Generating English Language From Formal Grammars

David Perryman

October 20th 2005

Contents

1 Project Description

1.1 Introduction

How closely is language connected to mathematical structures, and how easily it can be represented using them? Is it possible to capture the essence of English and express basic ideas, using a computer program, essentially powered by a random generator? By extension, how much of the art of writing in English can be captured in an entirely logical and mathematical environment? Furthermore, how powerful are formal grammars and what is the best method of approaching this problem? Where does formal logic break down, and how can it be enhanced to overcome its weaknesses?

1.2 Aims & Objectives

The primary goal for this project is to generate sentences, based upon a formal grammar, driven by random generator. Therefore the first problem to solve is finding an appropriate grammar that will represent a sentence in English (The fundamental construction of grammars is discussed in *Introduction to Formal Grammars* [2]). It will be necessary to research the basic structure of the English language [5] to find a suitable grammar. A list, or database of different types of word will be required (to be built up over the course of the project). These words will be randomly placed into the grammar to form sentences. Initially a limited grammar will be used, and limited dictionary of words, so only a small number of sentences will be produced. More complex rules will then be added to the grammar, such as creating a transformational grammar (first outlined by Chomsky [1]), and the dictionary expanded until the program is capable of expressing a significant proportion of English. The aim is not to be able to generate the entire English language; there are too many rules, and too many irregularities to attempt such a program. However, many concepts can be represented, not by building up very complicated sentence structures, but by conveying the equivalent meaning of a complicated sentence by using several shorter, and simpler sentences.

Once a program that can generate these sentences has been produced, the next phase is to explore the way in which sentences are connected, how they are linked by common elements, and begin generating sentences in pairs. There are several structures in the English language that allow for sentences generated in pairs, such as If/Then and Either/Or. These are the first kind of linked sentences to research, and attempt to incorporate into the program. Further research into how sentences are linked within paragraphs will enable the generation of several sentences at the same time, which are linked in some way. This will require the generation of sentences that are not entirely random, but based upon the preceding (and even subsequent) sentences in some way. Some kind of pseudo-random generation will be needed, where the random generator only fills in the unknowns of the sentence (the rest of the sentence is predetermined in some way). If this is significantly successful then it may be possible to explore how different paragraphs are connected, and attempt to generate several paragraphs at a time.

During this project a program must be used to represent a formal grammar, words must be randomly picked from a database, or store and placed into the grammar to

form sentences. The first problem is to create a program that can easily represent a grammar, allowing the grammar to be changed in a fairly trivial manner. The language used needs to be flexible enough to add different rules to the grammar, and change the way it functions in a simple way, so complexity can be added to the project in small increments. The language will need to be relatively portable, as the program may need to be compiled both on home machines, and on the university machines.

1.3 Sources of Information

There are several different sources of information that can be used; firstly there are programs (such as Microsoft Word) that check the grammar of a document. This is the opposite function of the project, however the process used is similar and it is therefore relevant. There are also programs that attempt to translate a document into another language, which involves analysing a piece of text, and breaking it down into the fundamental meaning, then rewriting it in another language. It is the fundamental semantic meaning, and generating text from that meaning that will be of interest, because generating text based upon some general concept is linked to the pseudo-random generation previously mentioned. Another source of information comes from several graduate students at MIT, who have created a computer science paper generator called SCIgen [3]. It randomly generates feasible looking computer science papers based upon a “hand-written context-free grammar”. The code for this program is available, and a lot of the concepts used in this program will be applicable to the project, however SCIgen does not produce sentences that are particularly linked, or even coherent at times, its main function is to produce something that looks like a computer science paper.

2 Requirements

2.1 A grammar must be created that is suitable for representing a sentence in the English language. This does not need to be very complex to start with, but a grammar will be required to be developed during the initial stages of the project. Creation of this grammar will require a research into the basic structure of the English language.

2.2 A dictionary of words will be required, that can be used to create sentences from the grammar. This does not need to be large to start with, but it does need to be well structured, as it will be expand greatly, when the program becomes more powerful.

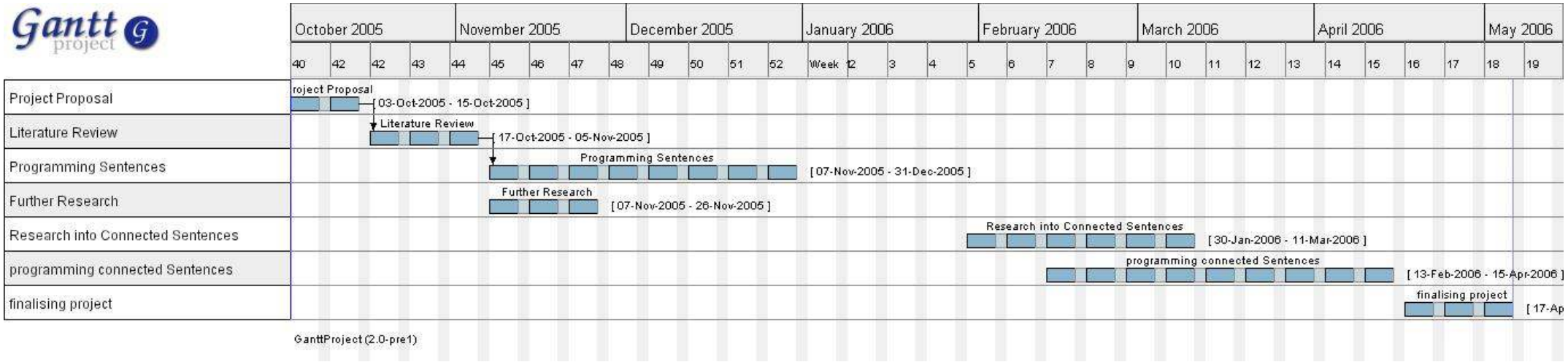
2.3 The program must take the grammar, the dictionary of words, and randomly combine them to create sentences, which are then output to the screen. Ensuring that the produced text is in correct English is the main task of this project, and will require more research into the way grammars work, and complicated structures of the English language.

2.4 The program must be enhanced so that it is capable of generating linked sentences. This study will expand to generating several sentences at once, which are all connected by some common element(s).

2.5 This program does not require a lot of user interface, because the primary function is to produce text, in a random manner. This only requires the user to press “go” and the text to be produced. There may be a limited amount of interface, for testing purposes, that controls the type of sentences that can be produced, but there is no plan to have any options in the final program.

2.6 The aim of this project is not to produce every possible sentence in the English language. However, all the sentences that are produced must be correct.

3 Project Plan



The first two weeks of the project (3rd Oct – 15th Oct) will be spent writing the project proposal, then the next 3 weeks writing the literature review. After that a period of doing research alongside programming, when the grammar for sentences will be developed. Initially a program must be devised that can be used to easily represent a formal grammar, and produce the appropriate output. The text that is generated, and the structure of how the text was produced are required. This program can be written without reference to the English language, just using very basic grammars with simplistic languages generated. Once a suitable program has been developed, a basic English grammar can be applied to it, that only produces a limited number of sentences, and then the complexity can be built up until it is capable of producing many different sentences. It is planned that this phase be finished by Christmas, and then there is a break for about a month during the revision / exam period. This will also give some time to catch up if insufficient progress has been made. The next phase involves researching connected sentences, and will take place after the exam period. Then there is a period of research alongside programming, so that the complexity of the connected sentences can be progressively built up.

During the course of the project documentation will be written alongside the research and development that are planned, this will help keep the write-up consistent, and ensure all the stages of development are correctly recorded. At the end of the project 3 weeks have been allowed for finalising, including ensuring that the written part of the project is up to date, and this will also leave time for any remaining work to be done.

4 Resources

The basic structure of the English language [5] will need to be researched, so that a grammar can be formed to describe a sentence. The information required initially is quite trivial; the library and Internet will be sufficient. However as the project progresses towards looking at the semantics, the grammar being used will become progressively more complex, so some language material may have to be researched, to get more in depth information.

In order to construct an appropriate grammar, formal grammars will have to be researched, both elementary grammars, and also more complex structures (such as transformation grammars [4]), which might be capable of expressing the English language. Most of the research will come from books, both at home, and also additional books in the library. At MIT there is an online resource, where the SCIgen project [3] is hosted. This project will be researched, to see what can be learnt form their approach the problem of generating sentences.

The chosen language for this project is java, there is a java IDE called jBuilder that can be downloaded for free (the foundation version) from Borland. This will be used to program the main project; in addition there is also a java compiler on the university server. Using java means that the program will be portable, and also that it will be possible to put the finished application onto the Internet at the end of the project.

5 Agreement

Student's Signature

Supervisor's Signature

6 References

[1] Noam Chomsky: *Syntactic Structures*, 1957

[2] Maurice Gross, André Lentin: *Introduction to Formal Grammars*, 1970

[3] Jeremy Stribling, Max Krohn, Dan Aguayo: SCIgen
http://pdos.csail.mit.edu/scigen/

[4] Howard Lasnik, *Syntactic Structures Revisited*, 2000

[5] William Idsardi, Introduction to Linguistics
http://www.ling.udel.edu/idsardi/101/