Generating English Language Based On Formal Grammars: Literature Review

David Perryman

December 11, 2005

Contents

Chapter 1

Literature Review

1.1 Identification of Sources

English language generation requires research into several different areas of theory about the way in which language is constructed, and about the way in which language can be represented using logical structures. These structures can then be represented in a computer, and the language produced.

The first area to research is the study of language from a linguistics viewpoint. When learning English, or indeed any other language, what are the key features that have to be grasped? It is important to understand the fundamental facts about languages before attempting to represent them mathematically. Finding out the different ways that linguists break down a language, will be valuable when trying to represent a language using mathematical structures.

The next area to research are the logical structures that may have the power to represent a language. Which are the most suitable out of the many different mathematical theories about the nature of language? The study of the different types of grammar is essential to the construction of a system to generate language, because grammars describe languages (in the mathematical sense).

It is also important to look at existing systems, which have attempted to perform similar tasks. What approaches have been taken, and how successful have these approaches proved? The analysis of other systems may help to

clarify the strengths and weaknesses of the mathematics that they are based upon, and therefore show which theories work better when put into an active environment.

1.2 Study of the English Language

"Knowing grammar, and 'knowing about' grammar are two different things" Newby [1987]

Most people implicitly know their native language, but actually knowing about how that language is constructed is a different thing, as if the language is automatically acquired, but the brain conceals the complexities of how it is constructed. When studying English, an important fact to establish is that there are many different forms of English, different dialects, accents and styles throughout the world. However all these variations of the language must share a common core of some kind, otherwise communication between people of different regions would be almost impossible. It is this "Standard English" that is to be studied, as it contains the fundamentals of the language.

The building block of any (western) language is its alphabet, the letters that are used to construct all the more complicated structures of the language. These are also called phonemes, the basic sounds of the language. The study of how phonemes are combined into words is outside the scope of this project, but it is significant to acknowledge their existence.

1.2.1 Words and Morphemes

Morphemes are combinations of one or more letters, and words are combinations of one or more morphemes.

(1) Play
(2) Play +s
(3) Play +ing
(4) Play +ful +ly

In these examples play is a morpheme, but also a word, play is therefore free morpheme (as opposed to a bound morpheme), as it can exist alone. In (2) an 's' is added to the end of play to get plays, the application of this morpheme to play, changes it from the verb infinitive to the 3rd person singular form. Adding the morpheme 'ing' in (3) changes the verb to the imperfect form, and in (4), two morphemes are applied, and the result is an adverb, changing the type of word entirely.

Morphology is clearly an important part of the study of language, as it can change the meaning, and tense of words. Even if the correct words are in a sentence, in the correct order, if the wrong morphemes are applied to words, then the sentence will not be grammatically correct. However there are some clear rules that can be applied to words in order to alter them in the desired manner, but there are many different irregularities in the English language, so they will have to be handled carefully.

1.2.2 Types of Words

There are 10 different types of words as identified in Newby [1987]:

Nouns, Verbs, Adjectives, Adverbs, Articles, Determiners, Pronouns, Prepositions, Conjunctions, Interjections.

All of these words must be considered when constructing a sentence, but the most important words are the verbs and nouns. The actions within a sentence, and the objects that the actions are being performed upon. These elements form the underlying structure of the sentence, and a lot of other words fall into place once the verbs and nouns are identified.

Verbs & Auxiliaries

There are four main forms of English verbs as identified by Zandvoort [1975]:

a) The stem - infinitive
b) The stem +ing - imperfect
c) The stem +sibilant (usually 's') - 3rd person singular
d) The stem +dental (usually 'ed') - past tense

Most verbs in English have these different forms; there are some that are irregular, these will have to be handled explicitly. As well as these regular type of verbs, there is also a class of verbs which are referred to as auxiliary verbs. These are used to change the mood, or the tense of other verbs in the same sentence Zandvoort [1975]. The set of auxiliary verbs can also be broken down further, in Lasnik [2000] a list of 'modal' auxiliaries is given:

may,might,will,would,can.could,must,shall,should

The verbs "to have", and "to be", are treated as special cases, as they can appear in addition to any modal auxiliary in a sentence. As can be seen from the examples (5-8) up to 3 auxiliary verbs can be in a sentence at any one time.

(5) He slept
(6) He might sleep
(7) He might have slept
(8) He might have been sleeping

The order of these auxiliaries is fixed to "modal-have-be" Lasnik [2000], but any one of them can be removed, and the sentence is still valid. It is important to see that the addition of the auxiliary verbs within the sentences changes the form of the verb that they are affecting. This change will be discussed further later.

Nouns & Pronouns

There are two different types of noun, common nouns, and proper nouns Bach [1974]. Common nouns require that a determiner, or article, be placed before them; to make any sense, and proper nouns can be used alone.

(9) The monkey
(10) Some monkeys
(11) Fred

In these examples, monkey is a common noun; it is not valid to say, "Monkey climbed the tree". However Fred is a proper noun; it is valid to say, "Fred climbed the tree". This is a relatively trivial problem to solve when

generating the language, but it does add complexity to the eventual grammar.

Pronouns, as identified by Zandvoort [1975] have three different forms, first person (I), second person (you, they), and third person (he, she, it). Pronouns are used in place of nouns, when talking about an object that is already defined. When using a pronoun, the Although the sentence would be valid, unless it is clear who is being referred to, then using a pronoun would cause ambiguity and the sentence would not make sense.

Adjectives & Adverbs

Adjectives are words, which are usually used to describe a common noun that follows them. Adjectives are inserted between the article and the noun and, as shown by the examples 12-14, any number of adverbs can be used at the same time; there is theoretically no limit. However sentences become unreadable if too many are used.

(12) The red bus
(13) The bright red bus
(14) The big bright red bus

It is possible to produce some bizarre sentences if random insertion of adjectives is applied, because the adjectives are closely related to the noun they are describing. When adding adjectives into a sentence it will be necessary to ensure that they are linked to the object that they are describing in some way.

Adverbs are similar to adjectives, usually formed by adding the morpheme 'ly' to the end Aarts [1998] of an adjective.

"Adverbs are used to modify a verb, an adjective, or another adverb" Aarts [1998]

Adverbs usually apply to verbs, and therefore when choosing which adverbs to insert, it is important that they relate to the verb correctly, otherwise the sentence will not make any sense.

Phrase Type	Head	Example
Noun Phrase	Noun	the children in class 5
Verb Phrase	Verb	play the piano
Adjective Phrase	Adjective	delighted to meet you
Adverb Phrase	Adverb	very quickly
Prepositional Phrase	Preposition	in the garden

table 2.3.1 Aarts [1998]

Remaining word types

The remaining types of word that have not been discussed are prepositions, conjunctions and interjections. Prepositions are words that fit in-between the verb and the noun in a sentence like to/out/on Zandvoort [1975]. These words are linked to the verb that precedes them, and only prepositions that agree with the verb can be used in a sentence. Interjections are words like oh/ah, which can occur in many different places, they are not very important to this project, as they do not form part of the structure of a sentence. Conjunctions are words like and/but, and can be used to join different parts of a sentence together these will be discussed later.

1.2.3 Phrases

Phrases are elementary collections of words; there are several different types of phrase, based around the different word types:

Noun phrase, Verb phrase, Adjective Phrase, Adverb Phrase, Prepositional Phrase Zandvoort [1975]

Each of the phrases has a "head" word (apart from a prepositional phrase), which is the main word in the phrase, and the rest of the phrase is based around that word. An elementary noun phrase consists of just the noun, with article, and optional adjectives. An elementary verb phrase consists of the verb, any auxiliary verbs, and possibly a noun phrase, or prepositional phrase that follows it. Table 2.3.1 shows these different types of phrases, and gives examples of them.

The structure of phrases can be quite complicated, and one of the aims of this project is to produce a grammar, which can adequately define this

structure.

1.2.4 Clauses

Clauses are formed from one or more phrases, and form more complex statements. A clause must contain a verb phrase. The other elements of a clause are outlined in Newby [1987] as:

Subject - Noun Phrase
Verb - Verb Phrase
Direct Object - Noun Phrase
Indirect Object - Noun Phrase
Subject Compliment - Noun Phrase / Adjective Phrase
Object Compliment - Noun Phrase / Adjective Phrase
Adverbial - Adverb Phrase / Prepositional Phrase

The verb phrase is extensive (which means the subject does something to the object) or intensive (which means that the subject is the same entity as the object). Some verbs are intransitive, which means that there is no object in the clause. The Subject and Object compliments are phrases, which are related to the subject and object respectively, and inform the user about what, or who the entity. The adverbial phrase is an additional phrase usually giving extra information about when or where the clause is taking place.

1.2.5 Sentences

Sentences are formed from collections of clauses, there are two main ways in which sentences can be formed, using co-ordination, and using subordination Traugott [1972]. Co-ordination of clauses is joining two sentences together using a conjunctive word, such as and/but/however. This is a very simple way of forming a more complicated sentence from two clauses, but the clauses have to be linked in some way (semantically), or the conjunction between them will not make sense. When using conjunction it is possible to use pronouns in the second clause, because usually the subject of the sentence will be defined within the first clause.
The other way to form sentences is by subordination of clauses; this is where another clause takes the place of the noun phrases, or adverbial, within a clause. It is possible to make many different, and complicated sentences in

this manner.

There are four main types of sentences that are identified in Newby [1987]:

Statements, Commands, Exclamations, Questions

Statements are used to convey information of some kind; they state facts. They are the most common type of sentence, and therefore will be the focus of study for most of this project. Commands are used to tell someone what to do, and typically start with the verb, exclamations are usually said by a person, and are not necessarily very well formed. The Question, or interrogative class of sentences are quite an important class of sentences, there are four different kinds of interrogative sentences according to Aarts [1998]:

Yes/no, Alternative, Wh-, Tag Questions

Yes/no questions require a response of yes or no, they are formed by switching the noun and verb at the front of the sentence:

(15) This is your book
(16) Is this your book?
(17) Is this your book, or hers?
(18) This is your book, isn't it?

This is a very simple way of creating a question from a statement style of sentence. Alternative questions, give a list of options for the answer (17). Wh- style questions are sentences that start with what/when/why/whereTag questions are questions formed by adding a tag to the end of a statement, which questions its validity, therefore making the whole sentence a question (18).

1.3 Study of Syntax

"Syntax is the study of the principals and processes by which sentences are constructed in particular languages" Chomsky [1957]

Syntax is a very broad topic, it covers all possible studies into the gram-

mars of languages. This includes looking for grammars to languages outside of linguistics, to explain other phenomena, which may have an underlying mathematical structure. This study attempts to find a formal method to define a set of sentences that are in a language, and therefore to exclude, or reject, the sentences that are not in that language.

1.3.1 Statistical Analysis

One plausible approach to solving this problem is statistical analysis; that is calculating the probability of each word appearing in a sentence, and working out what chance a word has of following another word. This kind of approach is outlined in Harris [1982], where a language is defined as a set of sentences, and each word in the language has a chance of appearing in each sentence. Some words may have zero chance of appearing in a sentence if other words are already present, and some words may have a 100% chance of appearing next in a sentence if another word is there. To construct such a system would require gathering information about the valid sentences of a language, and then performing some analysis of those sentences to calculate the statistics of the system. However this approach is dismissed in Chomsky [1957], using the following examples:

(19) Colourless green ideas sleep furiously
(20) Furiously sleep ideas green colourless

It is argued that because English is an infinite language, it is impossible to analyse every possible sentence, and that therefore the chance of generating a sentence like (19) is the same as (20). All of the words are the same, but (19) is grammatically correct, and in some far-fetched reality could be a sentence. In Chomsky [1957] it is argued that a statistical approach replaces the very slim chance of (19) with zero chance, which is incorrect (excluding a sentence that is valid), therefore invalidating the approach.

1.3.2 Finite State Machine

Another approach to producing sentences is to use a finite state machine, where words are produced as the machine moves from one state to the next. The machine would have a starting state, and one or more end states. Languages that are generated by finite state machines are called finite state

languages, and can be enhanced by adding a probability of following each path of the machine, this creates a "finite state Markov process".

However this approach is also excluded from being powerful enough to produce complicated languages in Chomsky [1957] because it can't handle applying more than 1 rule at any one instance. Therefore it is not powerful enough to describe natural languages.

An extension to the finite state machine method for language processing is the augmented transition network. In this approach, additional tests and checks are performed before progressing from one node to another. This greatly increases the power of the model, and is useful for parsing text, and recognising grammatical correctness. However it also lacks the ability to apply multiple rules at the same time, and does not allow for transformations upon the produced text, because there is no parse tree generated.

1.3.3 Phrase Structure Grammars

A phrase structure grammar, also known as a context free (CF) grammar, is defined in Gross [1970] by:

1) A finite terminal vocabulary
2) A finite auxiliary vocabulary
3) An axiom (start symbol in the auxiliary vocabulary)
4) A finite number of rules from the auxiliary set to the auxiliary + terminal set

A phrase structure grammar is defined by having a set of symbols (in the case of a natural language these are mainly the words), and one of these symbols is the start symbol. The provided rules are applied, one at a time, transforming the start symbol into different sets of the symbols. The rules are applied until it is not possible to apply any more rules (when all the symbols that are remaining are terminal symbols).

Phrase structure grammars are very powerful, and capable of producing many different derivations, depending on the set of rules provided. It is helpful to be able to display derivations in a tree format, so that it is clear how the result was achieved.

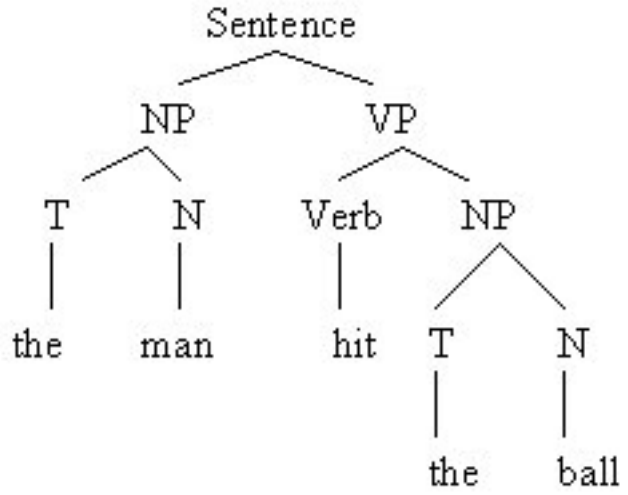

Figure 1.3.0 Example output from a phrase structure grammar

Although very powerful, CF grammars are also limited, because they are unable to handle linked structures. The best demonstration of this is returning to the morphology of auxiliary verbs. In Lasnik [2000] it is shown that "to have" and "+en" enter the sentence together, but the +en (which is a past morpheme) is attached to the next verb in the sentence (after the auxikiary). Furthermore when adding the verb "to be" into sentence, the morpheme "+ing" enters as well, attached to the verb that follows "to be".

"This represents a "cross-serial dependency". Phrase Structure rules can't in general deal with these dependencies." Lasnik [2000]

If phrase structure grammars can't deal with cross serial dependencies, then a different device is needed to produce such features as can be observed in the auxiliary verbs.

1.3.4 Transformational Grammars

A transformational grammar is a phrase structure grammar, with another set of rules that are applied to the terminal string that is produced. These new rules (called transformations) alter the tree that is produced by the phrase

grammar, and create new derivations from it. This is a powerful concept because it allows the cross dependencies to be produced by re-arranging the tree after the sentence has been produced Lasnik [2000]. Transformational grammars also allow the phrase grammar to become less complicated, because some of the rules that would have been in the production rules of the phrase grammar can be represented more easily as transformations.

(21) The dog is barking
(22) The dog barked

Examples 21-22 are a demonstration of a change that can be made when using a transformational grammar instead of a phrase structure grammar. In a phrase structure grammar, both these sentences would be different representations, and be produced by a slightly different set of production rules. However in a transformational grammar both these sentences can be produced from the same underlying phrase structure representation. If 21 was generated by the phrase structure, then a transformation could be defined to change the tense of the statement, and generate 22. These ideas are outlined in Chomsky [1957].

However these sentences seem linked more fundamentally than just being transformations of the same phrase grammar, they are both part of a family of sentences that contain the concept of a dog, and it barking. The underlying structure to the sentences is referred to as the "deep structure", it relates to the meaning of the sentence, and although it is unpronounceable it is theoretically there Fowler [1971]. There is a relationship between the deep structure and the semantics of the produced sentence Bach [1974], however it is difficult to capture, and not part of this project. The final output of the transformational grammar is the "surface structure", this is the pronounceable part of the sentence, and the aim of the production, but it contains both semantic and syntactic content, not pure semantics.

A transformational rule contains a structural analysis, and a structural change. To be able alter the structure of a production of the phrase grammar, some structural analysis must be performed on the tree that is produced. This is a pattern matching exercise, as outlined by Lasnik [2000].

(23) SA: X - en - V - Y SC: X1 - X3 - X2 # - X4

The example 22 is a transformational rule which moves an 'en' to the other side of a verb, and then binds it. The program has to look for the pattern X - en - V - Y, where X and Y are any term, en is the morpheme en (which would be added with the 'to have' auxiliary) and V, which is a verb. Once this pattern has been found, the 2nd and 3rd nodes are switched, and a word boundary placed after the 3rd node. This binds the 'en' morpheme to the verb.

There are four elementary operations that a transformation can perform on the tree of a phrase structure grammar, which are combined to form more complicated operations. While performing these changes, the transformation should try to preserve as much of the structural information as possible, so that as much can be known about the phrase structure derivation as possible. The four rules, as given by Lasnik [2000], are:

a) Adjunction of one term to another (left or right)
b) Deletion of a term, or sequence of terms
c) Adjunction of new material (that wasn't in the structure before)
d) Permutation (changing the order of two items)

Using these four alterations it is possible to do numerous amounts of transformations, some of which are outlined in Chomsky [1957]. Some transformations are obligatory (the rule must be applied if the structural analysis is met), and some are optional. The fact that it is necessary to alter the tree to produce the final sentence means that any program designed to use a transformational grammar will have to construct a tree structure (created by the phrase structure grammar), which can be passed to the transformational rules, and manipulated to form the new sentences.

1.3.5 Lexical Insertion

When building up the phrase structure a lot of work has to be done that is based upon the verb that is being used. Because some verbs are transitive, and others intransitive, it means that some verbs require a noun to follow (be the object in the sentence) and others explicitly do not allow a noun following. Furthermore the nouns that are chosen to occupy the subject, and object have to be of the correct type for the sentence to make sense Thomas

[1974].

(24) The party was fantastic
(25) The fantastic was party

Clearly 25 is not a correctly formed sentence, because the verb to be, requires a concrete subject, which is a noun that is physical object. Nouns can be classified into many different categories, things that are concrete/non-concrete, animate/inanimate, animals/humans, and many more besides. The best way to identify which nouns are appropriate is for the verb to specify when forms are acceptable, somewhere in its definition Fowler [1971].

If a verb specifies that must be followed by a concrete noun, then that also means it can be followed by any objects that are sub categories of concrete, such as animal/human, animate/inanimate. This means that when selecting nouns to fill the object, the ideal structure is some sort of tree, where everything below the concrete node, satisfies the condition of concrete. It is possible that a noun might satisfy several different categories at the same time, and then may have to be added to more than one of the nodes, if they are not sub-categories of each other. The process of selecting, and inserting the correct noun into the sentence is called lexical insertion. However if a verb specifies that it does not allow a noun to follow it, then this will have to be taken into account within the phrase structure grammar, before the lexical insertion takes place.

It may also be necessary for verbs to specify which prepositions (if any) are allowed (ore required) to follow it, because if an incorrect preposition follows the verb, then the sentence is incorrect. This could be achieved in a similar way, with the rules being taken into account as part of the phrase structure grammar. The preposition used, may affect the type of noun that is needed at the lexical insertion phase.

1.3.6 The Minimalist Program

Another slightly different approach to the problem of generating sentences is the minimalist program. This is essentially a new look at the underlying structure of sentences from the semantic, and syntactic viewpoint. It was designed because of the complexity that had developed in the traditional

approach, and to attempt to model the pattern of human thought more accurately Radford [2004].

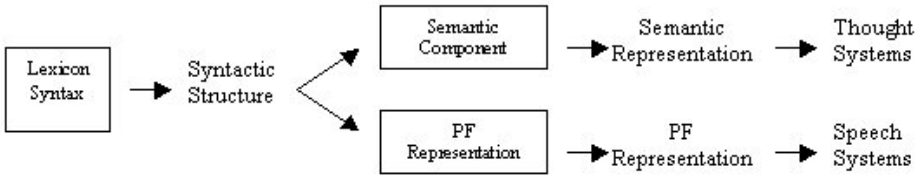

Figure 1.3.1 diagrammatic representation of the minimalist program, Radford [2005]

The derivation begins with the numeration of the key objects that are going to be talked about in the sentence (e.g. the verbs and nouns); a structure is produced containing these objects. At some point in the process there is a split between the LF (Logical Form) and the PF (Phonetic Form). The logical form contains all of the semantic information, and the phonetic form contains all of the phonetic information. The point at which this split occurs is sometimes called 'spell out' Cook [1996].

The minimalist program attempts to break down the problem into smaller, more easily solved problems, and has been developed based upon the way that children learn language. One of Chomsky's aims as he was working on the minimalist program, was to try and create a "universal grammar" one that defined why a language was learnable, and linked all forms of human grammars Radford [2004]. This means that some of the concepts are more general than just the study of English, and that the minimalist program is more closely linked to psychology, and linguistic theory.

1.3.7 Word Grammar

Word grammar is a grammar that does not use phrase structure as its base; it is based entirely upon dependencies between words in a sentence. Each word that can appear in a sentence is analysed, and the dependencies of that word calculated. For any particular word to appear in a sentence, all the dependencies associated with the word must be satisfied. This may take several passes to achieve, because the changes made to satisfy one word might invalidate another word Hudson [2005].

This technique is significant because it does not use phrase structure, which is what the vast majority of language studies are based upon, and also because it does not use a 'surface structure' and 'deep structure' system. Instead the collection of words is progressively parsed until all the dependencies are satisfied. This takes the representation from the semantic to the syntactic. The fact that this approach is successful means that phrases are not fundamentally tied to their phrase structure definitions, and that flexibility outside of phrase structure is possible. However this structure does mean creating full definitions for all the words that are used, which involves quite a large and complicated data store.

1.3.8 Tree Adjoining Grammar

Another grammatical technique is called the tree adjoining grammar; this builds up a language based upon trees, which are generated from a phrase grammar. A set of initial trees, and a set of auxiliary trees are combined using substitution and adjunction, to form derived trees, which are in the language Joshi [1997]. Using this approach, the language is broken down into smaller phrases, which are combined using the substitution and adjunction to form more complex sentences. This approach can be used to create a model for English (it is used in Cavazza [2005]), and is not very different from a transformational grammar. However it lacks some of the expressiveness of the transformational grammar, because the primary operations are substitutions and adjunctions, where as transformations can manipulate the structure of the nodes in more complex ways.

1.4 Study of Existing Systems

There are several different types of system that use grammars to analyse the structure of language, apart from simply generating it. Programs such as Microsoft Word have grammar checkers built into them, which analyse the text written, and attempt to work out if it is a valid sentence. Translation programs attempt to translate text from one natural language to another, which involves reading in the input text, performing some semantic analysis, and reproducing it in a different syntax. There are numerous sentence gen-

erator programs available on the Internet, most of limited functionality, but there are several approaches taken, and examining them is appropriate.

When trying to generate linked sentences, and groups of sentences that are all based around the same area, the main source of information is interactive computer games. Some new multiplayer games have interactive characters that are entirely AI driven, and respond based upon their relationships with the other characters within the environment. This type of game is where most of the current development into sentence production is happening, because of the marketing value of such games.

1.4.1 Grammar Checkers

"The Link Grammar Parser is a syntactic parser of English, based on link grammar, an original theory of English syntax." Temperley [2005]

One of the grammar checkers that is available on the Internet is the 'link grammar parser'; it uses a method of textual analysis called link parsing.

"Think of words as blocks with connectors coming out. There are different types of connectors; connectors may also point to the right or to the left. A left-pointing connector connects with a right-pointing connector of the same type on another word." Temperley [2005]

Essentially this method analyses the text, and matches the words with words from the dictionary. The words that are in the dictionary have rules attached to them, which describe what 'links' to other words they are allowed, or are required, to have. This approach seems to work quite well, but is probably not applicable to text generation. For each word that is in the dictionary, the rules have to be explicitly defined, which means that the system is not easily extensible. This system also fails to take into account the links between the different forms of verbs, and of morphology (each different word is separately defined). Not allowing for concepts like these would limit the power of a text generation system, and result in a lot of coding to produce all the links required. However using a system such as this could be useful to check the output of the system, to verify if the sentences produced are in correct English.

In Naber [2003] there is a description of a 'rule based' grammar checker, which is a grammar checker that breaks the text into tags, which have a particular word type, and checks those tags against a set of rules. Naber states:

"It turns out there are basically three ways to implement a grammar checker.
Syntax-based checking
Statistics-based checking
Rule-based checking" Naber [2003]

Syntax based checking involves parsing the whole text, and comparing each sentence against a grammar, if the text does not match the grammar, then it is incorrect. The main drawback of this approach is that it requires a full grammar of the language to be written. Statistics based checking looks at the text, and breaks it into simple phrases, assigning each word in the phrase with a word type (e.g. noun/verb). It uses statistics to calculate the likelihood of the words appearing in the order that they have been written, if it is a very low likelihood then the chances are that an error has been made. The main problem with this approach is that sometimes, unlikely sentences are grammatically correct, and sometimes likely looking sentences are in fact incorrect.

Rule based checking tags each word in the text with a word type, and compares the sequences of word types against error rules that is contains. If it matches one of the rules, then there is an error. This means that a grammar does not have to be defined to cover the whole language, and also extra rules can easily be added to account for previously undefined errors.

The concept of holding error rules to check if something is incorrect is useful; it may be possible to incorporate the idea into the project, to simplify the grammar.

1.4.2 Language Analysis Tools

"Apertium is designed to translate between related languages" Corb-Bellot [2005]

Apertium is an open source language translation tool, which is designed

to translate between two closely related languages; this type of translation is called "shallow transfer" because there is not too much semantic analysis required (the languages being closely linked). The type of architecture used within this system is called the MT (Machine Translation) system:

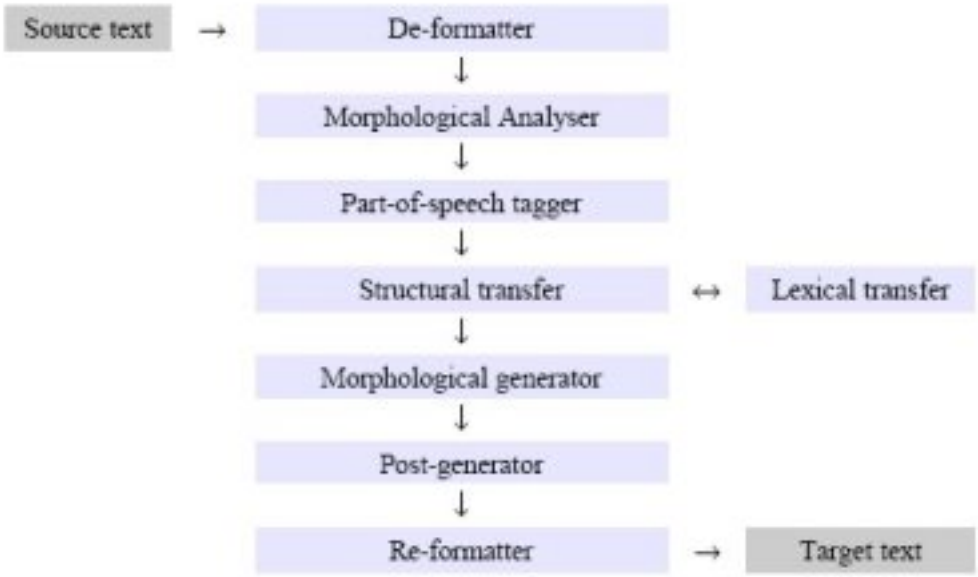

Figure 1.4 The eight modules of the MT system, Apertium [2005]

The modules of importance in this system are the structural transfer, and the lexical transfer. The structural transfer module breaks down the source text into chunks that can be directly translated (some just single words, others simple phrases). They are then passed to the lexical transfer module, which is a dictionary of translations, and converted into the target language. The morphological generator handles any morphemes that may have been entered into the system by translating into the new language. The remainder of the modules are concerned with parsing the original text, and restoring it's form once it has been translated.

Because this system is designed to translate between two fairly similar languages, the analysis of the text is not very strong. When creating a program to generate text, the second half of this system will have to be produced, from

the lexical transfer module downwards. However there will be no source language, so the concept of using translatable phrases will not be applicable. It would be possible to generate text based upon a dictionary that contains lists of acceptable phrases, but this would limit the power of such a system, restricting the amount of possible sentences that can be produced.

There is an English grammar tool called the linGO Grammar (linguistic online grammar), this is a project run at Stanford University to produce a grammar to describe the English language Copestake [2000]. It uses a head driven phrase structure grammar, and a 'lexical knowledge base'. A head driven phrase structure grammar (HPSG) is a phrase structure grammar that is based around 'head' words in each phrase (e.g. the verb in a verb phrase defines how the phrase is structured) Sag [2001]. This is an attempt to simplify the process by removing some of the elements from a transformational grammar, and placing them in the phrase structure, which is organised into hierarchies of different kinds of phrase.

The lexical knowledge base is a very important part of the system; it is a structure that contains encyclopaedic knowledge of the English language. This enables the grammar to produce sentences that have semantic meaning as well as syntactic correctness. Without a lexical knowledge base, the grammar is unable to combine ideas in a coherent way, and produce sentences that have meaning to the user. A structure of this nature will be required, if the program is to produce meaningful sentences.

1.4.3 Existing Random Generators

There are numerous different scripts and programs on the Internet, which generate text in a random, or pseudo random, manner. Most of the programs are fairly basic, but there are two main techniques used when creating random generators.

'Phrase Grammar' based

The phrase grammar based generators use a grammar whose components are phrases, rather than words. These phrases are interchanged with each other to produce sentences, and groups of sentences. There are also generators that simply switch in and out the verbs and nouns from a 'script', meaning

that the text that is generated is always the same apart from the names and actions. The more advanced versions have lots of different variables, and different sentence structure. These can generate some good passages (see Appendix 2.1 for a sample taken from Zelenski [1999]).

Although some of the more complicated grammars can generate plausible passages, this method of generation is fundamentally limited to the structure that exists within the grammar. Without creating a very complicated grammar, there is no way that sentences being produced can depend upon the sentences that have already been produced (no linking between sentences) and the majority of the logical analysis of linguistics is not taken into account. Therefore, although this type of program produces some nice results (within their domain), there is little scope for enhancement.

Phrase Structure based

The phrase structure based generators use a phrase structure grammar to model the structure of a sentence, and then randomly insert words into the positions in the grammar to form sentences. This is a more powerful model of producing sentences than the phrase grammar based systems, because sentences are produced in a free-form manner, not constrained by any 'script'. However this method usually has the side effect of producing completely unintelligible sentences, due to the random insertion. It is also difficult to link sentences together, as usually each sentence is just as random as the preceding sentences. A good example of this type of generator can be found at Kelly [1998].

The generator at Kelly [1998] proves that sentences can be created using the phrase structure method, however it uses quite a limited dictionary in its examples to ensure that valid sentences are produced (this makes it not much different from the phrase grammar version). The two areas that need to be improved using this technique are, ensuring that the correct nouns are inserted (so the sentence makes sense), and linking subsequent sentences together. Using an advanced system of lexical insertion should start to fix the issues with invalid nouns. Linking subsequent sentences together will require some higher form of semantic analysis, to ensure the sentences are about the same subject or topic.

1.4.4 Multiple Sentence Generation

One of the aims of this project is to explore the way in which sentences are linked together, and generate multiple sentences at the same time. This is connected to the semantics of the sentences that are produced, and the concept has been explored in the creation of interactive video games. Characters react in a realistic way when the user interacts with them, based upon the current circumstances in the game.

A method of generating speech between two characters is outlined in Cavazza [2005]; this method is based around the characters that are communicating and events that are happening in the game. It uses the affinity between the characters, the goals of the conversation, the events taking place, and the roles of the characters to build a semantic picture of the sentence that is to be produced. This semantic picture is then made into a sentence by using a 'Tree-Adjoining Grammar'.

(?interrogative) (?actor) (?take-part) (?event :type:party) (?event ?property) example of a semantic level structure from Cavazza [2005]

The examples outlined in this paper are set in quite a limited framework, with quite a specific goal, however the principal is shown. If it is possible to extend the ideas given here, then it could be applicable in linking sentences together. If the element of characters were introduced into the system, then these characters could also have affinities, and their actions could be partially defined by the current setting. This is slightly removed from the context of characters speaking to each other, but the principals of having some global 'setting' which helps keep the sentences linked within that setting might be useful.

"Faade is an artificial intelligence-based art/research experiment in electronic narrative" Mateas [2005]

Faade is a game where the user enters a story as them self, and interacts with the characters involved, changing the way in which the story evolves. The game is focused around interacting with the other characters in the system, and most of the play is focused on the behaviours and actions of these characters. The system understands natural language, and the characters

respond to what is being said, and reply. The game is governed by the situation that is presented to the player, and the storyline follows an arc, but is not fully predetermined. The system is controlled by a 'beat', which is part of the program that determines what should happen next in the story, and what the characters next actions should be.

Faade is a real example of character / situation based conversation generation, although it is also quite constrained in the setting that is given to the player, and the character goals are fairly static, it is a good development, and shows that this technique is viable.

1.5 Conclusions and Plan

From the research done there are several things that need to be considered when attempting to create a system to generate language:

Constructing a viable phase structure grammar
Lexical insertion of the correct nouns / prepositions
Morphology, changing the tense of verbs
Adding transformational rules to create a transformational grammar
Creating some governing rules to link sentences together

The first task is to construct a framework that will allow the creation of a complex phrase structure grammar, which will ultimately describe a sentence in the English language. This will need to produce a phrase structure tree, so that transformational rules can be applied later in the process. This structure can be created without reference to the generation of natural language, because it is based purely on the rules of phrase structure grammars, therefore it can be made using very simple test grammars.

Once such a framework exists, a basic description of the English sentence can be applied to it. The rules for lexical insertion can then be created, to ensure that the correct type of nouns and prepositions are used with the valid verbs, these rules will be quite a complex structure, but should be made in as flexible a way as possible, so that the system is extensible. Morphology can be considered when creating the transformational rules that apply after the phrase structure has been calculated, transformations can be defined to change the tense of a passage, or sentence.

Transformations can also be defined to change the meaning of the sentence that is being generated; examples such as the negation transformation, and the question transformation are given in Lasnik [2000]. When the transformations are being defined, testing will have to be done to ensure that some transformations don't result in generating incorrect sentences from unexpected conditions.

During the process of creating the sentence generator, entering the produced text into a grammar checker can check the 'correctness' of the sentences that are produced.

When the system is capable of producing an acceptable number of different sentences, then some governing rules can be created, so that the sentences created are linked in some way.

Chapter 2

Appendix

2.1 sample grammar

Sample grammar from a phrase grammar based generator, can be seen demonstrated at The Random Sentence Generator Zelenski [1999].
{ (start)
The (object) (verb) tonight. ;
}

{ (object)
waves ;
big yellow flowers ;
slugs ;
}

{ (verb)
sigh (adverb) ;
portend like (object) ;
die (adverb) ;
}

{ (adverb)
warily ;
grumpily ;
}

Bibliography

Dr Bas Aarts. The internet grammar of english, 1998. URL `http://www.ucl.ac.uk/internet-grammar/frames/contents.htm`.

Emmon Bach. *Syntactic Theory*. Holt, Rinehart and Winston, 1974.

M Cavazza. Dialogue generation in character-based interactive storytelling. In *AAAI First Annual Artificial Intelligence and Interactive Digital Entertainment Conference, Marina del Rey, California, USA*, 2005.

Noam Chomsky. *Syntactic Structures*. Moulton & Co., 1957.

V.J. Cook. *Chomskys Universal Grammar 2nd Edition*. Blackwell Publishers, 1996.

Ann Copestake. An open source grammar development environment and broad-coverageenglish grammar using hpsg. In *LREC 2000 2nd International Conference on Language Resources & Evaluation*, 2000.

Antonio M. Corb-Bellot. An open-source shallow-transfer machine translation engine. In *Proceedings of the European Association for Machine Translation, 10th Annual Conference, Budapest 2005*. Transducens group, Departament de Llenguatges i Sistemes Informtics, 2005.

Roger Fowler. *An Introduction to Transformational Syntax*. Routledge, 1971.

M Gross. *Introduction to Formal Grammars*. Springer-Verlag New York Inc, 1970.

Zellig Harris. *A Grammar of English on Mathematical Principals*. John Wiley & Sons Inc, 1982.

Richard Hudson. Word grammar, 2005. URL http://www.phon.ucl.ac.uk/home/dick/wg.htm.

Aravind K. Joshi. Tree-adjoining grammars. *Handbook of formal languages, vol. 3: beyond words*, 1997.

Charles I Kelly. Fun with randomly-generated sentences, 1998. URL http://www.manythings.org/rs/.

Howard Lasnik. *Syntactic Structures Revisited*. The MIT Press, 2000.

Michael Mateas. Faade: An experiment in building a fully-realized interactive drama. In *Game Developers Conference, Game Design track, March 2003*. Literature, Communication and Culture and College of Computing, Georgia Tech, 2005.

Daniel Naber. *A Rule-Based Style and Grammar Checker*. PhD thesis, Bielefeld University, 2003.

Michael Newby. *The Structure of English: A Handbook of English Grammar*. Cambridge University Press, 1987.

Andrew Radford. *Minimalist Syntax: Exploring the Structure of English*. Cambridge University Press, 2004.

Ivan Sag. Head-driven phrase structure grammar, 2001. URL http://hpsg.stanford.edu/ideas.html.

Daniel Temperley. Link grammar, 2005. URL http://www.link.cs.cmu.edu/link/.

Owen Thomas. *Transformational Grammar and the Teacher of English*. Holt, Rinehart and Winston, 1974.

Elizabeth Traugott. *The History of English Syntax*. Holt, Rinehart and Winston, 1972.

R.W. Zandvoort. *A Handbook Of English Grammar 7th Edition*. Longman, 1975.

Julie Zelenski. The random sentence generator, 1999. URL http://www-cs-faculty.stanford.edu/ zelenski/rsg/.