rat clean dog fell
a dirty clean clean dog fell
the box the window opened
a box box rat car
the cat ran
the black dog fell
the black black clean dog slid
a black black dog ran
the box fell
the black dirty fell
the black dirty dog slid
a black hungry white car

Matching - All: 0.823964
Matching - 1s: 0.407692
Novel Utterances: 0.940000
Grammatical Utterances: 0.300000
Average Grammaticality: 0.742340
Novel Grammatical Utterances: 0.240000
Average Novel Grammaticality: 0.725894

Non-random 50 Data - Run 17

Input Utterances

|the| |car|
cat
a| |white| |car| |opened|
car
a| |white| |cat|
the| |box| |fell|
the| |rat| |slept|
window|
the| |white| |cat| |died|
a| |black| |rat|
the| |red| |box|
the| |dirty| |cat|
the| |deep| |box| |broke|
car
a| |empty| |box|
the| |empty| |box| |broke|
box
rat
the| |deep| |box| |opened|
car
window|
car
the| |black| |rat| |died|
the| |red| |box|
cat
the| |expensive| |car| |crashed|
the| |cat|
a| |black| |rat| |fell|
rat
the| |expensive| |box|
cat
a| |black| |car| |opened|
car
box
a| |dog|
the| |red| |box| |opened|
the| |car|
cat
a| |hungry| |cat|
box
a| |red| |box|
car
the| |deep| |box|
the| |expensive| |car|
rat
the| |clean| |cat| |ran|
a| |hungry| |dog| |slid|
the| |red| |car|
box
|a| |expensive| |window|

Known Chunks

the car
cat
a white car opened
car
a white cat
a white
the box fell
the rat slept
window
the white cat died
the white
died
a black rat
the red box
the dirty cat
the dirty
the deep box broke
a empty box
the empty box broke
box
rat
the deep box opened
the deep
opened
the black rat died
the black rat
the black
the red
the expensive car crashed
the expensive
crashed
the cat
the
a black rat fell
fell
a black
the expensive box
expensive box
expensive
a black car opened
a black car
car opened
a dog
the red box opened
box opened
red box
red box opened
red
a hungry cat
a hungry
a red box
a red
a
the deep box
deep box
deep
the expensive car
expensive car
the clean cat ran
clean cat ran
the clean
ran
a hungry dog slid
dog slid
hungry dog slid
hungry
the red car
red car
a expensive window
a expensive
expensive window

Atomic Level Words

cat
car
window
died
box
rat
opened
crashed
the
fell
expensive
red
a
deep
ran
dog slid
hungry

Category Groups

|died| |ran|
|opened| |crashed|
|died| |fell|
|expensive| |red| |deep|
|red| |expensive|
|car| |box| |car| |car| |box|
|the| |expensive| |red| |expensive| |expensive| |red| |deep| |the| |expensive| |red| |expensive| |red| |deep| |hungry|

Output Utterances

the red red expensive deep car red red deep window

the expensive car box fell
a deep window
the car red red expensive red car expensive red expensive expensive deep car ran
the box red expensive box rat crashed
a fell
a dog slid
a red car died
a expensive box fell
a box red car died
a box dog slid
a crashed
the box window
a the box rat fell
the deep car red expensive red deep box died
the cat crashed
a fell
a window
a window
a a died
the window
a car a rat the expensive cat red car car opened
a deep car died
the car rat expensive red window
the red cat hungry dog slid
a died
the expensive window
the expensive deep window
a expensive deep box car cat crashed
a red car crashed
the red cat expensive expensive window
a deep box hungry dog slid
the red deep car fell
the cat ran
the expensive cat car hungry dog slid
the hungry dog slid
a died
the cat box crashed
a deep window
the red box cat car died
a fell
the red car opened
the hungry dog slid
a expensive red expensive red expensive cat a opened
a rat expensive car fell
the car box ran
the window
a hungry dog slid
a car box deep car car rat died
a deep box opened

Novel Output Utterances

the red red expensive deep car red red deep window
the expensive car box fell
a deep window
the car red red expensive red car expensive red expensive expensive deep car ran
the box red expensive box rat crashed
a fell
a dog slid
a red car died
a expensive box fell
a box red car died
a box dog slid
a crashed
the box window
a the box rat fell
the deep car red expensive red deep box died
the cat crashed
a fell
a window
a window
a a died
the window
a car a rat the expensive cat red car car opened
a deep car died
the car rat expensive red window
the red cat hungry dog slid
a died
the expensive window
the expensive deep window
a expensive deep box car cat crashed
a red car crashed
the red cat expensive expensive window
a deep box hungry dog slid
the red deep car fell
the cat ran
the expensive cat car hungry dog slid
the hungry dog slid
a died
the cat box crashed
a deep window
the red box cat car died
a fell
the red car opened
the hungry dog slid
a expensive red expensive red expensive cat a opened
a rat expensive car fell
the car box ran
the window
a car box deep car car rat died
a deep box opened

Matching - All: 0.792899
Matching - 1s: 0.300885
Novel Utterances: 0.980000
Grammatical Utterances: 0.200000
Average Grammaticality: 0.416841
Novel Grammatical Utterances: 0.200000
Average Novel Grammaticality: 0.415144

Non-random 50 Data - Run 29
Input Utterances

|the| |white| |rat| |ran|
|a| |rat| |died|
|the| |car|
|car|
|cat|
|the| |black| |cat|
|a| |expensive| |dog|
|rat|
|rat|
|the| |car| |slid|
|box|
|the| |clean| |box|
|cat|
|a| |hungry| |rat| |fell|
|a| |deep| |box| |fell|
|dog|
|a| |box|
|dog|
|rat|
|the| |hungry| |cat| |fell|
|window|
|a| |rat| |smelled|
|a| |dirty| |box|
|window|
|the| |rat|
|the| |dog| |smelled|
|the| |hungry| |dog| |slept|
|dog|
|a| |rat| |ran|
|a| |black| |box|
|the| |black| |dog|
|a| |cat| |smelled|
|rat|
|car|
|the| |dog|
|the| |rat| |smelled|
|the| |box|
|window|
|box|
|rat|
|the| |dog| |slept|
|the| |red| |box|
|box|
|box|
|the| |window| |broke|
|a| |expensive| |cat| |ran|
|the| |red| |box|
|the| |window| |fell|
|a| |red| |box| |slid|
|a| |hungry| |cat|

Known Chunks

the white rat ran
a rat died
the car
car
cat
the black cat
the black
a expensive dog
rat
the car slid
slid
the
box
the clean box
the clean
clean box
clean
a hungry rat fell
a hungry
fell
a deep box fell
a deep box
a deep
dog
a box
a
the hungry cat fell
the hungry cat
hungry cat
hungry cat fell
the hungry
window
a rat smelled
rat smelled
smelled
a dirty box
a dirty
dirty box
dirty
the rat
the dog smelled
the dog
dog smelled
the hungry dog slept
dog slept
hungry dog slept
hungry
slept
a rat ran
rat ran
ran
a black box
a black
black box
black
the black dog
black dog
a cat smelled
a cat
cat smelled

```
the rat smelled
the box
the dog slept
the red box
the red
red box
red
the window broke
window broke
broke
a expensive cat ran
a expensive cat
expensive cat
expensive cat ran
a expensive
the window fell
the window
window fell
a red box slid
a red box
red box slid
box slid
a red
a hungry cat
```

Atomic Level Words

```
car
cat
rat
slid
the
box
clean
fell
dog
a
window
smelled
dirty
hungry
slept
ran
black
red
broke
```

Category Groups

```
|fell| |smelled| |ran|
|car| |box|
|rat| |box| |dog|
|box| |rat|
|clean| |black| |red|
|dog| |rat|
|window| |rat| |box|
|hungry| |rat|
|black| |red|
|slid| |fell|
|car| |box| |cat| |rat| |box|
   |window|
|smelled| |slept|
|cat| |rat|
|rat| |cat| |box|
|dirty| |black| |red|
|fell| |broke|
|cat| |rat| |box| |window| |cat|
   |rat| |dog| |cat| |rat|
|car| |box| |cat| |rat| |box|
   |window| |cat| |rat| |dog|
   |dog|
```

Output Utterances

```
a box fell
the red cat slid
a rat broke
a rat slept
a dog fell
a dog fell
the car car slid
the box broke
a rat fell
the car clean rat ran
the red hungry clean rat slept
a dirty box fell
the rat ran
the box smelled
a hungry dirty box broke
a black cat smelled
the box slid
a rat slept
the black box smelled
the car window slid
a window broke
a box fell
the hungry broke
a box broke
the clean dog slept
the box smelled
a rat fell
a cat smelled
a hungry red cat fell
the dog ran
a hungry broke
the box broke
the car cat broke
a red box fell
the clean dog slept
a red hungry rat slid
a rat smelled
a dog slept
the window ran
the dog fell
the dog ran
a box ran
a dog fell
a dog ran
a red car window fell
the rat smelled
a dirty window slid
the rat slid
the window broke
the rat smelled
```

Novel Output Utterances

```
a box fell
the red cat slid
a rat broke
a rat slept
a dog fell
a dog fell
the car car slid
the box broke
a rat fell
the car clean rat ran
the red hungry clean rat slept
a dirty box fell
the rat ran
the box smelled
a hungry dirty box broke
a black cat smelled
the box slid
a rat slept
the black box smelled
the car window slid
a window broke
a box fell
the hungry broke
a box broke
the clean dog slept
the box smelled
a rat fell
a hungry red cat fell
the dog ran
a hungry broke
the box broke
the car cat broke
a red box fell
the clean dog slept
a red hungry rat slid
a dog slept
the window ran
the dog fell
the dog ran
a box ran
a dog fell
a dog ran
a red car window fell
a dirty window slid
the rat slid
```

Matching - All: 0.778107
Matching - 1s: 0.352941
Novel Utterances: 0.900000
Grammatical Utterances: 0.680000
Average Grammaticality: 0.876250
Novel Grammatical Utterances: 0.580000
Average Novel Grammaticality: 0.862500

Non-random 75 Data - Run 62

Input Utterances

```
|dog|
|car|
|the| |white| |dog|
|cat|
|box|
|a| |cat|
|the| |deep| |box| |opened|
|the| |black| |box| |smelled|
|cat|
|the| |cat| |slept|
|a| |window|
|the| |deep| |box|
|rat|
|a| |deep| |box|
|the| |clean| |window|
|the| |expensive| |cat|
|dog|
|a| |dog|
|a| |box| |fell|
|the| |expensive| |cat| |slid|
|the| |expensive| |cat| |slid|
|a| |hungry| |cat| |slept|
|a| |white| |rat| |slid|
|a| |white| |rat| |fell|
|a| |white| |cat| |ran|
|the| |clean| |rat|
|box|
|the| |cat|
|a| |white| |car| |smelled|
|rat|
|a| |white| |car|
|the| |hungry| |rat|
|a| |car|
|a| |box| |opened|
|box|
|the| |white| |cat|
|box|
|the| |rat| |died|
|the| |deep| |box|
|a| |car|
|a| |empty| |car|
|a| |dog|
|a| |deep| |box| |fell|
|the| |red| |car|
|the| |white| |cat|
|the| |dog| |died|
|the| |box| |broke|
|a| |clean| |box|
|the| |clean| |cat|
|a| |red| |box| |smelled|
|a| |empty| |box| |broke|
|a| |empty| |car| |slid|
|the| |empty| |box|
|dog|
|the| |clean| |cat| |died|
|the| |clean| |window| |fell|
|a| |hungry| |cat| |smelled|
```

```
|a| |box|
|the| |box|
|a| |rat| |slid|
|the| |expensive| |window|
|the| |clean| |car|
window|
|the| |deep| |box| |broke|
window|
|the| |red| |car|
cat|
cat|
|the| |expensive| |car| |opened|
window|
|the| |red| |car| |opened|
|the| |empty| |box| |fell|
|a| |dirty| |box|
cat|
|the| |window| |opened|
```

Known Chunks

```
dog
car
the white dog
the white
cat
box
a cat
a
the deep box opened
the deep
opened
the black box smelled
the black
smelled
the cat slept
the
slept
a window
window
the deep box
deep box
deep
rat
a deep box
a deep
the clean window
the clean
clean window
clean
the expensive cat
the expensive
expensive cat
expensive
a dog
a box fell
box fell
fell
the expensive cat slid
slid
cat slid
expensive cat slid
a hungry cat slept
a hungry cat
hungry cat
hungry cat slept
a hungry
a white rat slid
a white rat
white rat
white rat slid
a white
a white rat fell
rat fell
white rat fell
white
a white cat ran
cat ran
white cat ran
ran
the clean rat
clean rat
the cat
a white car smelled
a white car
car smelled
white car
white car smelled
the hungry rat
the hungry
hungry rat
hungry
a car
a box opened
a box
box opened
the white cat
white cat
the rat died
rat died
died
a empty car
a empty
empty car
empty
a deep box fell
deep box fell
the red car
the red
red car
red
the dog died
the dog
dog died
the box broke
box broke
broke
a clean box
a clean
clean box
the clean cat
clean cat
a red box smelled
a red box
red box
red box smelled
box smelled
a red
a empty box broke
a empty box
empty box
empty box broke
a empty car slid
car slid
empty car slid
the empty box
the empty
the clean cat died
cat died
clean cat died
the clean window fell
window fell
clean window fell
a hungry cat smelled
cat smelled
hungry cat smelled
the box
a rat slid
a rat
rat slid
the expensive window
expensive window
the clean car
clean car
the deep box broke
deep box broke
the expensive car opened
the expensive car
car opened
expensive car
expensive car opened
the red car opened
red car opened
the empty box fell
empty box fell
a dirty box
a dirty
dirty box
dirty
the window opened
the window
window opened
```

Atomic Level Words

```
dog
car
cat
box
a
opened
smelled
the
slept
window
deep
rat
clean
expensive
fell
slid
white
ran
hungry
died
empty
red
broke
dirty
```

Category Groups

```
|opened| |smelled| |slid|
|smelled| |slept| |slid| |ran|
   |died|
|opened|   |smelled|   |fell|
   |broke|
|dog| |cat| |rat|
|deep| |clean| |empty|
|clean| |deep| |white| |empty|
   |red|
|white| |clean| |empty| |red|
|hungry| |clean|
|dirty| |deep| |clean| |empty|
|cat| |box| |window| |rat|
|clean|   |deep|   |expensive|
   |white| |empty| |red|
|expensive|  |clean|  |white|
   |red|
|red|   |clean|   |expensive|
   |white| |empty|
|opened| |fell|
|fell| |slid| |died|
|car| |box| |window| |car| |cat|
   |box|   |car|   |cat|   |rat|
   |car|   |cat|   |box|   |cat|
   |car|   |cat|   |rat|   |cat|
   |dog|   |cat|   |rat|   |car|
   |box|  |window|  |car|  |cat|
   |box|  |box|  |window|  |rat|
   |box|  |car|  |box|  |window|
   |box|  |window|  |rat|  |box|
   |window|  |rat|  |car|  |cat|
   |rat|  |dog|  |cat|  |rat|
|car| |box| |window| |car| |cat|
   |box|   |car|   |cat|   |rat|
   |car|  |box|  |window|  |car|
   |cat|  |box|  |box|  |window|
   |rat|  |box|
```

Output Utterances

```
the red hungry dirty clean cat
   ran
the rat slid
the cat opened
a cat died
a car slept
```

the hungry expensive rat died
a empty red cat slid
a car fell
the expensive box ran
a box smelled
a rat opened
a expensive car slept
a cat fell
a clean cat died
a cat died
a box ran
a white dirty clean red red cat smelled
a expensive empty cat slid
the rat died
a cat opened
the expensive deep hungry red red empty expensive window opened
the empty box fell
a empty cat died
a dog smelled
a red cat died
the box opened
the white car fell
the deep clean clean deep car fell
a hungry deep cat fell
the window fell
a clean deep box ran
a dirty window smelled
the box broke
a box slept
a dog smelled
a box opened
a clean car fell
the cat smelled
a white deep dog slept
the hungry car fell
the cat smelled
a deep expensive window opened
the car ran
the rat died
the red deep empty deep rat fell
a hungry cat died
a box died
the empty hungry empty window broke
a box slid
a clean clean dirty rat smelled
the box died
a rat fell
a dog died
the expensive white deep empty empty expensive clean deep expensive box smelled
the hungry empty empty empty white white red car smelled
the dog slept
the deep dirty white deep cat fell
the empty clean clean car slept
a rat fell
the clean red clean deep deep empty cat slid
a window opened
the clean empty window slid
the deep hungry dog fell
the clean clean clean dog smelled
a clean window opened
a clean white clean white clean dog died
the deep empty expensive rat fell
a car opened
the rat fell
a car fell
a clean white box fell
the white rat broke
a white hungry empty hungry clean empty clean clean empty car slept
a window opened
a box fell

Novel Output Utterances

the red hungry dirty clean cat ran
the rat slid
the cat opened
a cat died
a car slept
the hungry expensive rat died
a empty red cat slid
a car fell
the expensive box ran
a box smelled
a rat opened
a expensive car slept
a cat fell
a clean cat died
a cat died
a box ran
a white dirty clean red red cat smelled
a expensive empty cat slid
a cat opened
the expensive deep hungry red red empty expensive window opened
a empty cat died
a dog smelled
a red cat died
the box opened
the white car fell
the deep clean clean deep car fell
a hungry deep cat fell
the window fell
a clean deep box ran
a dirty window smelled
a box slept
a dog smelled
a clean car fell
the cat smelled
a white deep dog slept
the hungry car fell
the cat smelled
a deep expensive window opened
the car ran
the red deep empty deep rat fell
a hungry cat died
a box died
the empty hungry empty window broke
a box slid
a clean clean dirty rat smelled
the box died
a rat fell
a dog died
the expensive white deep empty empty expensive clean deep expensive box smelled
the hungry empty empty empty white white red car smelled
the dog slept
the deep dirty white deep cat fell
the empty clean clean car slept
a rat fell
the clean red clean deep deep empty cat slid
a window opened
the clean empty window slid
the deep hungry dog fell
the clean clean clean dog smelled
a clean window opened
a clean white clean white clean dog died
the deep empty expensive rat fell
a car opened
the rat fell
a car fell
a clean white box fell
the white rat broke
a white hungry empty hungry clean empty clean clean empty car slept
a window opened

Matching - All: 0.723373
Matching - 1s: 0.327068
Novel Utterances: 0.920000
Grammatical Utterances: 0.386667
Average Grammaticality: 0.729818
Novel Grammatical Utterances: 0.306667
Average Novel Grammaticality: 0.706324

Non-random 75 Data - Run 37

Input Utterances

a| |deep| |box| |broke|
car|
the| |dog|
dog|
the| |cat|
a| |expensive| |window| |broke|
a| |window| |broke|
a| |window|
the| |empty| |box|
car|
a| |white| |box| |fell|
dog|
car|
window|
cat|
dog|
a| |box|
a| |white| |rat| |slept|
a| |red| |box|
cat|
the| |white| |car| |smelled|
a| |expensive| |box| |slid|
the| |box|
the| |dog|
the| |dirty| |car| |smelled|
the| |empty| |box|
a| |red| |car|
a| |clean| |car|
box|
a| |empty| |box| |broke|
a| |cat|
the| |black| |cat| |died|
the| |white| |rat|
window|
car|
window|
box|
the| |black| |dog| |fell|
the| |empty| |box|
rat|
a| |window|
a| |car| |smelled|
a| |hungry| |cat|
the| |window| |opened|
the| |rat| |ran|
window|
a| |white| |rat| |fell|
the| |clean| |rat|
the| |box|
the| |car| |smelled|
the| |dog| |ran|
the| |clean| |box| |broke|
a| |red| |car|
a| |cat| |died|
window|
a| |car|
the| |car|
the| |dog|
a| |deep| |box| |slid|
a| |black| |dog| |slid|
a| |rat|
the| |hungry| |dog| |fell|
a| |empty| |box| |broke|
window|
a| |window|

```
|cat|
|car|
|the| |box| |fell|
|rat|
|window|
|a| |box|
|a| |dirty| |window| |opened|
|window|
|a| |box| |slid|
|a| |white| |dog| |smelled|
```

Known Chunks

```
a deep box broke
car
the dog
dog
the cat
a expensive window broke
a window broke
a window
the empty box
a white box fell
window
cat
a box
a white rat slept
a red box
the white car smelled
the white
smelled
a expensive box slid
the box
the
the dirty car smelled
the dirty car
dirty car
dirty car smelled
the dirty
empty box
a red car
a red
a clean car
a clean
box
a empty box broke
a
broke
empty box broke
a empty
a cat
the black cat died
black cat died
the black
died
the white rat
rat
white rat
white
the black dog fell
dog fell
black dog fell
black
fell
the empty
empty
a car smelled
a car
car smelled
a hungry cat
a hungry
hungry cat
hungry
the window opened
window opened
opened
the rat ran
rat ran
ran
a white rat fell
a white rat
white rat fell
rat fell
a white
the clean rat
the clean
clean rat
clean
the car smelled
the car
the dog ran
dog ran
the clean box broke
the clean box
box broke
clean box
clean box broke
red car
red
a cat died
cat died
a deep box slid
deep box slid
a deep
slid
a black dog slid
a black dog
black dog
black dog slid
dog slid
a black
a rat
the hungry dog fell
the hungry dog
the hungry
hungry dog
hungry dog fell
a empty box
the box fell
box fell
a dirty window opened
a dirty window
a dirty
dirty window
dirty window opened
dirty
a box slid
box slid
a white dog smelled
a white dog
dog smelled
white dog
white dog smelled
```

Atomic Level Words

```
car
dog
window
cat
smelled
the
box
a
broke
died
rat
white
black
fell
empty
hungry
opened
ran
clean
red
slid
dirty
```

Category Groups

```
|smelled| |fell| |ran| |slid|
|car| |dog|
|dog| |car| |box| |rat|
|white| |clean|
|black| |hungry|
|empty| |clean|
|clean| |white| |empty|
|broke| |fell| |slid|
|box| |rat|
|rat| |car| |box|
|fell| |ran|
|box| |dog| |box| |rat| |dog|
   |box|
```

Output Utterances

```
a the slid
a broke
the car slid
the the box fell
the fell
the slid
the rat ran
a clean dirty box fell
a slid
the fell
the rat ran
a slid
the clean box slid
the black empty hungry box fell
a clean window opened
a died
a ran
a dog fell
a car a box slid
a car dirty slid
a white white window opened
the died
the empty dirty window opened
the box ran
a ran
a dog smelled
the white car fell
a clean box smelled
the smelled
the smelled
a fell
a box fell
a box ran
the a ran
a cat died
the ran
a fell
a fell
the window opened
the fell
a box fell
the window opened
a dog broke
the black died
a fell
the box ran
a window opened
the ran
a smelled
a dog broke
the empty cat died
the broke
the a dog smelled
a opened
the cat died
the ran
a black rat broke
a the fell
the ran
the box slid
the dog smelled
the black car slid
the died
the rat broke
the died
a car ran
a car empty car empty red black
   ran
the slid
the rat fell
the hungry black ran
```

the the hungry car fell
a clean hungry hungry empty box fell
the white clean box fell
the black smelled
the slid

Novel Output Utterances

a the slid
a broke
the car slid
the the box fell
the fell
the slid
a clean dirty box fell
a slid
the fell
a slid
the clean box slid
the black empty hungry box fell
a clean window opened
a died
a ran
a dog fell
a car a box slid
a car dirty slid
a white white window opened
the died
the empty dirty window opened
the box ran
a ran
a dog smelled
the white car fell
a clean box smelled
the smelled
the smelled
a fell
a box fell
a box ran
the a ran
the ran
a fell
a fell
the fell
a box fell
a dog broke
the black died
a fell
the box ran
a window opened
the ran
a smelled
a dog broke
the empty cat died
the broke
the a dog smelled
a opened
the cat died
the ran
a black rat broke
a the fell
the ran
the box slid
the dog smelled
the black car slid
the died
the rat broke
the died
a car ran
a car empty car empty red black ran
the slid
the rat fell
the hungry black ran
the the hungry car fell
a clean hungry hungry empty box fell
the white clean box fell
the black smelled
the slid

Matching - All: 0.726331
Matching - 1s: 0.306034
Novel Utterances: 0.933333
Grammatical Utterances: 0.253333
Average Grammaticality: 0.438542
Novel Grammatical Utterances: 0.186667
Average Novel Grammaticality: 0.398438

Non-random 75 Data - Run 54
Input Utterances

|the| |box| |smelled|
a| |clean| |box|
the| |hungry| |rat|
the| |red| |car| |slid|
rat|
window|
rat|
a| |box| |slid|
car|
the| |white| |box|
a| |hungry| |dog| |slept|
a| |rat|
dog|
a| |hungry| |dog| |fell|
window|
the| |empty| |car| |slid|
the| |window|
window|
cat|
the| |clean| |rat|
the| |window| |broke|
rat|
a| |car|
a| |car|
cat|
a| |car|
a| |dirty| |dog|
the| |expensive| |box|
car|
a| |window| |broke|
box|
the| |cat|
box|
cat|
the| |black| |cat| |ran|
the| |dog|
the| |white| |car|
a| |window|
box|
a| |dirty| |dog|
dog|
a| |clean| |dog| |slept|
the| |white| |car| |crashed|
a| |dirty| |rat|
rat|
car|
window|
window|
dog|
rat|
a| |hungry| |cat| |fell|
the| |white| |dog| |slept|
the| |dog| |died|
a| |car|
window|
a| |rat| |fell|
the| |black| |car| |crashed|
dog|
dog|
dog|
the| |empty| |car| |crashed|
the| |dirty| |dog| |fell|
a| |rat|
the| |car| |smelled|
the| |red| |box|
window|
the| |red| |car|
the| |dog| |ran|
the| |car| |opened|
a| |hungry| |rat| |slept|
dog|
cat|
the| |clean| |cat|
the| |car| |smelled|
a| |window|

Known Chunks

the box smelled
a clean box
the hungry rat
the red car slid
rat
window
a box slid
car
the white box
a hungry dog slept
a rat
a
dog
a hungry dog fell
hungry dog fell
a hungry
fell
the empty car slid
the empty
slid
the window
the
cat
the clean rat
the clean
clean rat
clean
the window broke
broke
window broke
a car
a dirty dog
a dirty
dirty dog
dirty
the expensive box
expensive box
a window broke
a window
box
the cat
the black cat ran
black cat ran
the black
ran
the dog
the white car
the white
white car
white
a clean dog slept
clean dog slept
dog slept
a clean
slept
the white car crashed
crashed
car crashed
white car crashed
a dirty rat
dirty rat
a hungry cat fell
a hungry cat
cat fell
hungry cat
hungry cat fell
hungry
the white dog slept
the white dog
white dog
white dog slept
the dog died
died

dog died
a rat fell
rat fell
the black car crashed
the black car
black car
black car crashed
black
the empty car crashed
the empty car
empty car
empty car crashed
empty
the dirty dog fell
the dirty dog
dirty dog fell
dog fell
the dirty
the car smelled
car smelled
smelled
the red box
the red
red box
red
the red car
red car
the dog ran
dog ran
the car opened
car opened
opened
a hungry rat slept
a hungry rat
rat slept
hungry rat
hungry rat slept
the clean cat
clean cat
the car

Atomic Level Words

rat
window
car
a
dog
fell
slid
the
cat
clean
broke
dirty
box
ran
white
slept
crashed
hungry
died
black
empty
smelled
red
opened

Category Groups

|fell| |slept|
|slid| |crashed| |smelled| |opened|
|window| |car|
|clean| |car| |dirty|
|dirty| |window| |car| |clean|
|hungry| |rat| |window| |car| |dirty|
|fell| |ran| |slept| |died|
|dog| |car| |cat|
|clean| |car| |dirty| |white|
|white| |clean| |red|
|black| |white| |empty|
|red| |white|
|fell| |ran|
|rat| |dog| |cat| |rat| |dog| |rat| |dog| |cat| |dog| |cat| |rat| |dog| |dog|

Output Utterances

the empty car opened
a clean car smelled
the clean rat died
a empty car smelled
a dog ran
the window broke
the clean red box
a car opened
a empty car smelled
a empty car smelled
the car smelled
the hungry
a dog died
the white clean empty car crashed
the black car slid
a dog fell
the clean car crashed
the car crashed
the dog slept
the white dirty red car opened
a window broke
a car crashed
the hungry
the window broke
a cat died
the dirty dirty dirty dirty car smelled
a cat fell
a clean hungry
a cat fell
the dirty car smelled
the car crashed
the window broke
a rat fell
a red rat slept
a hungry
a clean hungry
a dog ran
a car smelled
a car slid
the dog ran
the clean rat fell
the dog slept
the dog ran
a rat ran
the red empty car crashed
a rat fell
the dirty rat ran
a clean white black car smelled
the dog fell
the black car crashed
the clean car crashed
the car crashed
a dog fell
a clean red car crashed
a dog ran
the dirty car opened
a red white car opened
the black car slid
a rat fell
a window broke
the dog slept
a box
a window broke
a dog slept
a dirty cat fell
the black car crashed
the black car opened
the cat ran
a empty car crashed
a clean hungry
the dog fell
a cat fell
the box
a hungry
a dog fell

Novel Output Utterances

the empty car opened
a clean car smelled
the clean rat died
a empty car smelled
a dog ran
the clean red box
a car opened
a empty car smelled
a empty car smelled
the hungry
a dog died
the white clean empty car crashed
the black car slid
a dog fell
the clean car crashed
the car crashed
the dog slept
the white dirty red car opened
a car crashed
the hungry
a cat died
the dirty dirty dirty dirty car smelled
a cat fell
a clean hungry
a cat fell
the dirty car smelled
the car crashed
a red rat slept
a clean hungry
a dog ran
a car smelled
a car slid
the clean rat fell
the dog slept
a rat ran
the red empty car crashed
the dirty rat ran
a clean white black car smelled
the dog fell
the clean car crashed
the car crashed
a dog fell
a clean red car crashed
a dog ran
the dirty car opened
a red white car opened
the black car slid
the dog slept
a box
a dog slept
a dirty cat fell
the black car opened
the cat ran
a empty car crashed
a clean hungry
the dog fell
a cat fell
the box
a dog fell

Matching - All: 0.797337
Matching - 1s: 0.373494
Novel Utterances: 0.786667
Grammatical Utterances: 0.840000
Average Grammaticality: 0.957917
Novel Grammatical Utterances: 0.626667
Average Novel Grammaticality: 0.946504

Non-random 100 Data - Run 40

Input Utterances

```
|the| |window|
|a| |clean| |cat| |ran|
|window|
|a| |clean| |rat| |died|
|cat|
|window|
|the| |window|
|the| |black| |car| |opened|
|the| |deep| |box| |broke|
|dog|
|rat|
|rat|
|the| |black| |car| |slid|
|a| |black| |box|
|the| |dog|
|cat|
|car|
|window|
|window|
|cat|
|the| |dog| |ran|
|a| |rat|
|a| |expensive| |box| |fell|
|the| |car|
|a| |dirty| |rat| |fell|
|rat|
|a| |expensive| |car| |smelled|
|a| |rat|
|the| |expensive| |box| |opened|
|the| |dirty| |rat| |died|
|the| |empty| |car|
|a| |white| |car|
|box|
|the| |dog|
|a| |clean| |dog|
|a| |black| |box|
|a| |white| |box| |opened|
|box|
|the| |car|
|a| |car|
|the| |window|
|a| |cat|
|the| |hungry| |cat|
|the| |empty| |car|
|a| |black| |cat|
|a| |car| |slid|
|a| |deep| |box| |fell|
|a| |deep| |box|
|car|
|cat|
|the| |dog| |ran|
|a| |box| |opened|
|a| |black| |box|
|cat|
|a| |expensive| |car| |crashed|
|the| |car|
|the| |empty| |box|
|rat|
|box|
|cat|
|a| |black| |rat|
|a| |window| |fell|
|a| |white| |box| |broke|
|a| |deep| |box| |slid|
|the| |hungry| |cat| |fell|
|window|
|the| |car| |smelled|
|car|
|rat|
|cat|
|the| |box| |slid|
|window|
|a| |clean| |dog|
|the| |black| |rat|
|a| |empty| |car|
|a| |cat| |fell|
|the| |expensive| |box| |slid|
|the| |dirty| |window|
|cat|
|a| |empty| |car| |smelled|
|a| |rat| |ran|
|the| |clean| |dog|
|the| |cat|
|a| |cat| |slept|
|the| |black| |dog| |ran|
|the| |dog|
|the| |rat| |fell|
|the| |empty| |car| |slid|
|a| |cat| |smelled|
|rat|
|the| |expensive| |window| |fell|
|box|
|cat|
|the| |window| |fell|
|the| |car| |smelled|
|the| |clean| |cat| |smelled|
|box|
|a| |cat|
|window|
|a| |dog| |ran|
```

Known Chunks

```
the window
a clean cat ran
window
a clean rat died
cat
the
the black car opened
black car opened
the deep box broke
deep box broke
dog
rat
the black car slid
black car slid
a black box
the dog
car
the dog ran
ran
dog ran
a rat
a
a expensive box fell
expensive box fell
the car
a dirty rat fell
dirty rat fell
a dirty
fell
a expensive car smelled
expensive car smelled
a expensive
smelled
the expensive box opened
expensive box opened
the dirty rat died
dirty rat died
the dirty
died
the empty car
the empty
empty car
empty
a white car
a white
white car
white
box
a clean dog
a clean
clean dog
clean
a black
black box
black
a white box opened
box opened
white box opened
opened
a car
a cat
the hungry cat
the hungry
hungry cat
hungry
a black cat
black cat
a car slid
slid
car slid
a deep box fell
a deep box
deep box
deep box fell
a deep
deep
a box opened
a box
a expensive car crashed
car crashed
expensive car crashed
expensive
crashed
the empty box
empty box
a black rat
black rat
a window fell
a window
window fell
a white box broke
box broke
white box broke
broke
a deep box slid
box slid
deep box slid
the hungry cat fell
cat fell
hungry cat fell
the car smelled
car smelled
the box slid
the box
the black rat
the black
a empty car
a empty
a cat fell
the expensive box slid
the expensive box
the expensive
expensive box
expensive box slid
the dirty window
dirty window
dirty
a empty car smelled
empty car smelled
a rat ran
rat ran
the clean dog
the clean
the cat
a cat slept
slept
cat slept
the black dog ran
the black dog
black dog
black dog ran
the rat fell
the rat
rat fell
the empty car slid
empty car slid
a cat smelled
cat smelled
the expensive window fell
the expensive window
expensive window
expensive window fell
the window fell
the clean cat smelled
the clean cat
```

clean cat
clean cat smelled
a dog ran
a dog

Atomic Level Words

window
cat
the
dog
rat
car
ran
a
fell
smelled
died
empty
white
box
clean
black
opened
hungry
slid
deep
expensive
crashed
broke
dirty
slept

Category Groups

|fell| |smelled| |slept|
|window| |cat| |rat| |box|
|dog| |rat|
|rat| |window| |cat| |dog| |box|
|empty| |black|
|hungry| |black|
|ran| |fell| |died|
|smelled| |slid| |crashed|
|white| |empty|
|deep| |empty| |black| |expensive|
|expensive| |deep|
|cat| |car| |car| |box| |car| |window| |cat| |rat| |box| |box| |car| |box| |box|
|cat| |car| |car| |box| |car|
|fell| |opened| |slid| |broke|
|window| |cat| |rat| |box| |cat| |car| |cat| |dog| |rat| |window| |cat| |rat| |box| |rat| |window| |cat| |rat| |box| |box| |car| |box| |box|

Output Utterances

the box slid
a cat fell
the box broke
a empty black empty empty rat ran
the white empty white clean white deep dog the box slid
a cat broke
the dog black empty empty black window slept
the cat ran
a dirty clean black empty cat smelled
the dirty deep deep dog fell
a window opened
the cat fell
a dog cat slid
a car crashed
the car fell
the cat ran
a empty hungry clean car slept
the white car fell
the rat ran
a dog slid
the window opened
the empty hungry hungry box slept
the clean expensive expensive black rat fell
the box smelled
a rat broke
the rat slept
the expensive empty dirty empty empty car slid
a expensive box opened
the cat slid
the rat ran
the window died
a empty box ran
a window slid
the cat smelled
a rat smelled
a cat slid
the deep dog cat broke
a car fell
a window died
the dog expensive white hungry cat died
a expensive black empty box slid
the box smelled
the hungry car broke
a expensive empty expensive box ran
the black expensive dog box smelled
the cat smelled
a rat fell
the rat broke
the dirty dog deep dirty dirty empty rat slid
the rat slid
a box crashed
the window opened
a black expensive box ran
the empty cat broke
a car opened
a empty empty white rat fell
the expensive empty box fell
the window broke
a car crashed
the rat fell
the expensive box opened
the cat fell
a deep dog empty expensive expensive box broke
a empty dog white window fell
the hungry clean empty rat fell
a car opened
a box ran
a empty rat slept
the hungry box slid
a box died
a white hungry rat slept
a box fell
the expensive cat slept
the dirty white hungry empty cat smelled
a dirty clean expensive cat broke
the dog cat ran
the deep box smelled
a expensive rat slept
the rat opened
a white black expensive white black empty black black window smelled
a cat opened
a dog rat ran
the hungry box died
the expensive car slid
a cat slept
the expensive window broke
the dog box opened
the deep black empty deep cat crashed
a box fell
the cat smelled
a dog rat smelled
the expensive empty expensive cat smelled
a expensive car smelled
a box crashed
a rat slept
a expensive rat smelled
the empty window smelled
the white clean hungry window died
a empty deep cat slid
a empty cat slept

Novel Output Utterances

the box broke
a empty black empty empty rat ran
the white empty white clean white deep dog the box slid
a cat broke
the dog black empty empty black window slept
the cat ran
a dirty clean black empty cat smelled
the dirty deep deep dog fell
a window opened
the cat fell
a dog cat slid
a car crashed
the car fell
the cat ran
a empty hungry clean car slept
the white car fell
the rat ran
a dog slid
the window opened
the empty hungry hungry box slept
the clean expensive expensive black rat fell
the box smelled
a rat broke
the rat slept
the expensive empty dirty empty empty car slid
a expensive box opened
the cat slid
the rat ran
the window died
a empty box ran
a window slid
the cat smelled
a rat smelled
a cat slid
the deep dog cat broke
a car fell
a window died
the dog expensive white hungry cat died
a expensive black empty box slid
the box smelled
the hungry car broke
a expensive empty expensive box ran
the black expensive dog box smelled
the cat smelled
a rat fell
the rat broke
the dirty dog deep dirty dirty empty rat slid
the rat slid
a box crashed
the window opened
a black expensive box ran
the empty cat broke
a car opened
a empty empty white rat fell
the expensive empty box fell
the window broke
a car crashed

```
the cat fell
a deep dog empty expensive expensive box broke
a empty dog white window fell
the hungry clean empty rat fell
a car opened
a box ran
a empty rat slept
the hungry box slid
a box died
a white hungry rat slept
a box fell
the expensive cat slept
the dirty white hungry empty cat smelled
a dirty clean expensive cat broke
the dog cat ran
the deep box smelled
a expensive rat slept
the rat opened
a white black expensive white black empty black black window smelled
a cat opened
a dog rat ran
the hungry box died
the expensive car slid
the expensive window broke
the dog box opened
the deep black empty deep cat crashed
a box fell
the cat smelled
a dog rat smelled
the expensive empty expensive cat smelled
a box crashed
a rat slept
a expensive rat smelled
the empty window smelled
the white clean hungry window died
a empty deep cat slid
a empty cat slept
```

Matching - All: 0.698225
Matching - 1s: 0.312715
Novel Utterances: 0.940000
Grammatical Utterances: 0.410000
Average Grammaticality: 0.722539
Novel Grammatical Utterances: 0.350000
Average Novel Grammaticality: 0.704829

Non-random 100 Data - Run 99
Input Utterances

```
|the| |white| |dog|
|a| |car|
|rat|
|rat|
|a| |dog| |died|
|window|
|the| |dirty| |rat|
|window|
|car|
|a| |red| |box|
|the| |expensive| |cat|
|a| |deep| |box| |opened|
|dog|
|the| |expensive| |window| |broke|
|a| |clean| |car| |smelled|
|rat|
|car|
|the| |empty| |box|
|a| |dirty| |window|
|a| |empty| |box|
|cat|
|the| |empty| |car| |slid|
|a| |black| |dog|
|car|
|rat|
|the| |box|
|dog|
|window|
|car|
|the| |car|
|rat|
|a| |car|
|the| |deep| |box|
|the| |empty| |box| |slid|
|a| |black| |car|
|a| |box|
|a| |window| |fell|
|the| |clean| |box| |fell|
|car|
|cat|
|the| |empty| |box|
|dog|
|box|
|box|
|rat|
|rat|
|the| |rat|
|the| |window|
|the| |red| |box|
|a| |deep| |box| |opened|
|the| |cat|
|a| |rat| |ran|
|the| |deep| |box| |smelled|
|the| |clean| |box| |smelled|
|a| |white| |cat| |slid|
|a| |expensive| |box|
|the| |dirty| |box|
|the| |deep| |box| |slid|
|the| |rat|
|the| |clean| |window|
|the| |rat| |slid|
|car|
|box|
|a| |cat|
|a| |window| |fell|
|car|
|window|
|car|
|the| |cat|
|a| |empty| |car| |slid|
|window|
|a| |empty| |box|
|rat|
|the| |hungry| |rat| |slept|
|the| |rat|
|the| |deep| |box| |slid|
|the| |empty| |car| |crashed|
|the| |dirty| |cat| |ran|
|dog|
|a| |cat|
|rat|
|rat|
|a| |hungry| |cat| |slept|
|the| |empty| |box|
|the| |box| |opened|
|the| |clean| |car| |slid|
|cat|
|a| |clean| |car| |smelled|
|a| |expensive| |window| |broke|
|a| |dirty| |cat| |smelled|
|the| |expensive| |box|
|the| |expensive| |cat| |ran|
|cat|
|the| |window|
|the| |dirty| |car| |smelled|
|the| |red| |box|
|box|
|car|
|a| |white| |box|
|dog|
```

Known Chunks

```
the white dog
a car
rat
a dog died
window
the dirty rat
the dirty
car
a red box
the expensive cat
a deep box opened
dog
the expensive window broke
the expensive
broke
a clean car smelled
a clean
smelled
the empty box
a dirty window
a dirty
a empty box
cat
the empty car slid
the empty
slid
a black dog
a black
the box
the car
the
a
the deep box
deep box
the empty box slid
box slid
empty box
empty box slid
empty
box
a black car
black car
black
a box
a window fell
window fell
fell
the clean box fell
the clean box
clean box
clean box fell
the clean
the rat
the window
the red box
the red
red box
red
opened
deep box opened
a deep
the cat
a rat ran
rat ran
ran
the deep box smelled
deep box smelled
the deep
the clean box smelled
box smelled
clean box smelled
clean
a white cat slid
a white cat
white cat
white cat slid
a white
a expensive box
a expensive
expensive box
expensive
the dirty box
```

dirty box
dirty
the deep box slid
deep box slid
deep
the clean window
clean window
the rat slid
rat slid
a cat
a window
a empty car slid
a empty car
empty car
empty car slid
car slid
a empty
the hungry rat slept
hungry rat slept
the hungry
slept
the empty car crashed
car crashed
crashed
empty car crashed
the dirty cat ran
the dirty cat
cat ran
dirty cat
dirty cat ran
a hungry cat slept
a hungry cat
hungry cat
hungry cat slept
a hungry
the box opened
box opened
the clean car slid
the clean car
clean car
clean car slid
a clean car
car smelled
clean car smelled
a expensive window broke
a expensive window
window broke
expensive window
expensive window broke
a dirty cat smelled
a dirty cat
cat smelled
dirty cat smelled
the expensive box
the expensive cat ran
expensive cat
expensive cat ran
the dirty car smelled
the dirty car
dirty car
dirty car smelled
a white box
white box
white

Atomic Level Words

rat
window
car
dog
broke
smelled
cat
slid
the
a
empty
box
black
fell
red
opened
ran
clean
expensive
dirty
deep
slept
crashed
white

Category Groups

|slid| |ran| |slept|
|broke| |fell|
|smelled| |slid| |crashed|
|smelled| |slid| |ran| |slept|
|window| |cat| |box|
|empty| |red| |clean| |expensive| |dirty|
|box| |window| |car| |cat|
|red| |empty| |expensive| |dirty|
|clean| |empty| |dirty|
|deep| |empty|
|empty| |clean| |expensive| |dirty|
|expensive| |empty| |dirty|
|white| |empty| |expensive| |dirty|
|car| |cat| |box| |rat| |car| |cat| |box| |window| |box| |box|
|smelled| |slid| |fell| |opened|
|car| |cat| |box| |rat| |car| |cat| |box| |car|
|window| |window| |box|

Output Utterances

a rat smelled
a white window crashed
the dog
a car smelled
the expensive deep smelled
a cat opened
the cat smelled
the white white dirty clean cat slid
the cat slid
a crashed
a clean black deep cat opened
the cat ran
the crashed
a smelled
a red black deep expensive red dirty rat smelled
the white rat crashed
a dog
the box slid
the dirty empty expensive cat ran
a crashed
the dog
a broke
the cat opened
a white empty expensive car crashed
a slid
the clean empty empty clean deep car opened
a clean box smelled
a slept
the box crashed
a smelled
a broke
a slept
a cat smelled
a red dirty cat smelled
a dirty empty white empty expensive cat smelled
a dog
the red empty empty deep crashed
the the dirty window deep crashed
the the window empty box ran
a empty window smelled
a black window car slid
the broke
a opened
a expensive car crashed
a crashed
the fell
a smelled
a box slid
a box fell
a empty red expensive empty box slid
the red rat slept
a the deep the rat slid
a dirty dirty box slept
the clean car ran
a opened
the the broke
a rat ran
a box crashed
a slid
a black empty box fell
the deep red car slid
the expensive clean car fell
the broke
a expensive clean deep smelled
the clean dirty window black clean window dirty white deep dirty dirty empty cat ran
a smelled
the slid
a black window opened
the crashed
the white deep dirty cat slid
the smelled
a crashed
the box crashed
a ran
a expensive rat slept
the smelled
a red window slid
the empty car smelled
the window cat crashed
the slid
the ran
a empty white deep red dirty clean red empty empty expensive empty dirty empty empty black deep clean dirty window box fell
the ran
the broke
a expensive empty window slid
the box opened
the broke
the slept
a window slept
a dirty clean rat smelled
a the empty box smelled
a ran
a clean car slept
a car slept
a box slept
a box smelled
a window expensive deep opened
the black expensive clean dirty dirty window slid
the the crashed
a deep opened

Novel Output Utterances

a rat smelled
a white window crashed
the dog
a car smelled
the expensive deep smelled
a cat opened
the cat smelled
the white white dirty clean cat slid
the cat slid
a crashed

a clean black deep cat opened
the cat ran
the crashed
a smelled
a red black deep expensive red dirty rat smelled
the white rat crashed
a dog
the box slid
the dirty empty expensive cat ran
a crashed
the dog
a broke
the cat opened
a white empty expensive car crashed
a slid
the clean empty empty clean deep car opened
a clean box smelled
a slept
the box crashed
a smelled
a broke
a slept
a cat smelled
a red dirty cat smelled
a dirty empty white empty expensive cat smelled
a dog
the red empty empty deep crashed
the the dirty window deep crashed
the the window empty box ran
a empty window smelled
a black window car slid
the broke
a opened
a expensive car crashed
a crashed
the fell
a smelled
a box slid
a box fell
a empty red expensive empty box slid
the red rat slept
a the deep the rat slid
a dirty dirty box slept
the clean car ran
a opened
the the broke
a box crashed
a slid
a black empty box fell
the deep red car slid
the expensive clean car fell
the broke
a expensive clean deep smelled
the clean dirty window black clean window dirty white deep dirty dirty empty cat ran
a smelled
the slid
a black window opened
the crashed
the white deep dirty cat slid
the smelled
a crashed
the box crashed
a ran
a expensive rat slept
the smelled
a red window slid
the empty car smelled
the window cat crashed
the slid
the ran
a empty white deep red dirty clean red empty empty expensive empty dirty empty empty black deep clean dirty window box fell
the ran
the broke
a expensive empty window slid
the broke
the slept
a window slept
a dirty clean rat smelled
a the empty box smelled
a ran
a clean car slept
a car slept
a box slept
a box smelled
a window expensive deep opened
the black expensive clean dirty dirty window slid
the the crashed
a deep opened

Matching - All: 0.671598
Matching - 1s: 0.274021
Novel Utterances: 0.980000
Grammatical Utterances: 0.190000
Average Grammaticality: 0.434945
Novel Grammatical Utterances: 0.170000
Average Novel Grammaticality: 0.423413

Non-random 100 Data - Run 53

Input Utterances

|cat|
|the| |clean| |car|
|a| |hungry| |dog| |slept|
|rat|
|car|
|the| |red| |box|
|a| |red| |box|
|cat|
|a| |white| |box|
|a| |expensive| |window|
|the| |dog|
|a| |window|
|the| |clean| |dog| |fell|
|the| |white| |rat| |slept|
|box|
|window|
|dog|
|dog|
|cat|
|the| |hungry| |dog|
|the| |dirty| |rat|
|the| |expensive| |window|
|car|
|the| |expensive| |cat|
|box|
|the| |expensive| |box| |smelled|
|the| |dirty| |rat| |slept|
|dog|
|a| |expensive| |cat| |smelled|
|a| |dog|
|the| |box|
|rat|
|box|
|a| |expensive| |window| |broke|
|the| |deep| |box| |slid|
|car|
|the| |empty| |box| |broke|
|the| |cat|
|box|
|a| |red| |box|
|a| |expensive| |box| |smelled|
|car|
|a| |car| |opened|
|cat|
|a| |hungry| |dog| |fell|
|the| |rat| |ran|
|window|
|cat|
|the| |window| |opened|
|the| |empty| |box| |broke|
|rat|
|box|
|cat|
|the| |dirty| |box| |broke|
|a| |empty| |car| |slid|
|cat|
|the| |clean| |dog| |smelled|
|the| |expensive| |window| |broke|
|the| |hungry| |rat|
|the| |white| |rat| |slid|
|car|
|rat|
|a| |rat| |slid|
|a| |car| |crashed|
|the| |deep| |box| |smelled|
|the| |expensive| |cat| |died|
|a| |car|
|cat|
|the| |clean| |box|
|a| |car| |slid|
|rat|
|the| |cat|
|the| |white| |cat| |slid|
|a| |car|
|car|
|a| |hungry| |cat|
|the| |red| |box| |slid|
|a| |window| |opened|
|the| |window| |fell|
|the| |clean| |dog|
|box|
|box|
|a| |empty| |car|
|a| |window| |fell|
|a| |box| |fell|
|car|
|a| |car|
|rat|
|a| |dirty| |car|
|a| |expensive| |cat| |fell|
|the| |window| |fell|
|window|
|dog|
|the| |empty| |car|
|a| |hungry| |dog| |smelled|
|box|
|the| |dirty| |rat| |ran|
|box|
|the| |clean| |window|
|the| |expensive| |box| |slid|

Known Chunks

cat
the clean car
a hungry dog slept
rat
car
the red box
a red box
a white box
a expensive window
the dog
a window
the clean dog fell
the white rat slept
the white
slept
box
window
dog
the hungry dog
the hungry
the dirty rat
the dirty
the expensive window
the expensive
the expensive cat
the expensive box smelled
box smelled
smelled

```
the dirty rat slept
rat slept
a expensive cat smelled
a expensive cat
a expensive
a dog
a
the box
the
a expensive window broke
broke
window broke
expensive window
expensive window broke
expensive
the deep box slid
deep box slid
the deep
slid
the empty box broke
the empty box
empty box
empty box broke
the empty
the cat
a red
red box
red
a expensive box smelled
a expensive box
expensive box
expensive box smelled
a car opened
car opened
opened
a hungry dog fell
hungry dog fell
a hungry
fell
the rat ran
rat ran
ran
the window opened
the window
window opened
box broke
empty
the dirty box broke
the dirty box
dirty box
dirty box broke
dirty
a empty car slid
a empty car
empty car
empty car slid
car slid
a empty
the clean dog smelled
the clean dog
clean dog
clean dog smelled
the clean
the expensive window broke
the hungry rat
hungry rat
hungry
the white rat slid
the white rat
rat slid
white rat
white rat slid
white
a rat slid
a rat
a car crashed
car crashed
crashed
the deep box smelled
the deep box
deep box
deep box smelled
deep
the expensive cat died
died
cat died
expensive cat
expensive cat died
a car
the clean box
clean box
clean
a car slid
the white cat slid
the white cat
cat slid
white cat
white cat slid
a hungry cat
hungry cat
the red box slid
red box slid
box slid
the red
a window opened
the window fell
window fell
a window fell
a box fell
a box
box fell
a dirty car
a dirty
dirty car
a expensive cat fell
cat fell
expensive cat fell
the empty car
a hungry dog smelled
a hungry dog
dog smelled
hungry dog
hungry dog smelled
the dirty rat ran
dirty rat
dirty rat ran
the clean window
clean window
the expensive box slid
the expensive box
expensive box slid
```

Atomic Level Words

```
cat
rat
car
slept
box
window
dog
smelled
a
the
broke
expensive
slid
red
opened
fell
ran
empty
dirty
hungry
white
crashed
deep
died
clean
```

Category Groups

```
|smelled| |slid| |fell| |died|
|slept| |slid| |ran|
|slid| |opened| |crashed|
|smelled| |broke| |slid| |fell|
|broke| |opened| |fell|
|smelled| |fell|
|rat| |car| |box|
|car| |rat| |box| |window|
|box| |rat| |window| |dog|
|window| |box|
|dog| |box| |window|
|expensive| |window| |red|
|red| |expensive|
|empty| |expensive| |dirty|
|hungry| |expensive|
|cat| |rat| |box| |window|
|rat| |cat| |box|
|window| |cat| |box|
|red| |expensive| |clean|
|deep| |expensive|
|cat| |rat| |car| |box| |car|
   |window| |car|
|cat| |box| |dog| |box| |window|
   |cat| |rat| |car| |box|
   |cat| |box| |window| |dog|
   |box| |window| |car|
   |window| |cat| |box|
   |window| |dog| |cat| |box|
   |dog| |cat| |box| |window|
   |dog|
```

Output Utterances

```
the white
a dog opened
a cat slept
a cat broke
the box broke
the box slept
the empty car ran
a empty empty box slid
a rat fell
a expensive box fell
the expensive box fell
a expensive cat smelled
a expensive dog smelled
the dirty window fell
a rat broke
the cat smelled
the window crashed
a window fell
a window fell
the deep
the expensive cat ran
a cat opened
a cat fell
a window broke
the rat slid
a dog fell
the clean car slid
the car fell
the box slid
the expensive rat slept
a expensive box fell
the window broke
a box smelled
the expensive box fell
a white
the window died
a white
the rat died
the rat ran
the cat fell
a cat fell
the car opened
the dog smelled
a dog died
a white
a white
a box slept
the expensive box crashed
a box crashed
the expensive box smelled
the rat died
a window crashed
a window fell
a expensive cat crashed
the window crashed
```

```
the expensive cat opened
the car fell
a clean deep
a expensive box smelled
a cat smelled
the box slid
a box crashed
a clean deep
the cat fell
a deep
the hungry window broke
the dog smelled
a empty car opened
the car broke
a dog fell
a expensive box slid
the empty expensive box fell
a box broke
a rat crashed
the box fell
a expensive box slept
a clean white
the white
a hungry expensive cat slid
a expensive box opened
the rat died
a white
the window crashed
a cat fell
the box smelled
the car fell
the box ran
a expensive box ran
the car ran
the rat crashed
the box smelled
the expensive box fell
the dirty expensive car slid
a car ran
the window broke
the expensive box fell
the rat smelled
a dirty dog smelled
the window smelled
the box fell
```

Novel Output Utterances

```
a dog opened
a cat slept
a cat broke
the box broke
the box slept
the empty car ran
a empty empty box slid
a rat fell
a expensive box fell
the expensive box fell
a expensive dog smelled
the dirty window fell
a rat broke
the cat smelled
the window crashed
the expensive cat ran
a cat opened
a cat fell
a window broke
the rat slid
a dog fell
the clean car slid
the car fell
the box slid
the expensive rat slept
a expensive box fell
the window broke
a box smelled
the expensive box fell
a white
the window died
a white
the rat died
the cat fell
a cat fell
the car opened
the dog smelled
a dog died
a white
a white
a box slept
the expensive box crashed
a box crashed
the rat died
a window crashed
a expensive cat crashed
the window crashed
the expensive cat opened
the car fell
a clean deep
a cat smelled
the box slid
a box crashed
a clean deep
the cat fell
a deep
the hungry window broke
the dog smelled
a empty car opened
the car broke
a dog fell
a expensive box slid
the empty expensive box fell
a box broke
a rat crashed
the box fell
a expensive box slept
a clean white
a hungry expensive cat slid
a expensive box opened
the rat died
a white
the window crashed
a cat fell
the box smelled
the car fell
the box ran
a expensive box ran
the car ran
the rat crashed
the box smelled
the expensive box fell
the dirty expensive car slid
a car ran
the window broke
the expensive box fell
the rat smelled
a dirty dog smelled
the window smelled
the box fell
```

Matching - All: 0.847633
Matching - 1s: 0.475904
Novel Utterances: 0.900000
Grammatical Utterances: 0.620000
Average Grammaticality: 0.830000
Novel Grammatical Utterances: 0.520000
Average Novel Grammaticality: 0.811111

References:

Batali, J. (1998). Computational simulations of the emergence of grammar. in J. Hurford, M Studdert-Kennedy, and C. Knight (eds.) *Approaches to the Evolution of Language: Social and Cognitive Bases.* 405-426. Cambridge University Press: Cambridge, GB.

Brooks, R. (1991). Intelligence without representation. *Artificial Intelligence.* 47, 139–159.

Chalmers, David. (1990). Syntactic Transformations on Distributed Representations. *Connection Science,* Vol. 2, Nos 1 & 2, 53-62.

Chomsky, N. (1959). A review of B.F. Skinner's Verbal Behaviour. *Language*. 35 (1), 26-58.

Chomsky, N. (1965). *Aspects of the theory of syntax*. Cambridge, MA: MIT Press.

Clark, A. (1998). *Being There: Putting Brain, Body, and World Together Again*. MIT Press: Cambridge, MA.

Colunga-Leal, E. and M. Gasser. (ms). A model of the acquisition of spatial-relation concepts and words.

Cook, V.J. and M. Newson (1996). *Chomsky's Universal Grammar*. Blackwell: Cambridge, MA.

Damasio, A. and H. Damasio. (1994). Cortical systems for retrieval of concrete knowledge: The convergence zone framework. In: Koch, C. and L. Davis (Eds.). *Large-Scale Neuronal Theories of the Brain*. Cambridge, MA: MIT Press.

Elman, Jeffery. (1995). Language as a dynamical system. In: Port et al.

Elman, J, E. Bates, M. Johnson, A. Karmiloff-Smith, D. Parisi, and K. Plunkett. (1998). *Rethinking Innateness.* MIT Press: Cambridge, MA.

Hebb, Donald. (1949). *The Organization of Behavior*. New York, NY: Wiley.

Hock, H. and B. Joseph. (1996). *Language History, Language Change, and Language Relationship*. Mouton de Gruyter: New York, NY.

Hofstadter, D. (forthcoming). Analogy as the core of cognition. In K.J. Holyoak, D. Gentner, and B. Kokinov (eds.) *The Analogical Mind: Perspectives from Cognitive Science*. MIT Press: Cambridge, MA.

Holland, John. (1975). *Adaption in Natural and Artificial Systems.* Ann Arbor, MI: University of Michigan Press.

Holland, J. (1998). *Emergence*. Addison-Wesley: Reading, MA.

Johnson, J. and E. L. Newport. (1989). Critical period effects in second language learning: The influence of mturational state on the acquisition of English as a second language. *Cognitive Psychology, 21*, 60-99.

Kelso, J.A.S. (1998). *Dynamic Patterns*. MIT Press: Cambridge, MA.

Kirby, S., and J. Hurford. (1997). Learning, culture, and evolution in the origin of linguistic constraints. in P. Husbands and H. Harvey (eds.) *Fourth European Conference in Artificial Life*. 493-502. MIT Press: Cambridge, MA.

Mitchell, Melanie. (1996). *An Introduction to Genetic Algorithms.* Cambridge, MA: MIT Press.

Newport, E. L. (1990). Maturational constraints on language learning. *Cognitive Science,* 14, 11-29.

O'Grady, W. (1997). *Syntactic Development*. University of Chicago Press: Chicago, IL.

Port, R. and J. Dalby. (1982). Consonant/vowel ratio as a cue for voicing in English. *Perception and Psychophysics.* 32 (2), 141-152.

Port, R. and T. van Gelder. (1995). *Mind as Motion: Explorations in the Dynamics of Cognition*. Cambridge, MA: MIT Press.

Redington, M. and N. Chater. (1998). Connectionist and Statistical Approaches to Language Acquisition: A Distributional Perspective. *Language and Cognitive Processes*. 13 (2/3), 129-191.

Redington, M., N. Chater, and S. Finch. (1998). Distributional information: a powerful cue for acquiring syntactic categories. *Cognitive Science*. 22(4), 425-469.

Rovee-Collier, C. (1991). The "memory system" of prelinguistic infants. *Annals of the New York Academy of Sciences,* 608, 517-536.

Rumelhart, David. (1997). The architecture of mind: A connectionist approach. In: Haugeland, John (Ed.) *Mind Design II*. Cambridge, MA: MIT Press.

Rumelhart, David and D. Zipser. (1985). Feature discovery by competitive learning. In: Rumelhart, D. and J. McClelland (Eds.). *Parallel Distributed Processes: Explorations in the Microstructure of Cognition, Vol 1.* Cambridge, MA: MIT Press.

Sharma, J., A. Angelucci, and M. Sur. (2000). Induction of visual orientation modules in auditory cortex. *Nature*. 404 - 6780, 841 - 847.

Skinner, B. F. (1957). *Verbal Behavior*. Appleton-Century-Crofts: New York, NY.

Thelen, Ester. (1995). Time-scale dynamics and the development of an embodied cognition. In: Port et al.

Thelen, E., and L. Smith (1994). *A Dynamic Systems Approach to the Development of Cognition and Action.* MIT Press: Cambridge, MA.

Vogt, P. (2000). *Lexicon Grounding on Mobile Robots*. Unpublished PhD Thesis, Vrije Universiteit Brussel.