GeneTUC

An NLP System for Biomedical Texts

Anders Andenæs

December 18, 2000

Abstract

GeneTUC is an NLP system built on the TUC architecture at the Norwegian University of Science and Technology. Its primary aim is to extract factual assertions from biomedical research articles and compile these into a knowledge base. GeneTUC has currently an 8 percent success rate on a large corpus of Medline abstracts.

On a wider horizon, GeneTUC may become a full-fledged knowledge system, capable of answering queries as well and performing story summarisation and common sense reasoning.

Acknowledgements

I wish to thank my two supervisors: *Tore Amble*, for providing the best help one could ask for. His devotion to the TUC project is much admired. *Tor-Kristian Jenssen*, for giving new perspective and motivation for the project.

Eivind Hovig, Department of Tumor Biology, Institute for Cancer Research at the Norwegian Radium Hospital, for his invaluable help in making me understand some of the complexity of molecular biology and genetics.

Finally, *Einar Fløystad Dørum* and *Torgeir Rhoden Hvidsten*, for having the knowledge and skills I lack.

Preface

Three weeks ago, my confidence in GeneTUC was at an all-time low. The results were dismal, and I had little faith in GeneTUC ever becoming competitive in comparison with other systems. But, then the clouds cleared, all of a sudden. The rate of success took a leap, that is, it continued its exponential growth which had persisted all autumn. In addition, a conversation with my supervisor made me realise that this application of GeneTUC is only the first milestone.

The true potential of the TUC architecture lies beyond simply extracting atomic facts from natural language texts. Compiling such a database is a necessary step towards a deeper understanding of written text. What is just as interesting, is how this database may be utilised later on. One application of the database is to provide answers for questions like "Does gene A regulate protein B?". But it can also be used in more complex terms like explaining, "*How* does gene A regulate gene B?", or reasoning, "Does gene A affect expression of gene B?". The fundamentals of the TUC architecture allows for such applications, albeit in a presently distant future.

This is how my confidence in TUC and GeneTUC was restored, and why my conclusions are much more positive than they had been only weeks ago.

This report is written in GNU Emacs and typeset in Palatino 10pt, using LaTeX 2$_\varepsilon$.

Anders Andenæs, December 18, 2000

Contents

List of Figures

1 Introduction

This chapter presents the GeneTUC system. The thesis is presented along with an overview of the rest of this report.

1.1 GeneTUC

GeneTUC is an application built upon the TUC framework from the Institute of Computer and Information Science at the Norwegian University of Science and Technology. The GeneTUC application utilises much of the experience made developing the BusTUC bus route oracle, but is primarily aimed at extracting knowledge from research articles pertaining to molecular biology and genetics.

The goals of the project can be summarised as follows:

- **Transfer TUC** framework to a new domain not apparently connected to bus routes. This tests the framework's domain independence, as well as to what extent the bus route semantics needs to be re-written to conform to another domain, and how easy this is.
- **Assess the scalability** of TUC, both in terms of a large vocabulary and processing speed, and a potentially very large semi-permanent database.
- **Improve** the existing system. Check for errors and omissions in the grammar and the domain-independent semantics. Also, find errors in the architecture not detected while running BusTUC.

In addition to the goals related to TUC, GeneTUC also has aspirations regarding genetics:

- Create a natural language processing (NLP) **system capable of extracting assertions** from plain text, chiefly abstracts of research articles as found in large collections on the Internet, thereby
- **creating a database of such assertions** publicly accessible by
- creating **multi-purpose NLP interface** to interact with the database and other publicly accessible databases. This NLP should be both easy to use, expressive and efficient.

1.2 Thesis

Biomedicine and human genetics has gained an increasing amount of attention from medical researchers and research facilities over the last years. The Internet has provided us with easy access to large collections of scientific literature, e.g., the Medline database (PubMed), which contains abstracts from scientific articles. The abstracts are stored in their original form, as large chunks of free text.

We claim that a NLP system based on the TUC architecture will be able to extract factual assertions from such texts, thereby compiling a knowledge base of biomedicine and genetics. Running this system on a large corpus of Medline abstracts will give a measure of its success rate.

1.3 This report

This document reports on the progress of GeneTUC since [And00] and the current state of the application. Some further developments of TUC and GeneTUC are suggested at the end of the report.

Chapters 2 through 4 provide background for the project and basic knowledge to motivate the development of an application like GeneTUC. Chapter 5 presents the specifics of GeneTUC. Chapters 6 and 7 contain results and discussion of the project. Chapter 8 presents the pending conclusions and Chapter 9 suggests some future enhancements of GeneTUC and TUC.

2 Background

This chapter will give a brief description of the background for the report. The aim of the first section is to try to familiarise the reader with some of the basic concepts of text mining. Some of the currently most popular techniques are described in brief.

The knowledge-based approach to text mining is discussed next. Knowledge systems in general are introduced, and the general methodology is explained in brief. Also, this section gives a short summary of why it is appealing to use knowledge systems for text mining.

The chapter concludes with a summary of what is meant by natural language processing, and a description of the TUC system.

The contents of this chapter is based on what is found in [And00].

2.1 Text mining

The term *text mining* refers to retrieving knowledge from a piece of text. This can be performed manually by reading through the text, or automatic by having a computer process the text and extract factual assertions. The manual process can be assisted in various ways by automatic systems.

Information Extraction (IE) is an application of natural language processing which takes a piece of free text and produces a structured representation of the points of interest in it. One way for this to work is to perform syntactical and semantical analysis of the input in order to produce sound output.

Information Retrieval (IR) is what is often referred to as computerised searching, and can provide an aid i manual text mining. In IR, we seek to find the sources of the knowledge; extracting the knowledge from these sources is left to the reader. Free text searching and stored query searching are two of the most common IR techniques.

2.1.1 Free text search

The straightforward way of IR searching for information in textual data, is by executing a free text search in the data source. The large search engines on the Internet, e.g. *AllTheWeb* and *AltaVista*, are based on this, as well are

search functions in common application programs. Free text searching is not a monolithic technique as such, but a way of searching for words in corpora without considering context. Many methods for free text searching are easy to implement, and this is therefore the most widely used searching paradigm.

As the free text searching is based on a purely syntactical analysis of the query string and data base, the degree of relevance in the search result will vary. The search engine will rarely contain any domain knowledge, thus searching for a homonym[1] will possibly return undesirable results. A search for *temple* will find places of worship as well as information on the human anatomy.

In an effort to overcome this flaw, most search interfaces offer some kind of query language. The degree of user-friendliness inherent in these interfaces is different from case to case, but often the threshold for effective use is unnecessarily high. Constructing an efficient query string requires the user to be competent in some proprietary query language, or regular expressions (RegExp's).

2.1.2 Stored queries

Another way of addressing the problem, is by trying to match the query at hand with some previously stored query. An example of this approach is the WWW search engine *"Ask Jeeves"* [Ask]. This service catalogues the answer to all the searches it conducts, thus making itself more competent each time it is accessed.

The stored query technique is essentially just an adaptation of and interfacing to free text searching. Hence it suffers from the some of the same deficiencies as the latter. The strong point of the stored query technique is the possibility to customise the stored queries in a way that provides the highest quality output.

2.1.3 Cross-matching

The two methods described above are examples of IR techniques. What is more interesting in our context are techniques based on IE. Some of these methods will be described in the following section and in section 2.5.2.

Some efforts have been made in developing systems for cross-matching keywords in a cause-effect relationship. The *Arrowsmith* system [SS99], is basically an extension to the *MEDLINE* [MED] searching facility. It uses the results of MEDLINE searches to infer knowledge of causalities. As an example, consider the following scenario: The tabloids often claim that excessive intake of caffeine causes headache. To investigate this, one could submit MEDLINE searches for the words "caffeine" and "headache". These results would then be fed into the Arrowsmith system, which, in turn, would try to find some unknown factor X which is such that caffeine causes X and X causes headache.

The shortcoming of such a strategy is that the causality relationship one seeks, need not be a single-step one. If the connection between the cause and the effect only occurs through a multitude (and unknown number) of steps, this method fails.

[1] One of two or more words spelled and pronounced alike but different in meaning, see 3.7.2.

2.2 Knowledge-based approach

Using knowledge systems is a powerful and flexible method for mining texts. This section briefly explains what is meant by the term knowledge systems, and how these are applied to extracting information from biomedical texts. Although a knowledge-based approach is not the only way of creating an NLP system, the section closes with some remarks on why using knowledge systems is appealing.

2.2.1 Knowledge Systems

Knowledge Systems are a subfield of Artificial Intelligence (AI). According to [Nil98], the phrase *knowledge systems*, or, more accurately, *knowledge-based systems*, is used to describe programs that reason over extensive knowledge bases, containing facts and *rules*. This knowledge base is implemented in some formal knowledge representation language. In TUC's case, this language is Prolog, but Common Lisp is another widely used language.

An important concept in knowledge systems is semantic nets. A semantic net is the interrelationships between all known concepts in the system. The interrelationships can be of different types, most common are *a kind of*, denoting generalisation / specialisation, *is a*, denoting instantiation, and *has a*, denoting association.

2.2.2 General Methodology

Using the knowledge-based approach to perform text mining requires a natural language-capable knowledge system. Constructing such a system is beyond the scope of this report, hence focus will be on extending an existing system.

The first step is analysing the domain. The system's vocabulary will have to be augmented to be sufficient to extract information and answer queries using the correct terminology. New concepts will have to be placed correctly in the existing hierarchy. (How the hierarchy is constructed is implementation dependent.)

The grammar must often be updated, reflecting how sound sentences describing the domain may be formed. This includes defining valid combinations of nouns and verbs. (See chapter 4 in [And00].)

Then the knowledge base will have to be fed with information. In the case of a natural language-competent system, this is most likely a straightforward task. The system will readily accept texts written in plain language. Structured texts, i.e. texts using some kind of field-formatting, presumably not in natural language, must be reformatted upon entry. Rather, the information could be input directly into the knowledge base, if the internal format of the latter is known.

2.2.3 Appeal

The knowledge-based approach, using natural language processing, may seem cumbersome at first. The preparation of the system, building semantic

nets and defining a sensible grammar, is both tedious and time-consuming. Work on the TUC project (see Section 2.4) was started in the early 1990's, but the grammar and semantics are still far from complete. Furthermore, this method would seem less intuitive than some of the methods described earlier in this chapter.

Still, basing retrieval on searching a knowledge base, holds potentially great advantages. The user friendliness of a well-constructed natural language interface, in lieu of a conventional[2] searching interface, needs not be stressed.

It is crucial to recognise that the knowledge based-approach relies on semantical, rather than syntactical, analysis of the data base. Unlike the conventional searching methods, do homonyms not pose a problem, given an adequate semantic net. Thus the query "Which *band* plays at Studentersamfundet next Friday", will surely output the name of a musical group (or none) and not instructions on how to tie objects together.

Another strong point of the knowledge base is the ability to extract implicit knowledge. Returning to the caffeine and headache example, a correctly constructed knowledge base would find any connection between those two. Provided such a connection exists and is implied by the data base, of course.

2.3 Natural language processing

The FAQ[3] for the comp.ai.nat-lang [NLP] newsgroup gives the following definition of Natural Language Processing (NLP), or *computational linguistics*:

> Computational linguistics (CL) is a discipline between linguistics and computer science which is concerned with the computational aspects of the human language faculty. It belongs to the cognitive sciences and overlaps with the field of artificial intelligence (AI), a branch of computer science that is aiming at computational models of human cognition. There are two components of CL: applied and theoretical. [...] The applied component of CL is more interested in the practical outcome of modelling human language use. The goal is to create software products that have some knowledge of human language. Such products are urgently needed for improving human-machine interaction since the main obstacle in the interaction between human and computer is one of communication. [...] Natural language interfaces enable the user to communicate with the computer in German, English or another human language. Some applications of such interfaces are database queries, information retrieval from texts and so-called expert systems.

TUC, described in the next section, is an example of such a natural language processing system.

[2] i.e. using RegExp's or a proprietary query language.
[3] Frequently Asked Questions

2.4 TUC - The Understanding Computer

The TUC project was initiated at NTH[4] in the early 1990's. It was based on a number of previous efforts in creating a natural language interface for querying data bases, among them CHAT-80 [Per83], PRAT-89 and HSQL. The research goals for the project could be summarised as follows:

- Give computers an operational understanding of natural language
- Build intelligent systems with natural language capabilities
- Study common sense reasoning in natural language

The TUC project seeks to define a language denoted by NRL[5]. This language is as readable as plain English, but has well-defined syntax and semantics. In TUC, NRL serves as both a declarative knowledge definition language, and as a query language [Amb94].

TUC relies on grammatical analysis for marking sentence elements. A sentence not being grammatically correct (according to TUC's internal grammar), will be rejected without further treatment. Enhancing TUC is thus both a question of adding to its vocabulary and semantics, *and* defining new grammatical constructs.

2.4.1 Inner workings

The language analysis in TUC is a five step process [Bra97], as shown in figure 2.1[6]:

- **Lexical analysis**
  The individual words of the input string is looked up in TUC's internal dictionary. If a word is not found in the dictionary, the lexical analyser tries to find it in a case specific data base containing words mentioned in earlier sentences. The lexical analyser also performs some spelling correction.
  If the input sentence is successfully analysed, the set of words are output as tokens in their inflective root forms, together with their possible word classes.

- **Syntactic and semantic analysis**
  The list of tokens is parsed using a differential attribute grammar. The parser builds a TFOL[7] [Amb99] formula representing the semantics of the sentence. It will output the first TFOL representation it finds that is syntactically and semantically satisfying.

- **Anaphora resolution**
  Anaphora[8] are replaced with the internal object they represent.

[4] Norwegian Institute of Technology, now the Norwegian University of Science and Technology
[5] Naturally Readable Logic
[6] Figure taken from [Bra97].
[7] Temporal First Order Logic
[8] Anaphora are words or phrases taking its reference from another word or phrase, e.g. she, it, then, see 3.4.

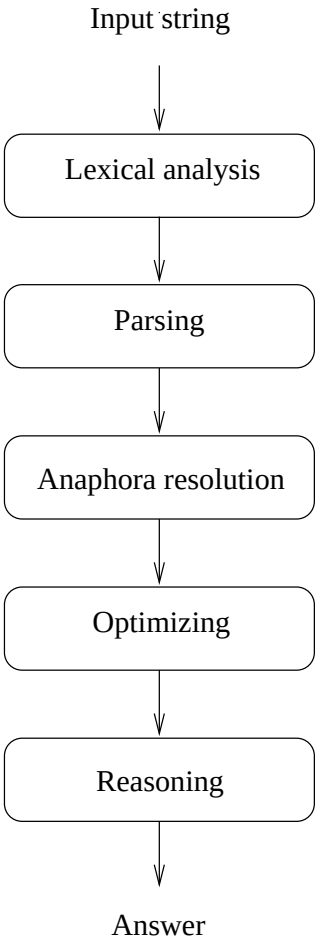

Figure 2.1: TUC's language analysis process

- **Optimising**
  The TFOL formula is skolemised and simplified into a TQL[9] formula.

- **Reasoning**
  TUC uses the TQL formula to answer questions posed upon the system.

For clarity reasons, TUC's knowledge base is referred to in two ways. The *a priori* knowledge, i.e. the facts and rules hard-coded into the system, will be called its *permanent database* or *semantics*. The facts extracted from input text, is stored in the *semi-permanent database*.

2.5 IE in genetics and molecular biology

During the last years, extracting information from scientific articles about genetics has caught the attention of NLP specialists worldwide. Many undertakings have been set forth, some yielding quite good results.

2.5.1 The domain

The domain of human genetics is extremely complex. It is therefore virtually impossible for any human to retain a complete overview of all the genes, proteins and chromosomes involved, or how they interact with each other.

At the same time, a vast amount of research is put into this field, all over the world. Conferences are held, articles publicised, books written; the quantity of information is excessive. Collecting all this information, and extracting the essence of it, is a task for computers. Correlating information from multiple sources, has already become a science in its own right.

The following section describes some of the ongoing initiatives.

2.5.2 Existing work

Finding a suitable ontology as a basis for the natural language system, is a key element in the process. [BCK+00] outlines a system which is based on UMLS[10]. UMLS is a gigantic system, comprising some 475,000 semantic concepts and 600,000 categorisations. UMLS is a collection of several knowledge sources applicable in the biomedical domain: the Metathesaurus, the Semantic Network and the SPECIALIST Lexicon. The Metathesaurus has a structure similar to that of TUC, displayed in figure 2.1. In addition, a new method of displaying the extracted knowledge using *keynets* is described in the article. In fact, many systems, primarily of American origin, are based on the UMLS.

The *EDGAR*[11] system, described in [RTWH00], uses a somewhat similar approach. Like the system in [BCK+00], EDGAR is also based on the UMLS ontology. Unlike the aforementioned system, EDGAR uses a stochastic part of speech tagger along with an under-specified syntactic parser to analyse the texts. The output of the parser provides input for a rule-based system that uses

[9] TUC Query Language
[10] Unified Medical Language System
[11] Extraction of Drugs, Genes And Relations

both syntactic and semantic information to extract factual assertions from the text.

ARBITER[12] as described in [RRH00], also uses existing domain knowledge accessible through UMLS to extract assertions about macromolecular bindings. ARBITER searches for syntactic cues in the input material, in addition to a partial parsing of the texts. Such syntactic cues or "barrier words" indicate boundaries between potential binding arguments, and may be conjunctions, modals, prepositions or verbs. If the barrier is any form of the verb "bind" and the arguments are valid binding arguments, a relation is found.

Highlight, developed at SRI Cambridge and described in [TMO+00, MT00], is a multi-purpose NLP system customised into a system for performing information extraction from biological articles. (A strategy similar to this report's way of customising TUC.) Provided some simplifications of the source material, Highlight has yielded impressive results.

Highlight is based on Sicstus Prolog, but uses Hidden Markov Models (HMMs) for parsing. It is not a full-blown NLP system, in that it does not allow for natural language queries, but requires queries to be input into a pre-defined templates. The user input a number of keywords, and is allowed the added functionality of linguistic and positional constraints. The added constraints may improve precision, at the cost of recall, similar to adding an extra mandatory keyword.

All these systems mentioned above, and others ([SPT00]), uses what is referred to as shallow parsing techniques[13]. As opposed to TUC's strict parsing, the shallow parsing is based on statistics and probability. To create a good parser using this technique, high demands are set upon the founding statistics. Obtaining such numbers is hard and requires minute analysis of a large corpus of texts. Added to this is the likelihood that disjoint domains may have differing traditions in terms of manner of speech and what is considered good form and language. A shallow parsing NLP system spanning large or multiple domains may well be as complex, or even more complex, than the strict parsing one.

What sets all these systems apart from TUC, is TUC's heavy reliance on a complete grammatical and semantical analysis of the input. Whereas TUC rejects ungrammatical information and information not conforming to its semantic base, these systems perform template filling on partial sentences, thus making them more robust in terms of what input they accept. On the other hand, TUC may potentially extract more complex information from the material, given the deeper understanding of the semantics and grammar of the language ([And00]). By successfully parsing the sentences as a whole, utilising its understanding of how sentence elements act on each other.

TUC may extract knowledge stated implicitly in the material, i.e., use *modus ponens* to extract rules from the input. The strict approach also has great advantages when it comes to dealing with scoped modifiers, most commonly negations. The grammar will determine what parts-of-speech the negation dominates, giving more accurate information than a superficial parser would.

Finally, the strict parser will be better suited to disambiguate vague sentences and statements, given a successful parse of the input. The grammar

[12] Assess and Retrieve Binding Terminology

[13] For a description of shallow and strict parsing, see Appendix B

recognises parts-of-speech in very high detail, thus making it easier for the semantic analyser to come up with a correct interpretation.

3 Linguistics for non-linguists

In order to appreciate a grammatical NLP system like TUC, knowledge of the fundamental principles of grammar is required. In this chapter some of the key concepts of natural language are introduced, for which an understanding is crucial when dealing with TUC, or NLP systems in general. Much of what is written in this chapter is based on the excellent book, [HJL98].

The chapter concludes with a section on Naturally Readable Logic. This is a subset of natural language, with certain properties making it ideal as a basis for NLP systems. TUC, and consequently GeneTUC, is based on analysis and processing of NRL, rather than natural language.

This chapter is a re-worked version of the the linguistics chapter in [And00].

3.1 Word classes

The different word classes are the fundamental building blocks of the language. This section describes the most important word classes, their functions and use, according to [HJL98] and [GHb]. Although not exhaustive, in the sense that some classes are left out, this provides a foundation for the discussions later in the report.

3.1.1 Verbs

Verbs is the class of words used for denoting actions. These can be categorised further according to

- **Regularity**
  Verbs can be placed in the subclasses regular and irregular depending on how they are inflected in the past and past participle form. The regular verbs are all inflected according to a general schema, whereas the irregular ones have individual patterns of inflection.

- **Transitivity**
  All verbs can be put into at least one of the these transitivity classes:

  – Intransitive - not taking object, e.g., “John *laughs*”

- Copular - taking a subject predicative, e.g., "John *became angry*"
- Transitive - taking one object, e.g., "John *saw Mary*"
- Ditransitive - taking two objects, e.g., "John *gave Mary a rose*"
- Complex transitive - taking a direct object and an object predicative, e.g., "John *found Mary titillating*"

Note that transitive verbs may require an adverbial:

Mary put the book *on the desk*.

3.1.2 Nouns

Nouns give names to persons, places, things and concepts in general. *Common nouns* denote any member of a set of concepts, e.g., a car, thoughts, a girlfriend. *Proper nouns* give names to a specific member of the set, e.g., John, the Theory of Relativity, Oslo Airport Gardermoen.

Nouns can be derived from verbs and vice versa. Thus, the English are said to "verb their nouns":

The noun *progression* is derived from *to progress*.
The verb *to house* is formed from *house*.

The *Gerund* is a type of word which can act as both a verb and a noun. It is formed as the present participle of the verb. Examples of usage:

As a noun: John likes *programming*.
As a verb: John is *programming* his VCR.

Nouns can also function as adjectives, modifying other nouns, as in

John likes *action* movies.
"...and a partridge in a *pear* tree."

A noun phrase can be expanded by *apposition*, that is two usually adjacent nouns or noun phrases having the same referent standing in the same syntactical relation to the rest of the sentence. Most commonly, a proper noun and a noun phrase further describing the noun is used, like

Bill Clinton, the president of the USA, committed perjury.
They shot *my cousin Vinny*.
There is a *rumour that petrol prices will drop* after the next EU summit.

3.1.3 Adjectives

Adjectives are words used to modify the noun, either as a part of a noun phrase or following a copular verb.

The adjective can be complemented, forming adjective phrases. These phrases are formed in four ways:

- **Adverb and adjective**
  John is *rarely late*.
  This report is not *good enough*.

- **Adjective and prepositional phrase**
  Mary is *fond of boxing*.
  John is *sitting on the chair*.

- **Compared adjectives**
  I am *taller than* you.
  Mary thought *as hard as she could.*

- **Adjective and subordinate clause, participle or infinitive clause**
  I'm *afraid John died*.
  Mary is *good at doing nothing*.
  This key is *supposed to fit*.

3.1.4 Adverbs

The adverb is a word class that modify verbs, adjectives, other adverbs or complete sentences. Adverbs can be combined into adverbial phrases, with the same function as adverbs. The adverbs are grouped into three subclasses:

- **Simple**
  The first subclass is a simple modifier, e.g., "I am leaving *tomorrow*", "I'll eat my dessert *first*"

- **Interrogative**
  Interrogative adverbs are used for asking questions, e.g., "*Where* is my other sock?", "*When* was that?"

- **Conjunctive**
  The conjunctive adverbs connect independent clauses, e.g., "It was raining; *consequently*, John stayed at home", "I think; *therefore*, I am"

3.1.5 Pronouns

Pronouns are used in place of nouns. There are five principle groups of pronouns:

- **Personal**
  Personal pronouns point directly to a person or an object, e.g., "*He* is a good teacher", "Mary saw a film. *It* was scary"

- **Possessive**
  Possessive pronouns are pronouns showing ownership or possession, e.g., "Get off *my* lawn!", "The dog tried to bite *its* tail"

- **Demonstrative**
  Demonstrative pronouns focus the attention on the object pointed out, e.g., "*These* boots are made for walking", "Who's *that* girl?"

- **Reflexive**
  The reflexive pronouns point back at the noun or pronoun that has just been named, e.g., "Mary looked at *herself* in the mirror", "They've bought *themselves* a new car"

- **Relative**
  The relative pronoun joins a subordinate clause to a main clause, e.g., "John saw the girl with *whom* he was in love", "The parrot *that* I bought not half an hour ago, is dead"

3.1.6 Prepositions

Prepositions are words used to show a relationship between its object (noun or pronoun following the preposition) and another word in the sentence

In a galaxy, far, far away.
First *among* equals.

A prepositional phrase includes a preposition, the object of the preposition and a number of modifiers on the object. The prepositional phrase may have an adjectival or adverbial function

Adjectival function: The car *outside the house* is nice (the phrase gives more information on the subject).
Adverbial function: Mary looked *at the man*.

3.1.7 Conjunctions

Conjunctions are employed to connect words, phrases or clauses, possibly indicating the relationship between the elements they connect in the sentence. There are three types of conjunctions:

- **Coordinating**
  Coordinating conjunctions connect elements having the same grammatical function, e.g., "Sticks *and* stones may break my bones", "Many are called, *but* few are chosen"

- **Correlative**
  Correlative conjunctions act as coordinating conjunctions, but work in pairs to connect elements in a sentence, e.g., "*Neither* rain *nor* snow will stop him", "I like *both* vanilla *and* chocolate"

- **Subordinating**
  Subordinating conjunctions connect two elements with different grammatical function, most commonly an independent and a dependent clause, e.g., "It looks *as though* it's going to rain", "*Since* you've been gone, I've been missing you"

3.2 Sentence categories

Sentences may be categorised according to function and structure. In terms of structure, the most important property is whether the verb is placed in front of or behind the subject. Also, sentences belonging to a certain category may have a function similar to a sentence from one of the other categories.

3.2.1 Declarative

The declarative sentence, in which the verb is placed behind the subject, is the most common of the major sentence groups. It is usually less marked in form and less restricted in function than the other groups. As implied by the name, declarative sentences state facts, as in

> John likes to play guitar.
> It was the best of times, it was the worst of times.
> Mary has not eaten her peas yet.

Declarative sentences can be positive, i.e., affirm a fact, or negative, denying a fact. The first two examples above are thus positive declarative; the last one is negative.

3.2.2 Imperative

Imperative sentences are most often employed to issue a command. The imperative sentence often lacks an explicit subject and use the verb in its base form:

> Shut the door!
> Please keep your luggage with you at all times.

The subject of the sentence is, if not the addressee, often given from the context.

3.2.3 Interrogative

Interrogative sentences are used to query the addressee for information. In contrast to the declarative sentences, the verb often precedes the subject in interrogative sentences:

> Have you ever loved a woman?
> Where did all this mail come from?

Sometimes interrogative sentences have a non-interrogative function. Such *rhetorical questions* act as statements or commands, while avoiding having to use declarative sentences, which may seem blunt or obvious:

> Who cares?
> Do you mind closing the door when you leave?

3.3 Ellipsis

Ellipsis is a phenomenon often encountered in dialogue. Ellipsis is the omission of a phrase mentioned earlier in the discourse, a fact that can be inferred from the context.

Elliptic sentences are categorised as sentence fragments; words or phrases not included in a phrase structure but still carry a communicative function:

> Mary showed up late, but then again, she always used to [show up late].
> Do you know how to get there? Yes, I do [know how to get there].

3.4 Anaphora

Using personal pronouns for referring to persons or object mentioned earlier in the discourse, is called anaphora or anaphoric references. Anaphoric references require the speaker and addressee to share enough common knowledge to resolve the anaphora. *Internal anaphora* are anaphora referencing persons or objects cited in the same sentence, *external anaphora* reference earlier sentences:

> Internal: Mary had given up on her fear of flying, and started to like *it*.
> External: John had no idea what *she* was talking about.

The pronoun *it* is extremely general, often making the resolution of the anaphora hard, or introducing ambiguities.

Cataphoric references is a phenomenon closely related to anaphora. A cataphoric reference is a reference dependent on something following the references, i.e., a forward pointer:

> *She* didn't know what to do with *it*, but Mary thanked politely for the gift, and placed it swiftly in the back of the closet.

3.5 Garden-path sentences

Garden-path sentences is a a class of seemingly ambiguous or incoherent sentences. The sentences may seem grammatically incorrect at first glance, but are in fact not. The term "garden-path" relates to the fact that the human reader is "lead down the garden-path" into an interpretation which is ultimately incorrect, and is left confused when the initial parse fails:

> The horse raced past the barn fell.
> The complex houses married and single students and their families.
> The computer screens all the entrants.

Often, the confusion is created by incorrectly determining the class of certain words, or erring when parsing parts-of-speech. Garden-path sentences display three properties, according to [JM00]:

- The are *temporarily ambiguous*, i.e., the initial portion of the sentence is seemingly ambiguous, but the whole sentence is not
- The human parsing mechanism will somehow prefer one of the multiple interpretations of the initial portion
- One of the dispreferred parses is the correct one

Some interesting observations on the psychology of Garden-path sentences is found in [Pat98].

3.6 Metaphor

Metaphors are figures of speech, in which words and phrases literally denoting one concept is used to described another, unrelated concept, suggesting likeness on some level.

> He was so angry he was about to explode.
> Act quickly, time is about to run out!
> Elliot Ness brought the Mafia to its knees.

Arguably, many of our common-day metaphors are motivated by a relatively small number of *conventional metaphor schemas*, such as organisation-as-person, anger-as-heat and time-as-resource. Metonymy[1] is a concept closely related to metaphor.

3.7 Lexical Semantics

Lexical semantics focuses on the meaning of single words, rather than the meaning of whole sentences. The "atom" of lexical semantics is the *lexeme*, which can be thought of as a pairing of a orthographic and phonological form with some sort of symbolic meaning. Lexemes are compiled in a *lexicon*, a finite list of lexemes.

Four of the key concepts of lexical semantics are described in the following sections. Examples are provided in Table 3.1.

3.7.1 Synonymy

The definition of a synonym is apparently simple. Synonyms are two or more different lexemes with the same meaning. A criteria for this is *substitutability*, i.e., two lexemes may be substituted for one another in a sentence without changing the meaning or loss of acceptability.

Generally, the concept of substitutability crosses all domains, thus two synonyms may be interchanged regardless of context. This opens for the weaker notion of *restricted synonyms*, which are lexemes that are substitutable in some context, but not generally.

3.7.2 Homonymy

Homonyms are lexemes with the same form, but with different, and unrelated, meanings. The distinct senses of a homonym often have very different etymological[2] origins.

Related to homonyms are *homophones*, distinct lexemes sharing pronunciation, and *homographs*, which have identical form but distinct pronunciation.

[1] Metonyms are situations where a concept is described by another concept *closely related* to it, e.g., artist-for-artist's-works as in "She was a great fan of the Beatles".

[2] Etymology is the history of a linguistic form since its earliest occurrence in the language where it is found, and by decomposing, tracing and analysing its transmissions and influences from forms in other languages.

Phenomenon	Lexeme	Comments
Synonymy	car - automobile	
Restricted synonymy	great - big	Synonymous when describing size
Homonymy	ball	Spherical object and formal gathering for social dancing
Homophony	great - grate	
Homography	bow	Archer's bow /'bO/ and forward part of ship /'bau/
Polysemy	plane	Aircraft and flat surface (from Latin planum)
Hyponymy	bus - vehicle	A bus is a hyponym of vehicle

Table 3.1: Some examples of different phenomena in lexical semantics.

3.7.3 Polysemy

Polysemes are almost like homonyms, in the sense that they are identical lexemes. They differ from the latter by having *related* meanings. Distinguishing polysemy from homonymy is, as one might expect, not necessarily straightforward.

How will one know if two senses of a word are related? Often, etymology and a native conception of the word can give a pointer to distinguish between polysemes and homonyms.

3.7.4 Hyponymy

Hyponymy is an asymmetrical pairing of lexemes where one lexeme denotes a subclass of the other. The more specific lexeme is denoted a hyponym of the more general one. Conversely, the more general lexeme is denoted the *hypernym* of the more specific one.

Hyponymy is closely related to, and a prerequisite for, creating a *taxonomy*. A taxonomy is a particular ordering of the elements of an ontology[3] into a tree structure, where hyponymy is the ordering constraint.

3.8 Punctuation

Punctuation is used to clarify meaning and separate structural units in the written word. The following is a quick run-through of some rules and recommendations for using punctuation.

3.8.1 Comma

Commas often cause some distress among inexperienced writers. Below is a brief discussion of the comma's usage:

- **Compound sentences**
  Generally, when concatenating two or more clauses using a coordinat-

[3] An ontology is a set of objects existing in a domain or micro-world.

ing conjunction, chiefly "but". However, when the second clause is very short, and the subject is ellipted, use of commas are less common.

> He wanted to go, but he could not make up his mind.
> The secretary was in her late twenties and built like a goddess.

- **Initial adverbials**
  When starting sentences with adverbial clauses, a comma should be used.

  > Generally, her cooking doesn't taste that good.
  > Keeping in mind last Saturday's effort, our national site's chances of qualifying are slim.

- **Trailing adverbials**
  On the other hand, trailing adverbials are not separated by a comma, unless they belong to a separate information unit.

  > She came by at lunch while I was taking a nap in my office chair.
  > She came by at lunch, as mothers tend to do.

- **Enumerations**
  Commas should be placed between parallel phrases or clauses. There is usually no comma before the "and" at the end of an enumeration.

  > My brother likes bananas, apples, chocolate and ice cream.
  > The car went through the road block, raced past the officers and crashed into an off-license.

- **Parenthetical elements**
  Parenthetical elements or final elements representing separate information units, such as non-restrictive relative clauses, comments and appositions, should be set off by commas.

  > Jimmy Hoffa, the notorious Teamsters leader, was formally declared dead in 1982.
  > I wanna know, have you ever seen the rain?

- **Disjuncts and conjuncts**
  Disjuncts[4] are often set off by commas. There are some exceptions, mainly single-word disjuncts placed mid-sentence. Conjuncts[5] should always be marked off by commas.

  > Luckily, this wasn't the case.
  > The plan couldn't possibly go wrong.
  > Mary could, nevertheless, never stop thinking she had done something wrong.

[4] Disjuncts are adverbials which are loosely connected to a sentence and convey the utterer's assessment of the sentence's content.

[5] Conjuncts are adverbials which indicate the utterer's assessment of the connection between two clauses.

3.8.2 Semicolon

Semicolons are used between two independent main clauses which are not connected by a coordinating conjunction. Use of the semicolon is at the writer's discretion, and may in most cases be replaced by a period.

> Peter was late; Monday was never a good day for him.
> People were starting to wonder about the PM; the Governments last initiative on public transportation was clearly ludicrous.

3.8.3 Colon

A colon is used to signal to the user that what follows is an explication of what has already been said. The text following the colon is often not complete clauses.

> And the Lord said: Let there be light.
> Be sure to bring the following things: sleeping bag, thermos, a sharp knife and food to last for three days.

3.9 Naturally Readable Logic

For a computerised system today, natural language as such is far too complex to be fully understood. Rather than trying to do the (at this time) impossible, we focus on a subset of natural language, denoted by Naturally Readable Logic (NRL). As the name implies, NRL is founded on a well-defined syntax and semantics, while being as readable as natural language.

As defined in [Amb94], the requirements for NRL are the following:

- NRL is definable
- NRL is acceptable as English (or other languages on ported systems)
- NRL is a logical language, all acceptable conclusions from a set of statements are verifiable
- NRL is sufficient, i.e., "everything" can be said in NRL

A necessary prerequisite for using NRL for storing knowledge, is that everything we wish to say, may be stated clearly, or, according to Ludwig Wittgenstein's dictum [Wit16]:

> Everything that can be said, can be said clearly. Whereof one cannot speak, thereof one must be silent.

From a logical viewpoint, the premise for the hypothesis is that all statements can be represented using a set of object, events and relations between the two classes. In a stricter sense, one could consider all basic facts as being stored in a temporal database, containing the basic events and relations, i.e., properties of objects. Note, however, this hypothesis falls short when dealing with less fact-oriented utterances, like poetry, emotions, dreams and prayer [Amb00b].

Note that the system needed for analysing complex semantic structures, like metaphors and analogies, is not a part of NRL. Also, NRL pays no heed to punctuation, leading to some undesirable side effects. Punctuation can alter the meaning of sentences, especially when separating multiple clauses and adverbials in single sentences. In such cases, NRL will represent the meaning conveyed by the sentence with punctuation removed.

4 Human genetics

This chapter serves as a brief introduction into the field of human genetics. It is by no means a thorough discussion of the subject, but hopefully provides a minimal foundation for the discussions later in the report.

This section is largely based on [HGP], [Cas92] and [SR99], the cell cycle illustration is adapted from [CCT].

4.1 Key entities

The chromosomes, DNA, genes and proteins are all key entities in genetics, and are essential for creating and maintaining the organism.

4.1.1 Chromosomes

The base pairs in the human genome are organised into distinct units called chromosomes. Most human cells contain two sets of 23 chromosomes, 22 *autosomes* and one of each sex chromosome X and Y (male) or two X's (female). The two sets are given from each of the parents.

When properly prepared and stained with dye, the chromosomes reveal a pattern of dark and light bands, visible when viewed through a light microscope. These bands are regional variations in the amount of the different base pairs along with the attached proteins. Some of the major chromosome anomalies can be detected using this process, e.g. trisomic Down's syndrome, in which the cells include a third copy of chromosome 21.

However, far from all changes in DNA can be detected using the *karyotype* method, as described above. Abnormalities due to mutations are too subtle for this method, but are still responsible for many illnesses, such as cystic fibrosis and predisposition to cancer.

4.1.2 DNA

The *genome* of an organism is the complete set of information describing the structure and activity of the organism. The human genome is comprised of

the DNA[1], and the associated proteins, and is organised into structures called *chromosomes*. The chromosomes are found in the *nucleus* of every cell. To understand how the DNA contains all information for building and maintaining life, information of its structure and organisation is needed.

The DNA consists of two strands wound tightly together, forming a "ladder". Each step of this ladder is made up of a base pair, either Adenine-Thymine (A-T), or Cytosine-Guanine (C-G). The *DNA sequence* is the order of the bases on the DNA as observed on one strand. The DNA sequence specifies the genetic instructions needed to create and maintain the organism.

The genome is represented in the human organism in two copies. The genome is in other words diplod, with one genome from each parent contributed through the chromosomes.

4.1.3 Genes[2]

Genes are the specific sequence of nucleotide bases in the DNA. Genes carry the information about heredity. The gene contains the information which is required to construct *proteins*. The human genome is estimated to comprise 50,000 to 120,000 genes. The gene consists of *exons* (protein-coding regions) and *introns* (non-coding regions). The introns are eliminated from the gene through RNA processing. An average-sized processed gene (mRNA) spans 3,000 base pairs ([Cas92]).

Genes serve as templates for the synthesis of proteins. Three bases, called *codons*, manage the process. This is done indirectly through the use of amino acids and mRNA[3]. The RNA, which is *transcribed* from the DNA in the cell's nucleus, resembles a single strand of DNA[4]. This mRNA moves from the nucleus to the *cytoplasm*. The synthesis of proteins is performed using *ribosomes*, which *translate* the mRNA to proteins. The genetic code is a series of codons which specify amino acids required to make specific proteins. The *gene expression* and protein synthesis process of eukaryotic cells[5] is shown in Figure 4.1.

In laboratories, mRNA has been isolated. This mRNA serves as a template when synthesising cDNA[6]. The cDNA may be used for locating genes on a map of chromosomes.

4.1.4 Proteins

Proteins are the structural components of living cells and tissue, and thus an important building block in all living organisms. According to [Cas92], humans can synthesise about 80,000 different kinds of proteins. On a molecular level, proteins are made up of long chains of subunits called *amino acids*; twenty different kinds.

As mentioned before, three codons direct the process of synthesising proteins. For example, the base sequence ATG will code for the amino acid *methio-*

[1] DeoxyriboNucleic Acid

[2] This section about the genes is written in GeneTUC-compliant NRL, see Appendix A.

[3] messenger RiboNucleic Acids

[4] The main differences lie in RNA having the sugar Ribose rather than Deoxyribose in its structure, and that Thymine is replaced with the base Uracil.

[5] Cells having a distinct nucleus.

[6] complementary DNA; a replica of the transcribing DNA.

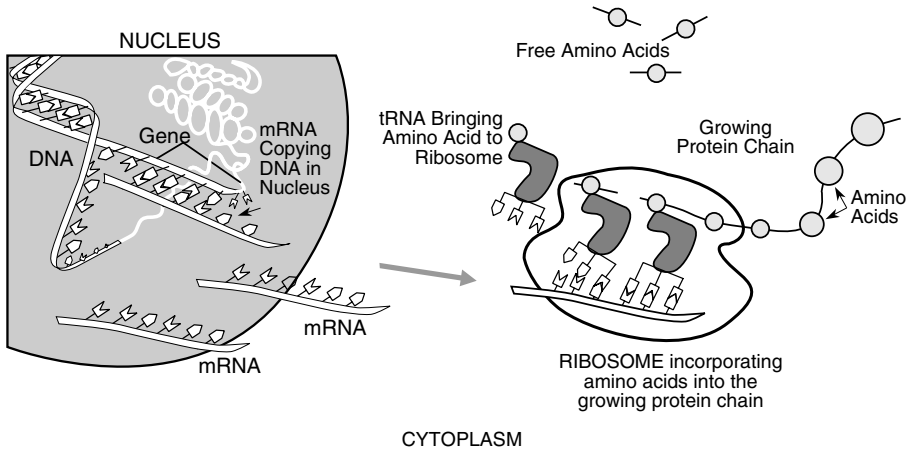

Figure 4.1: Eukaryotic expression of genes. The transcribed mRNA leaves the nucleus and enters the cytoplasm. There, the codons specify the particular amino acids that make up the protein. This process, called translation, is performed by ribosomes. Illustration taken from [Cas92]

nine, which contains sulphur and is important in bodily functions. Since three bases code for one amino acid, a protein coded by an average-sized gene will contain 1,000 amino acids. This number is a bit uncertain; the gene also contains control regions not part of the protein before or after the coding regions.

4.2 Interaction

This section introduces some of the most important interactions between cells, genes and proteins, in addition to those in the protein synthesis process described in Section 4.1.3.

4.2.1 The cell cycle

During the cell cycle, shown in figure 4.2, a cell divides into two daughter cells. During this process, the DNA is unwound, and the two strands are disentangled as a complement is synthesised from each strand, adhering strictly to the base-pairing rules. *Mutation* errors may occur when there are errors in how the new strands are synthesised. Each of the new cells receive one of the "new" DNA molecules.

4.2.2 Protein binding

Proteins bind to DNA and RNA as well as to similar and different proteins to perform a number of tasks. At the DNA, they both perform tasks in keeping chromosomal structural integrity, as *transcription factors* (TFs) to induce transcription of genes. The TF proteins attaches to precursors in close vicinity of

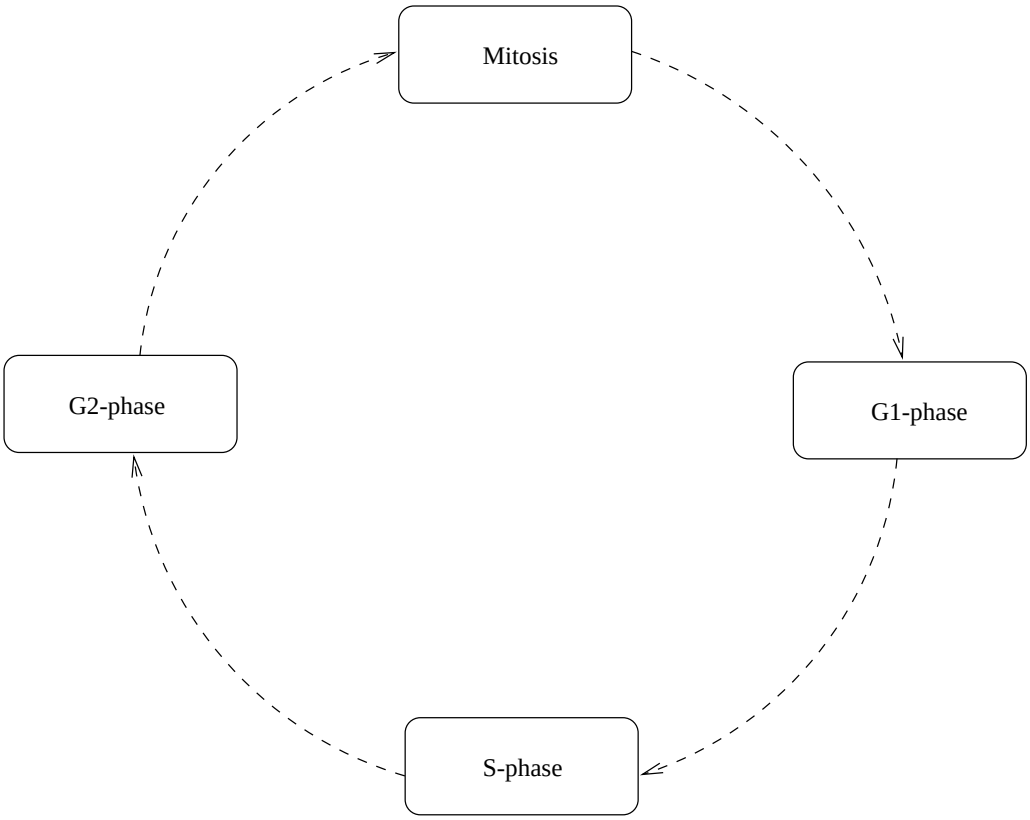

Figure 4.2: The cell cycle. During *mitosis*, cells are divided, giving each of the resulting cells identical complements of the number of chromosomes of the somatic cells of the species[6]. The genes are expressed and the proteins synthesised in the G1 (Gap 1) phase. This phase is the normal state of a cell and the long-term end state of non-dividing cells. In the DNA synthesis, or the S phase, the entire DNA content of the nucleus is replicated. In the G2 (Gap 2) phase, the cell synthesises protein. In this phase, the cell is tetraploid, i.e. having twice the number of chromosomes.

the transcribing gene on one of the DNA strands.

When the TFs are in place, a large enzyme complex, i.e., proteins attached to each other, from a protein class called *RNA polymerases* binds to the DNA to initiate transcription. The polymerase is activated by the TFs and the transcription begins.

The actual protein synthesis takes place in the ribosomes, which are large RNA-protein complexes. The ribosomes are found in the cytoplasm of the cell, but also in the mitochondria and chloroplasts.

4.2.3 Gene regulation

Gene expression is initiated by the polymerases, but they cannot initiate transcription by themselves [SR99]. Combinations of short sequence elements in the vicinity of the gene act as signals for TFs to bind to the DNA. A group of such sequences is often clustered upstream of the gene, and constitute the genes *promoter*. When a number of such TFs have bound to the promoter, the RNA polymerase binds to the TF complex and initiates the synthesis of RNA.

The TFs are said to be *trans-acting*, because they are synthesised by genes at remote locations and migrate to the site of action. Conversely, the promoter is said to be *cis-acting*; its function is limited to the DNA duplex on which it resides.

Enhancers and *silencers* are groups of cis-acting, short sequence elements, which regulate the expression of genes up and down, respectively. Unlike promoters, these sequences are located at variable distances from the gene. They bind gene regulatory proteins and, subsequently, interact with TFs as a result of changes in the helical structure of the DNA.

[6] As opposed to *meiosis*, where each cell is given half the number of chromosomes, giving rise to sperm and egg cells.

5 GeneTUC

This chapter describes some of the work done on the GeneTUC system. It recounts some of the differences between GeneTUC and the original TUC, and what enhancements have been made on GeneTUC, and subsequently TUC, during the project.

5.1 History

GeneTUC stems from the BusTUC system [Amb00a, Bra97]. Work on the system was initiated in January 2000 as a student project [And00]. The original framework was augmented with a moderate number of words from the biomedical domain, regarding gene - protein interactions. All concepts were entered manually, using Medline abstracts as a "training set", trying to incrementally expand GeneTUC's capabilities on a per-sentence basis.

Later, databases containing gene and protein names and their synonyms were imported from the HUGO[1] Gene Nomenclature database and the Swiss-Prot Annotated protein sequence database[2], respectively. This massively increased the size of GeneTUC's permanent database. The permanent base now contains names of more than 10,000 genes and more than 5,000 proteins. A list of adjectives and adverbs was imported from the WordNet[3] [JM00] lexical database.

The development has been towards creating a very general semantic base, allowing for constructs normally not regarded as meaningful[4], but being grammatically correct. This has been done under the assumption that the contents of the input, which is taken from the Medline corpus, has been proof-read and contains no or few meaningless sentences. Furthermore, irrelevant information will not interfere with the essentials of the semi-permanent database. When the semantic and grammatic base is sufficiently "trained", one might constrain the semantics, filtering out "noise" in the input.

[1] The Human Genome Organisation, http://www.hugo-international.org/hugo/
[2] http://www.expasy.ch/sprot/sprot-top.html
[3] http://www.cogsci.princeton.edu:80/~wn/
[4] e.g., genes are *agents* (see below), thus they can speak, as all agents can.

5.2 Goals

The goals for the GeneTUC project can roughly be divided into two parts. The first part is the goals related to further development of the TUC architecture. The second part is the goals related to understanding texts related to genetics.

5.2.1 TUC-related

The TUC framework has thus far only given rise to one application, the BusTUC bus route oracle. One of the key objectives of this project was therefore to assess how easily the TUC architecture could be transferred to another domain, far removed from bus routes. TUC is designed to be domain-independent system, suggesting that porting the framework to a new domain is feasible. Porting the system must be done without interfering with the domain-independent parts of the framework, notably the grammar and part of the semantics. This to maintain some sort of "sideways compatibility" between the applications, i. e., all successful parses in GeneTUC should have a BusTUC equivalent. An example:

> A protein is a marker to predict the outcome.

becomes:

> A bus is a vehicle to pass the airport.

In this manner, development of one TUC application is a development of the entire TUC architecture.

As TUC is still very much a work in progress, the grammar and the domain-independent semantics are not complete. By entering another domain, new errors and omissions are detected; errors and omissions not easily found when concentrating solely on one limited domain (bus route queries). The new application thus aids in the development of the entire framework.

GeneTUC's vocabulary exceeds TUC's by at least one, probably several, orders of magnitude. Hence, GeneTUC is a good test of how well the architecture scales in terms of size. Will the increased vocabulary and semantic net cause the application to run appreciably slower? As the results will show, this is not the case.

GeneTUC is also set to evaluate how well the TUC architecture is ported to one of the most complex knowledge domains of our time. Research and development efforts is molecular biology and genetics provide us with new results and knowledge on a daily basis. This information need not only be collected, it must also be stored and maintained once it is extracted.

Finally, the extended use of the system helps debug the framework, uncovering errors not related to the grammar or semantics, i.e., errors related to the program execution. Some types of sentences cause the program to stop execution; this behaviour has to be checked.

5.2.2 Genetics-related

The primary goal in respect to genetics and molecular biology is to create a NLP capable of extracting factual assertions from a large corpus of literature.

These assertions could be simple, like:

> Here we demonstrate, using a cell free system, that at low concentrations of heparin, FGF4 binds only to FGFR-2, while much higher heparin levels are required for binding to FGFR-1

giving something in the lines of (excerpt):

> bind / fgf4 / 'fgfr-2' / A

or more complex assertions like:

> Risk factors for vascular disease in general and stroke in particular had no visible influence on CRP levels .

becoming (excerpt):

> (A isa influence, B isa level, adj / visible / A / C, adj / crp / B / D,
> nrel / on / influence / level / A / B, has / risk / influence / sk(58) / A, event / real / E)
> =>false

This ultimately leads to a large database consisting of assertions extracted from free-text sources, such as Medline citations. Using the NL interface, all this information would be readily accessible and a valuable asset to geneticists worldwide. GeneTUC, with its knowledge of both written language and logic, could also be used as an NL interface to existing knowledge bases. Potentially, it can become a universal interface to a large number of distributed knowledge bases.

The information GeneTUC extracts is an augmentation of the efforts to map the human genome, such as Celera's[5] and the HGP [HGP]. These projects are well on their way, but a simple mapping of the genome does not eliminate the need for researching into how genes and proteins interact in the organism. Additionally, much science effort has already been put into this field of research over the years. This knowledge needs to be assembled and organised for easy access, a task which GeneTUC might very well perform.

5.3 Adapting TUC

Bus route queries are most often posed in a direct manner, along the lines of[6]:

> When is the next bus from the train station to the airport?

Plain and simple as it may seem, this question does contain some subtleties. The asker expects to get an answer referring to a certain bus departure from the train station, even though the question makes no reference to the word "departure" or any related words, whatsoever. Also, the time reference is a bit vague. The "next" bus may refer to the first bus departing after this instant or the next bus after an earlier mentioned departure. Still, by applying a few

[5] http: / / www.celera.com

[6] Not that BusTUC never encounters ambiguous or complex questions.

protein binds DNA / protein
protein dimerizes with protein
protein interacts with protein
gene codes protein
protein represses gene
protein recruits gene
protein complexes with protein
protein regulates gene / protein
protein inhibits gene / protein
protein associates with protein
protein phosphorylates protein
protein dephosphorylates protein
protein inactivtes gene / protein
protein induces gene / protein

Figure 5.1: Key relationships in the GeneTUC system.

conventions, extracting the meaning of this interrogative sentence is not too hard.

Scientists seldom feel obliged to keep the language in their articles direct and simple, and the tendency is even more apparent in the article abstracts. The abstracts are kept compact, forcing the authors to cram as much information as possible into a small number of sentences. The result is often long sentences densely packed with information, with cascaded subordinate clauses and extended use of ellipsis and anaphora. This affects the readability of the texts, for humans and computers alike.

Another challenge is coping with the different styles of writing of each author. An abstract is typically less than fifteen sentences, and in a large corpus, many different authors will have contributed, each having her own way of presenting her material. GeneTUC must therefore cope with both direct language, as well as more elaborate ways of expression.

5.3.1 Key relationships

Starting the GeneTUC project, it appeared to be wise to focus on a few key relationships. A cancer researcher was consulted [Hov00], and an incomplete list of the most common interactions between genes and proteins was produced, shown in Figure 5.1.

As the project has evolved, these relations have been kept at the centre of attention. But adding new relations to the list require minimal effort, and GeneTUC will try to extract as much knowledge as it possibly can from all sentences, no matter what. What might be considered, is a mechanism for mapping other, synonymical relations injectively into those in Figure 5.1. Information retrieval is simplified if the active vocabulary is kept small but expressive.

The reasons for focusing on interactions between genes and proteins rather than focusing on the genes themselves, are many. The most important of which, is that the genes and proteins direct the processes in the living cell. Understanding how these interactions are conducted is tantamount for understanding how the organism works.

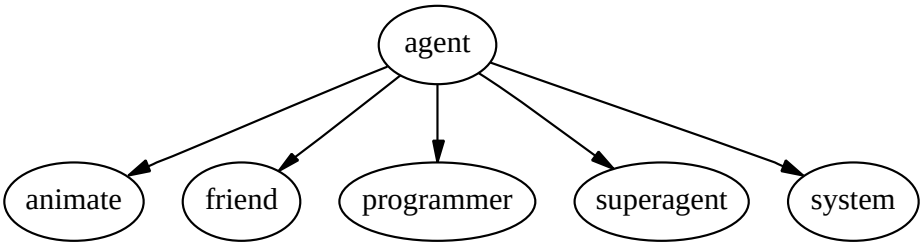

Figure 5.2: TUC's agent subclasses.

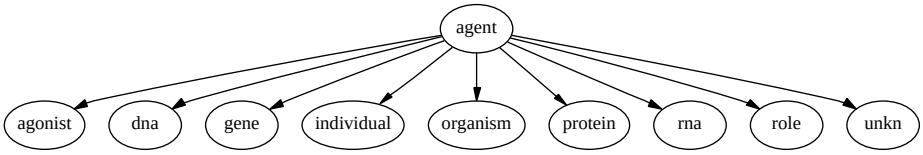

Figure 5.3: Those subclasses of GeneTUC's agent concept not found in TUC.

5.3.2 Agents

In TUC's ontology, only agents and their subclasses are considered to actively pursue goals or trying to attain effects. Agents are thus thought of as mainly humans and animals, and the roles they may fill. However, TUC also denotes a system, in the computer system sense, an agent. Although seemingly self-flattering, this is required for answering questions regarding TUC's own operation. The immediate subclasses of agent in the original TUC is shown in Figure 5.2.

In the biomedical terminology, a number of concepts are treated as though they are agents, behaving actively. In the following example, a mutant protein *prevents* adipogenesis, the genetic command for fat production:

> Furthermore, the mutant protein prevented thiazolidinedione-induced adipogenesis in 3T3-L1 cells, whereas expression of recombinant wild-type PPARgamma2 promoted adipogenesis.

Preventing something from happening is normally considered an act of intent, and it would not be unreasonable to say that only sentient beings are capable of acting on intent. One could expand the semantic base saying that proteins are allowed to prevent, and leave the proteins elsewhere in the ontology. This would require a large number of such additions to be made, not only for quite a few protein - action verb combinations, but also for DNA, RNA, genes and so forth. Rather, the notion of agents was extended somewhat, leaving us with a larger class of agents, some of which are shown in Figure 5.3. Note that many of the concepts are placed in multiple loci in the ontology (not shown in the figure).

5.3.3 Unfamiliar words

Early versions of GeneTUC required all words encountered in the input corpus to be known beforehand. This developed to be somewhat of an Achilles'

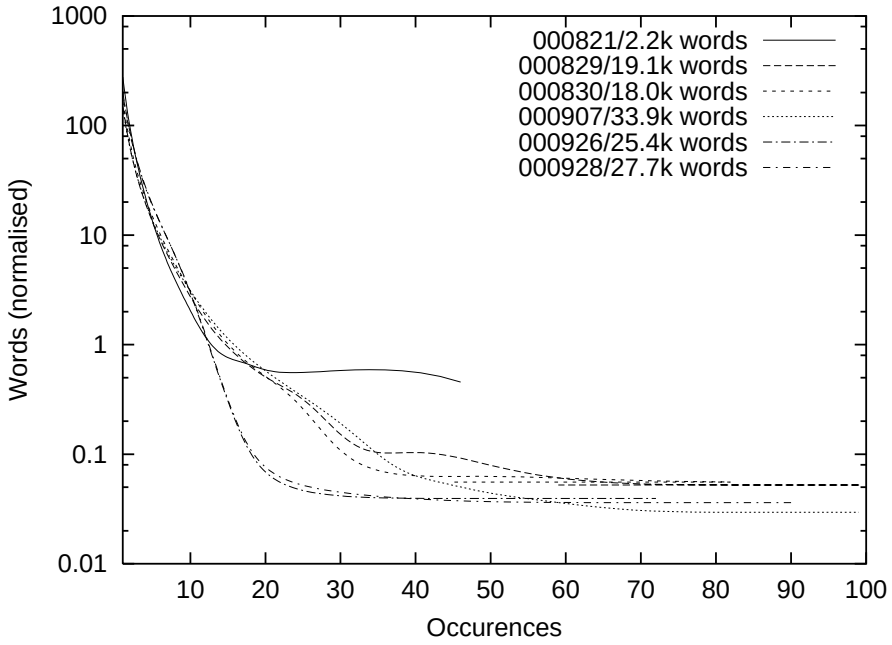

Figure 5.4: The normalised distribution of frequencies of unknown words in the test corpus. On early runs, GeneTUC was terminated before it had processed the input completely, thus the rising number of words.

heel for the system. Scientists come up with new names for genes, proteins and substances when they are discovered. Keeping up with the nomenclature does not seem feasible. The number of *hapax legomena*[7] in the test corpus was very high (Figure 5.4). As the vocabulary was extended, the number of such words remained alarmingly high, as the cost-benefit ratio for adding such words is very low. On inspection, the hapaxes were found to mainly consist of gene and protein names.

Fukuda et al. ([FTTT98]) suggest a method for identifying previously unknown protein names in a text, using an inference mechanism and knowledge on how such substances are named. Although highly successful in its domain, the method is not easily modified to recognising protein and gene names both, and keeping them apart. In Figure 5.3, a concept "unkn" has been introduced as an agent. This concept encompasses all words not recognised by GeneTUC's lexical analysis. According to the presumption that most unknown words in our input are proteins, genes or subclasses thereof, making agent their common ancestor is a satisfactory solution.

Even still, the gene and protein name databases should be kept as up-to-date as possible, as the quality of the output suffers when introducing unrequired generality. The classification of unknowns as agents is merely a temporary remedy and not a permanent solution. In addition, not all unknowns are genes or proteins, some are not even nouns. Therefore, this method must be

[7]From Greek, meaning "words mentioned once", i.e., words encountered only one time in a corpus.

employed with some caution.

5.4 Other changes

The strategy chosen for expanding the semantics, and consequently the grammar, was based on the inclusion-exclusion principle. In short, the semantics was made very general at first, accepting almost all possible combinations of words, although the sentence still had to be grammatically correct. Later, the semantics have be gradually constrained, hopefully giving a higher quality on the output.

5.4.1 Vocabulary

The preliminary version of GeneTUC [And00], contained about 4,800 words, not including domain-specific proper names, i.e., gene and proteins names and their aliases. It was therefore clear that the vocabulary needed expansion, preferably import of dictionaries from existing sources, rather than manual entering of single words.

The WordNet[8] was chosen, due to its availability and size. The complete lists of adjectives and adverbs were imported directly into GeneTUC, adding 43,000 new words to the vocabulary. The words were added using the highest level of generality, that is all adjectives can act on all nouns and all adverbs can act on all verbs, making the semantics less strict. Although this calls for re-evaluation of the semantics on a later stage, it abides to the inclusion-exclusion principle stated above.

Adjectives and adverbs are easily imported, their low semantic significance (in this context) makes it convenient to accept unneeded generality. Nouns and verbs are harder, if the added generality of the semantics is to be kept under reasonable constraints. Unnecessary generality in the ontology is undesirable, thus the adding of nouns will most likely be have to be done by hand. Likewise, verbs have to be added manually to ensure that only the right concepts can act as subjects and objects for a given verb. 1,400 words have been added manually to the vocabulary.

5.4.2 Standard complements

Prepositional and adverbial phrases act as modifiers on the noun, verb and adjectives (see Chapter 3). As numerous such complements may modify each noun, verb or adjective, having to enter them all in by hand is very unwelcome.

Subsequently, the notion of standard complements was resorted to. This means that GeneTUC allows for a number of complements of the form “<Prep> <Concept>”, where <Prep> is a lexeme from the class of prepositions, and <Concept> is a lexeme from the ontology, e.g., “of thing” and “to place”.

[8] Available at http://www.cogsci.princeton.edu/~wn/

5.4.3 Syntactic substitutions

Sometimes it is convenient to perform simple syntactic substitutions in the input. For names in the permanent database, TUC has a feature for mapping long compound phrases into simpler names, e.g., “magnetic resonance imaging” maps into “mri”, which is a *de facto* and widely used acronym for this method of diagnostic imaging. Until recently, TUC had no similar mechanism for concepts, making classes with compound names, like “high density lipoprotein” (hdl), troublesome. As this mechanism works on a syntactic level, prior to parsing, substitutions are not restricted to adjectives and nouns, you could also swap a phrase like “not intelligent” for “stupid”.

5.4.4 Ditransitive verbs

TUC’s grammar has mechanisms for dealing with ditransitive verbs, but these have been temporarily disabled. GeneTUC requires such a mechanisms to work properly, thus it was re-implemented. GeneTUC handles ditransitive verbs by performing a lexical re-phrasing. Phrases like

> John gave Mary a kiss.

is treated as if it actually read

> John gave a kiss to Mary.

In the second phrase, the verb “give” is not in its *bona fide* ditransitive sense. It is rather ordinary transitive, with the prepositional phrase “to Mary” acting as an adverbial complement. The ditransitive verbs can utilise the standard complements commented on above.

6 Results

This chapter presents a qualitative and quantitative analysis of the GeneTUC project. The chapter opens with a few examples, serving to show the capabilities and current state of GeneTUC. The statistics of the success rate of GeneTUC is presented next. The chapter concludes with an investigation of GeneTUC's potential.

6.1 Examples

This section presents some examples of GeneTUC's capabilities. The output is briefly commented on. For an explanation of TQL syntax, see [Bra97].

> E: crp is not a useful marker to predict the outcome of an acute cerebro - vascular event on hospital admission.

```
.....................................................................

(A isa crp,B isa marker,C isa person,D isa outcome,
E isa event,F isa admission,adj/useful/B/G,adj/acute/E/H,
adj/cerebro/E/H,adj/vascular/E/H,adj/hospital/F/I,
nrel/on/event/admission/E/F,nrel/of/outcome/event/D/E,
predict/C/D/J,srel/in/thing/B/J,event/real/J,be1/A/K,
event/real/K,srel/nil/marker/B/K)=>false

.....................................................................
```

This effectively demonstrates how GeneTUC successfully deals with negative declarative sentences in the text. The negation is preserved in the TQL code.

> E: we demonstrated that ce accepted by ldl particles from hdl are secondarily transferred to chylomicrons by cetp.

```
demonstrate/id/that/'I'/sk(97)/sk(98)
event/real/sk(98)
sk(99)isa particle
adj/ldl/sk(99)/A
sk(100)isa hdl
nrel/from/particle/hdl/sk(99)/sk(100)
accept/sk(99)/ce/sk(101)
event/real/sk(101)
ce isa substance
sk(102)isa agent
sk(103)isa chylomicron
cetp isa gene
nrel/by/chylomicron/gene/sk(103)/cetp
transfer/sk(102)/ce/sk(104)
srel/to/thing/sk(103)/sk(104)
event/sk(97)/sk(104)
srel/secondarily/mode/nil/sk(104)
```

This sentence is seemingly correctly interpreted, but some discrepancies are detected on closer inspection:

- Some verb complements (srel) are incorrectly identified as noun complements.
- The agent of transfer is not identified as the CETP gene, but as an unnamed agent.
- The agent sk(102) is not needed.

A more optimal solution would therefore be in the lines of:

```
demonstrate/id/that/'I'/sk(97)/sk(98)
event/real/sk(98)
sk(99)isa particle
adj/ldl/sk(99)/A
sk(100)isa hdl
srel/from/thing/sk(100)/sk(101)
accept/sk(99)/ce/sk(101)
event/real/sk(101)
ce isa substance
sk(103)isa chylomicron
cetp isa gene
transfer/cetp/ce/sk(104)
srel/to/thing/sk(103)/sk(104)
event/sk(97)/sk(104)
srel/secondarily/mode/nil/sk(104)
```

The errors are due to the presently very general semantics. The sketched solution can be achieved by enforcing more strict semantics.

> E: in patients with insulin resistance syndrome troglitazone improved the atherogenic lipoprotein phenotype as well as hyperinsulinemia.

```
.....................................................................
Unresolved reference:  phenotype

sk(156)isa patient
troglitazone isa drug
nrel/with/patient/drug/sk(156)/troglitazone
adj/insulin/troglitazone/A
adj/resistance/troglitazone/A
adj/syndrome/troglitazone/A
sk(157)isa phenotype
adj/atherogenic/sk(157)/A
adj/lipoprotein/sk(157)/A
improve/sk(156)/(sk(157),hyperinsulinemia)/sk(158)
event/real/sk(158)
srel/in/mode/nil/sk(158)
hyperinsulinemia isa unkn

.....................................................................
```

This is an example of how (missing) punctuation can cause GeneTUC to misinterpret. Or, to be more precise, interpret differently. Without a comma between “syndrome” and “troglitazone”, recognising “insulin resistance syndrome troglitazone” as one long name phrase is perfectly correct, although not in accordance with the author’s intention of the sentence. GeneTUC has no amendment for such issues at present.

> E: only 38 unique mutations in the human lcat gene have been reported worldwide.

```
.....................................................................
sk(174)isa set
has/set/cardinality/sk(174)/38
has/set/member/sk(174)/A=>A isa mutation
has/set/member/sk(174)/A=>adj/unique/A/B
has/set/member/sk(174)/A=>nrel/in/mutation/gene/A/lcat
has/set/member/sk(174)/A=>adj/human/lcat/B
has/set/member/sk(174)/A=>adj/reported/A/sk(175,A)
has/set/member/sk(174)/A=>event/real/sk(175,A)
has/set/member/sk(174)/A=>adj/worldwide/A/sk(175,A)
has/set/member/sk(174)/A=>event/real/sk(175,A)

.....................................................................
```

This is perfectly correct, although “reported” could be interpreted as a verb in passive form.

> E: modest elevations of circulating homocysteine are common in patients with vascular disease.

```
.....................................................................

sk(177)isa elevation
adj/modest/sk(177)/A
homocysteine isa amino_acid
nrel/of/elevation/amino_acid/sk(177)/homocysteine
adj/circulating/homocysteine/A
sk(178)isa patient
sk(180)isa illness
adj/vascular/sk(180)/A
has/patient/illness/sk(178)/sk(180)
event/real/sk(179)
adj/common/sk(177)/sk(181)
srel/in/agent/sk(178)/sk(181)
event/real/sk(181)

.....................................................................
```

This is also correct. The fact that “patients have illnesses” is coded into the semantics.

> E: the ldl of hypertriglyceridemic patients was rich in protein and triglycerides had a low content of cholesterol and phospholipids and was smaller than normal thus resembling the atherogenic lipoprotein known as small dense ldl.

```
.....................................................................


--- Sentence is too difficult ---

.....................................................................
```

This sentence, which does not parse, illustrates some of the shortcomings of our strategy. The sentence, as presented above, displays a structure somewhat similar to that of the garden-path sentence[1], but it does in fact lack two commas. A comma after “triglycerides” and “phospholipids” makes the sentence human-readable. Alas, GeneTUC has no knowledge of commas.

> E: although it has been reported that decreased cholesteryl ester transfer protein (cetp) activity results in the formation of small ldl plasma cetp activity was normal in the hyperlipoproteinemic patients and the normalization of ldl characteristics by drug therapy was not accompanied by an increase of cetp activity.

[1] Did you find that triglycerides have a low content of cholesterol and phospholipids? They don’t.

```
.....................................................................

--- Sentence is too difficult ---

.....................................................................
```

This sentence has a rather long initial adverbial, ending at the second occurrence of “cetp”, after which a comma should be placed. As with the previous example, our parsing strategy falls short of parsing such sentences.

For further examples, see Appendix A

6.1.1 Garden paths

Human readers often find garden-path sentences confusing and hard to read, this is also the case for GeneTUC. Garden-paths are grammatically perfectly valid sentences, and the (for human readers) perceived temporary ambiguity is just a deception. One example from Section 3.5;

> E: The computer screens all the entrants.

```
.....................................................................
Unresolved reference:  computer

sk(1)isa computer
A isa entrant=>screen/sk(1)/A/sk(2,A)
A isa entrant=>event/real/sk(2,A)

.....................................................................
```

Garden-paths are demanding for top-down parsers like the one found in GeneTUC. However, as long as the semantics are up to it, garden-paths are no insurmountable tasks for GeneTUC.

6.2 Numbers

GeneTUC’s rate of success was measured using a large collection of abstracts; the kind of input material GeneTUC is aimed at. Measurement was performed at uneven intervals along the course of the project.

6.2.1 The training set

GeneTUC was trained using a collection of abstracts taken from the Medline corpus. Some stats for the training set:

- approx. 1,500 abstracts
- 2,713,435 bytes
- 403,067 words

- 39,041 different words
- 17,568 sentences

Initially, one complete run took approximately 48 hours. The final version, debugged and with a few time bombs removed, used only 12 hours to complete the same task. Successful parses use less time than unsuccessful ones, but with the attained ratio of success, most of the increased performance can be attributed to a leaner and more efficient program.

Effectively, only the initial five percent of the training set were used for active training, because manual updates of the semantics and grammar were performed by a sequential but not complete run-through of the input. This file therefore constitutes both a training and test set for GeneTUC.

6.2.2 Success rate

The rate of success at the start of the project was discouragingly low. The first run parsed only one sentence successfully, namely the following:

> E: This increase was dose and time dependent.

```
.....................................................................

Unresolved reference:  increase

sk(1)isa increase
adj/dose/sk(1)/sk(2)
event/real/sk(2)
adj/time/sk(1)/sk(2)
adj/dependent/sk(1)/sk(2)

.....................................................................
```

which does not give any useful information at all, as long as we do not know what the “increase” refers to.

The development of the success rate, shown in Figure 6.1, was therefore all the more encouraging. The success rate shows exponential growth. This exponential growth is regretfully only temporary. The growth rate will no doubt eventually decrease, making the success rate form a sigmoid, but it is difficult to anticipate when this will occur. Meanwhile, the development leads us to believe that the chosen focal points of the development effort have been the right ones.

6.2.3 New material

What was surprising, but also reassuring, was how the successful parses were distributed throughout the input corpus. One would expect that the bulk of the successful parses were at the beginning of the corpus, in the part where the training had taken place. But, as Figure 6.2 clearly shows, the successful parses are distributed evenly all over the corpus, suggesting that GeneTUC’s training has been very successful. Not only has it learnt to parse what it was

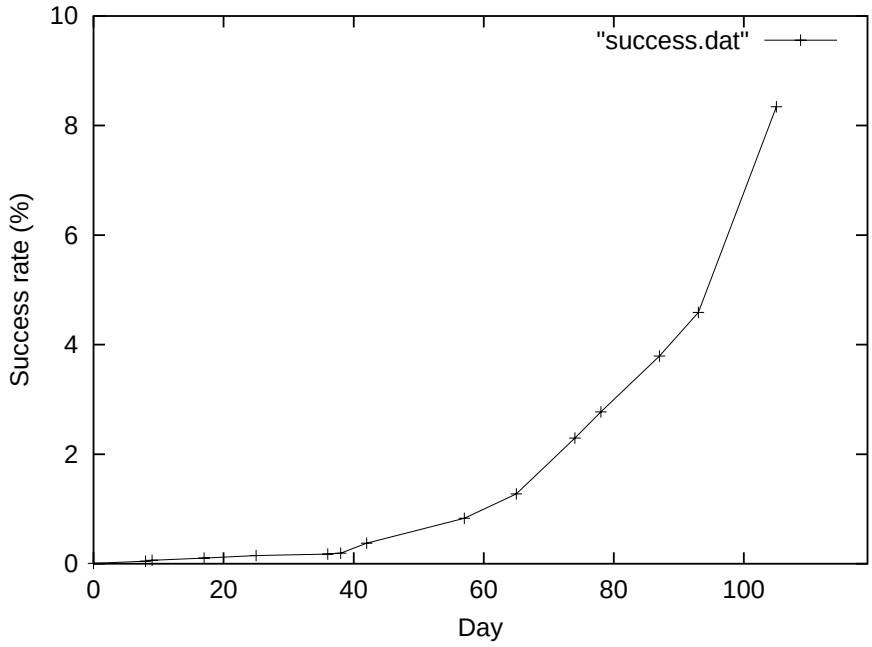

Figure 6.1: The increasing success rate of GeneTUC.

directly trained to do, it is also able to generalise and recognise sentences only similar to the ones in the training set.

The nature of the successful parses has not been investigated. It is hence difficult to say whether the language changes significantly throughout the corpus. The different abstracts do stem from different authors, each having their own personal style of writing. An abstract is not very long, usually about ten to fifteen sentences. Each column in the histogram represents more than a hundred different abstracts, which indicates that GeneTUC can handle many different styles of writing.

6.2.4 Scalability

GeneTUC's vocabulary exceeds BusTUC's with more than one order of magnitude. Thus was GeneTUC a measure of how well the TUC architecture would scale in terms of size, and whether this would have an impact on execution speed.

As noted earlier in this chapter, execution speed was in fact increased by four times during the course of the project. This is attributed to optimisations in program operation. We recorded no indication of the increased size making the application run slower.

We have not studied or tried to estimate the complexity of our top-down parsing technique. Jurafsky and Martin ([JM00], chapter 10) has a superficial discussion of the complexity of different parsers.

In terms of execution size and memory usage, GeneTUC uses less than 30 MB when started, compared to 16 MB for BusTUC. This is as expected, the

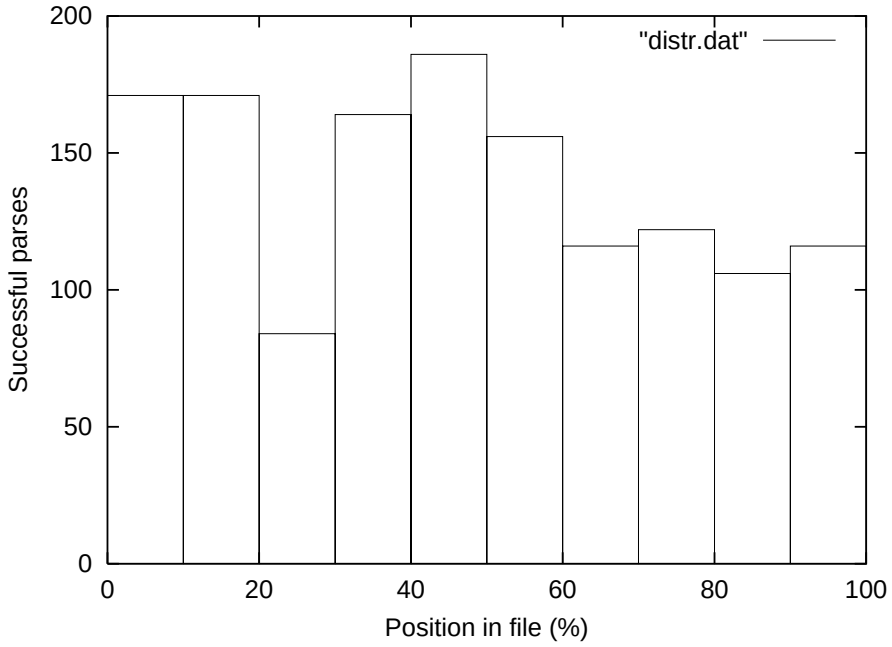

Figure 6.2: Distribution of successful parses in the input corpus.

	Successful	Semantics	Too difficult
Before	18 (45%)	12	10
After	28 (70%)	-	12
Future	33 (82.5%)	-	7

Figure 6.3: Current and projected success rates of GeneTUC.

increased vocabulary taken into account. (Bear in mind that BusTUC has large bus route tables which GeneTUC does not.) GeneTUC's memory usage will naturally rise as the semi-permanent database grows bigger. After being fed with the training set, GeneTUC claimed just below 50 MB of memory.

6.2.5 Potential

To investigate further the potential success rate of future versions of GeneTUC, 40 sentences taken from the training set were inspected thoroughly. Of these, eighteen sentences parsed as is, twelve sentences failed due to incomplete semantics and ten sentences failed due to timeout restrictions (first row of Figure 6.3, labelled "Before").

By manually amending the semantics, without altering or augmenting the grammar, the results in the second row ("After") of Figure 6.3 were obtained. This gave an overall success rate of 70%.

Finally, of the remaining five sentences which did not parse successfully, two were deemed possible by altering the grammar of GeneTUC in a consistent way, thereby obtaining a theoretical overall success rate of 82.5% (third row

of Figure 6.3, labelled “Possible”). The unsuccessful parses were related to punctuation, similar to last two sentences of Section 6.1, and initial adverbial phrases.

Although the statistic foundation for this exercise is thin, it is hopefully an indicator of how well GeneTUC can perform in future version. More thorough inspection is needed to provide more conclusive results.

7 Discussion

The types of errors, or incomplete parses, are roughly divided into three categories. The first category contains errors related to incomplete semantics. Such errors can and will be corrected.

The second category contains errors relating to incomplete or erroneous grammar. These errors are more wicked than the errors in the semantics, because the former needs to retain compatibility with the (correct part of the) existing grammer. Correcting these is therefore time-consuming, but feasible.

Some errors are connected to the architecture of the language. These are not easily amended; some of them are impossible to correct or circumvent without fundamentally changing the premises for the TUC architecture. Needless to say, errors in the first categories are easier to correct than architecture-related errors.

But the TUC architecture provides us with possibilities and potential apparently not found in the immediate competitors. These possibilities are related to taking IE one step further.

7.1 The semantics

Many of the transient errors, i.e., errors that can be amended immediately or in a short time, are caused by the need for efficiency. To avoid parsing taking too long, some pruning of the parse tree is performed, thus failing some sentences which the grammar and semantics otherwise are capable of handling.

The principal source of such errors are the complementing, or modifying, phrases. These phrases, most often taking the form “<Prep> <Concept>” may act on nouns, adjectives and verbs alike, see Section 5.4.2. When numerous complements are cascaded on each-other, it becomes increasingly difficult to correctly and efficiently establish the antecedent of each of these complements. An example:

> LDL (1 less than d less than 1) was isolated by sequential ultracentrifugation from the serum of normolipidemic controls and patients with hyperlipoproteinemia.

This sentence contains four complementary phrases (plus one conjunction), making a large number of different parses possible. Identifying the antecedents

is crucial, as the complements may alter a sentence's meaning, depending on the resolution. If GeneTUC has to backtrack repeatedly to come up with a solution consistent with the semantics and grammar, the parsing might very well become time-consuming. To avoid such sentences from slowing down the parsing, they are aborted if not successful after a set time interval.

Another error, related to the former, occur as a consequence of long and complex adverbial phrases. Theoretically, there is no bound on how large or how many adverbial phrases a single sentence may contain. Adding to this the complexity and polymorphic nature of the adverbial phrase, such fragments are an obvious potential time bomb for systems like GeneTUC. This is also known as the Prepositional Phrase (PP) attachment problem. To achieve acceptable system throughput, not all possibilities of adverbials can be explored.

The creation of a semantic base is an equilibrist's art. Making the semantics too strict will cause rejection of sentences employing unusual modes of expression, thus missing out on information from the input. If the semantics is made too lenient, a lot of irrelevant and potentially erroneous information is extracted.

GeneTUC is primarily aimed at extracting information from sources which have been quality-controlled and moderated. Consequently, the strictness of the semantics is not as crucial as it would, had it been aimed at other sources of input[1]. Still, there is a performance penalty to be paid for making the semantics too lenient, so care has to be exhibited.

A problem which has found a temporary solution is the long noun phrases. Nouns may act adjectively on other nouns, and chains of multiple nouns is not uncommon. How to represent this in logic is, however, unclear. The following sentence is an example:

> The rate of plasma cholesterol esterification by lecithin : cholesterol acyltransferase (lcat) was essentially the same for the two diets .

The phrase "plasma cholesterol esterification" is a concatenation of three fully qualified nouns, simple by themselves. Yet, a problem arises when they stand together like above. One suggested solution, although imperfect, has been to treat concatenations as cascaded complements, in effect:

> The rate of *esterification of plasma of cholesterol* by lecithin : cholesterol acyltransferase (lcat) was essentially the same for the two diets .

or, using right to left evaluation, as:

> The rate of *esterification of cholesterol of plasma* by lecithin : cholesterol acyltransferase (lcat) was essentially the same for the two diets .

However, until final decision is made, the initial part of the concatenation is treated like plain adjectives acting on the final element. The TQL equivalents of the two solutions (using left to right evaluation) are shown in Figure 7.1.

[1]Like bus route queries, where no restriction is imposed on the kind of questions the public may ask, hence the semantics must try to make sense of possibly nonsensical sentences.

```
sk(1) isa esterification                          sk(1) isa esterification
cholesterol isa substance                         adj/plasma/sk(1)/A
plasma isa body_part                              adj/cholesterol/sk(1)/A
nrel/of/esterification/body_part/sk(1)/plasma
nrel/of/plasma/substance/sk(1)/cholesterol
```

Figure 7.1: TQL code for two interpretations of "plasma cholesterol esterification". (Excerpt.)

Naming of newly discovered biomedical entities is not regulated[2], and some clashes between gene names and existing words are bound to occur. This complicates matters severely[3], especially when the genes are given names like "LARGE", "WAS" and "BASE". For the time being, genes with such names are commented out of the permanent database, pending a more permanent solution.

When creating an ontology, some concepts will inevitably fit in multiple loci in the hierarchy. A gene is both an *agent* (see above) and a *sequence*. On encounter of the word gene in the input, GeneTUC has to decide upon which of the two interpretations to use. Once this decision is made, it becomes final in the scope of the sentence. This may create problems in sentences referring directly to more than one sense of a concept. Of the two following sentences, only the second succeeds:

> Cyclin E2 is a gene and it codes for methionine.
> The Cyclin E2 gene codes for methionine.

This is because GeneTUC's semantics say that only sequences are allowed to code for amino acids. In the first sentence, the anaphoric reference after the conjunction is not restricted by the binding of "Cyclin E2" to the *gene* sense in the leading part of the sentence. In the second example, "Cyclin E2" is forcibly bound to the *gene* sense by the inclusion of the word "gene", thus not conforming to the semantics[4].

7.2 The grammar

The TUC architecture is still in an early phase, its origin dating only a few years back. The grammar has been gradually expanded and refined ever since the start, but there is still a long way to go. The difference between TUC's grammar and the theoretical bound of the grammar is large. In its current state, the bulk of the input is rejected by the grammar, even if it may be considered proper English. One example is:

> This study was initiated to test the hypothesis that plasma homocysteine concentrations are increased in insulin resistant individuals .

[2]The HUGO Nomenclature Committee is working on this, but authors still tend to make up their own names.

[3]This is the price to be paid for letting the semantic base become too general. When constraining the allowed semantics, some of these problems are expected to solve themselves.

[4]Whether the semantics are adequate in this instance, is another discussion.

sk(1) isa dinner	sk(1) isa dinner
sk(2) isa agent	adj / made / dinner / sk(2)
make / sk(2) / sk(1) / sk(3)	event / real / sk(2)
event / real / sk(3)	

Figure 7.2: Two parses of "The dinner was made". In the left column, a passive-voice interpretation. The dinner does not make itself, thus an anonymous agent is added. To the right, an active-voice interpretation. The dinner is "made", it is no longer in its original form.

There are two issues here. The first is the infinitive clause "to test" complementing the main verbal "initiated" which has not yet been added to the grammar. The second issue is the *that-clause*; the contents of the hypothesis. Although the entire phrase "the hypothesis that" could be substituted by the conjunction "whether", this is not an optimal solution, and a bit contrary to the concept of strict parsing.

Another "pediatric disease" of GeneTUC's is the incompleteness of the semantics. Sentences in accordance with the grammar do not necessarily map successfully to TQL statements; they rely on the semantics of the sentence to comply with the semantics in GeneTUC.

Coordinating conjunctions in sentences are a challenge to GeneTUC. Conjunctions may connect everything from single words up to complete, independent clauses. Finding exactly which elements are connected by the conjunction can therefore be difficult, not only for GeneTUC, but for humans, too.

The distinction between adjectives and verbs in their passive form is often unclear. For instance, the past participle form of the verb "make" is "made", but this is also an adjective. There is, however, a slight difference in meaning: The verb means "created" or "prepared", while the adjective means "fictitious" or "invented". The following sentence will therefore have (at least) two possible interpretations:

> The dinner was made.

The two interpretations, using TQL, are shown in Figure 7.2. Knowing which of these is the more correct is extremely hard, especially without any additional information. In an application like GeneTUC, the interpretation to the left is preferred, because this retains a direct relationship between two objects, stating implicitly that the dinner had to be made by someone.

7.3 Errors related to the architecture

The NRL is deeply founded in logic. This means that all statements in NRL has to have an equivalent in a logic formula. Some types of conjunctions violate this property, e.g:

> Our results suggested that an abnormal lipid composition *and or or* small particle size might cause a decrease in the receptor affinity of LDL.

The conjunction of two (or more) conjunctions, and the conjunction of prepositions cannot be expressed using the logical foundation of NRL. Sentences like

the one above therefore can not be interpreted in NRL. (Note that the original author probably meant “and/or”.) Furthermore, conjunctions involving any form of ellipsis will cause the system to fail.

NRL has no mechanism for dealing with mid-sentence punctuation, be it commas, dashes or other. If removing the punctuation makes the sentence become ungrammatical, parsing fails. This causes problems, as the text corpus (from Medline) often contains colons, hyphens, dashes and parentheses. GeneTUC’s parser simply removes these characters from the input stream, but that leaves the contents of the parentheses hanging in mid-sentence, so to speak. The contents of parentheses are most often appositions or explanatory elaborations of something mentioned immediately prior to them, and seldom alter the meaning of the sentence significantly. Therefore, GeneTUC was modified to remove the contents of parentheses along with the parentheses themselves.

The problems with commas, or any punctuation which may alter the meaning of a sentence, still prevails. And this is potentially severe. An example with an historical ring to it:

> Hesitate not to kill the weak King Edward is right.

By inserting punctuation at appropriate positions in the sentence, any of three statements may arise:

> Hesitate - not to kill the weak King Edward is right.

The King shall live on, and all those waiting to assassinate him must stand down.

> Hesitate not - to kill the weak King Edward is right.

The king must be overturned. Those weak in spirit must gather themselves.

> Hesitate not to kill the weak; King Edward is right.

The king has ruled that the weak must die.

The original sentence, without punctuation, is triply ambiguous due to two things. First, the negation can bind in one of two directions, either the verb “hesitate” or “kill”. Second, the suffixal phrase may either stand adverbially, reinforcing “kill”, or adjectively to “King Edward”. There is no natural interpretation of the original sentence, it is in fact meaningless as it stands. Thus, GeneTUC will not understand such a sentence.

One suggested solution for the punctuation problem has been to split all sentences at punctuation. This would solve some problems, chiefly sentences containing comma-splices[5]. However, this strategy falls short in most cases, as sentence fragments and dependent clauses seldom are valid sentences.

7.4 Taking IE further

The discussion thus far has been tainted by the fact that GeneTUC’s present rate of success is well below that of competing systems, and that GeneTUC development can be perceived as cumbersome and esoteric. But what must be

[5] Joining two independent clauses by a comma but no conjunction. This is, in fact, a grammatical error.

understood, is that there are strengths inherent in the system which greatly supersede those of the competition. GeneTUC is not content with extracting simple assertions from single sentences, this is only the first in a row of milestones.

While GeneTUC's deep founding in temporal logic imposes some restrictions on its operation and maximum attainable rate of success, it is also the cause of some of its strong points. One of the most prominent is the versatility of logic. TFOL or TQL statements can be converted into other forms by well-defined techniques, be it query languages, databases or generated natural language, although the latter is quite complex[6].

The temporality of the logic is a strong point in itself, and where the true potential of the TUC architecture lies. Using a system like GeneTUC for merely extracting factual assertions from a text is a bit like buying a Ferrari and only driving it in first gear. There are hundreds of horse-powers below the hood waiting to be unleashed, metaphorically speaking. All relations (verbs) in GeneTUC are classified as events, either *real* (factual) or perceived.

E: john is stupid.

```
.....................................................................

john isa man
adj/stupid/john/sk(1)
event/real/sk(1)

.....................................................................
```

refers to a real event (line 3). It is a fact that John is stupid, whereas

E: mary knows that john is stupid.

```
.....................................................................

mary isa woman
know/id/that/mary/sk(3)/sk(4)
event/real/sk(4)
adj/stupid/john/sk(5)
event/sk(3)/sk(5)

.....................................................................
```

expresses Mary's opinion of John, which does not necessarily reflect John's true nature, or even the general perception of his intelligence. Without considering Mary's ability to evaluate other people's level of intelligence (she might very well base her knowledge on extensive tests of measurement of cognitive capacity), the only definite information pertaining to the *real world* stated in the above sentence, is that Mary knows something. Delving further into the epistemics is beyond the scope of this report.

This example is analogous to statements like "our results suggest that..." often encountered in the Medline abstracts. They do not provide conclusive

[6] "*Language understanding is somewhat like counting from one to infinity; language generation is like counting from infinity to one*" - Yorick Wilks.

proof, they only serve as an indication, and should therefore not be considered as definite facts. Still, their content is useful for explanatory purposes and for drawing conclusions.

The database of temporal facts can also be utilised in terms of story summarisation, i.e., creating a short synopsis of the key elements of a larger text. A future GeneTUC may be asked to explain *how* processes work or to *describe* the interaction between entities. For the story summarisation to work, GeneTUC would need to be augmented with a larger database of what is considered common sense. It must also have some way of knowing exactly what it is to explain, and how this should be done. In order to explain the protein synthesis in eukaryotic cells, it needs not only know what actions constitute the protein synthesis, it also needs to know in what order they occur, and how to present this to the asker.

Another premise for story summarisation is to bring the system understanding up from a single-sentence to a block level. The contents of multiple sentences needs to be related to one-another. If the author says she is about to describe the cell cycle, the system must be able to fathom that the next statements do indeed describe the cell cycle, and not just see them as unrelated statements, as GeneTUC (mostly) does today.

TUC has some success in resolving anaphora. Direct reference, like

> John has a dog named Spark. *The dog* is an old German shepherd.

where there is a trivial coherence between the antecedent (a dog named Spark) and the reference (the dog) are currently resolved correctly. This trivial coherence is either generalisation (as above) or explicit class alignment, independent of the concept hierarchy[7]. But the mechanism needs to be enhanced, especially if block level understanding is the aim. Most notably, there must be a way of referring anaphorically to more than just concept, e.g.,

> My house is red. My Ferrari has *the same colour*.

> I assume that Liverpool will win the Worthington Cup this year. *This assumption* is based on *their* recent performance.

The first anaphora of the second example is especially tricky, as it refers to a complete statement, rather than a single word or object. However, the temporal logic proves helpful here also. Provided that there is a way of telling GeneTUC that an assumption is something that is assumed (the details of which have not been explored), TUC could use the "non-real" event generated by "I assume that" (see the "John is stupid" example), and use this as the antecedent of the reference.

Our interpretation of Wittgenstein's dictum is that even though not all statements can be explicitly stated in logic, the intended contents of all statements can. Everything we wish to say can therefore be expressed in an NRL equivalent. But the mapping between the original statement and its NRL equivalent is often not trivial. There is also the question whether the chosen logic (TFOL) has the sufficient expressiveness for the task.

[7] mRNA is an RNA, but it is aligned with the class "template", because the mRNA serves as a template in the proteins synthesis.

8 Conclusions

GeneTUC has matured over the months since its beginning in [And00]. The original system, essentially just BusTUC without the bus tables, had a very restricted vocabulary (less than 5000 words). This vocabulary was primarily aimed at bus route queries and everyday matters, such as “When is the next bus to the airport?” and “What time is it?”. A restriction set upon the development of GeneTUC is that it is to maintain “sideways” compatibility with BusTUC, i.e., the systems are to remain equivalent, bar some of the vocabulary and semantics. Experience has shown that this restriction is not always convenient; separating the two applications further would have been desirable on numerous occasions. A compromise has been to introduce a number of flags which enable or disable different features of the system.

8.1 TUC

Making the GeneTUC system has some secondary effects on the general development of the TUC architecture. Enhancing GeneTUC’s performance in most directions implies enhancing that of TUC’s, due to the restriction on sideways and backwards compatibility (with the original TUC framework). Some of the vocabulary and semantics are, however, domain-specific.

Thus far we have seen remarkably few changes to the grammar. A few errors have been discovered, some of which having severe consequences, but all in all the grammar has kept up with the new semantics in a remarkable way. Some flaws needing to be corrected have been uncovered, and the grammar is still not complete, but it has shown to be very robust and covering a great number of sentence patterns.

Increase in the grammar size has been relatively small, mostly because there have been made very few changes to the grammar. The scalability of the grammar has therefore not really been assessed.

However, the architecture seems to scale very well with the increased vocabulary and semantics. The larger permanent database has not made the application run slower, and the increase in memory usage is well within what is acceptable. With further growth, it may be convenient to distribute the semantics into more files, instead of piling it all up into one huge file. Sicstus Prolog

does not, unfortunately, allow for a single predicate[1] in multiple files, but there are ways to circumvent this restriction.

Even though it has not been impeded by the growth in size, execution speed is still an issue. The parsing time has been improved, but is still to high for high-volume input (our training set is a minute subset of the the abstracts found in the Medline). On the other hand, input needs only be run through the system once, relaxing the speed requirement.

8.2 GeneTUC

Inclusion of external dictionaries and manually adding thousands of words to the vocabulary has sent GeneTUC's results soaring over the last months. Some of this success has come at the expense of a more general semantics, which is obviously not good or wanted. However, it is the author's belief that restricting the semantics later in a top-down manner is easier and more appropriate than creating strict semantics right from the start, mainly due to the better overview the top-down method gives.

In terms of appropriateness, the high growth rate of the number of successful parses imply that GeneTUC may become a useful tool for NL processing of biomedical texts. Nonetheless, the currently low rate of success suggests that much work still lies ahead before the results are comparable to other systems in the same domain. The performance on unseen material is high, which means that using a standard training set and test set technique will successfully develop the application further.

It seems right to focus on the abstracts of articles. Article titles have too little content to provide useful information, and are often ungrammatical (lacking a verbal). Complete articles contain scores of information, but it would be difficult to separate interesting knowledge from "noise". Abstracts have the ideal combination of high information density and relevance, along with linguistic compliance.

A direct comparison of GeneTUC and the competing systems is hard, because our source material regarding these are scarce in terms of numbers, bar recall and precision according to MUC [(DA98]. At present, GeneTUC's recall is substantially lower than that of the others; 8.3% versus 51% for ARBITER [RRH00] and 29% for Highlight [TMO+00] (numbers for EDGAR not available). GeneTUC's precision has not been measured.

One must also bear in mind that the results presented in this report does not do GeneTUC proper justice. The real test of the NL understanding strategy comes when IE is taken beyond extracting simple facts from independent sentences.

[1] e.g., all adjectives.

9 Future work

The chapter suggests some areas of future research and development effort related to TUC and GeneTUC.

9.1 Preprocessor

Adding a preprocessor unit to the TUC architecture has been discussed. This unit would perform substitutions and deletions on a purely syntactic level, thereby simplifying the the material the parser has to work with. For the time being, such changes to the input are performed by the lexical analyser, but isolating this function in a separate module invoked before the lexical analysis seems to be a better solution. This module is possibly best implemented in a language with more efficient string handling than Prolog, like C or Perl.

Acronyms[1] are often encountered in the kind of texts GeneTUC is aimed at understanding. On first use of such acronyms in a text, it is customary to also list the expansion of the acronym along-side, e.g.:

> We studied the effect of dietary olive and corn oil on high-density lipoprotein (hdl) metabolism in golden Syrian hamsters.

It would be convenient to have a mechanism in GeneTUC that "learns" such acronyms along the way by inspecting the contents of the parenthesis. This mechanism may be placed in a syntactical preprocessor.

9.2 Semantics

GeneTUC's semantics is very general. Nonsensical (but grammatically correct) compositions of concepts, relations and modifiers are allowed, in order for the application not to reject any potentially important information. Compiling an initially very general semantics and thereafter refining it is simpler than making the semantics strict from the start. But this means that the semantics need to be particularised on a later stage.

[1] A word formed from the initial letter or letters of each of the successive parts or major parts of a compound term.

For GeneTUC's purpose, creating an injective mapping into a few number of key relationships, thereby simplifying the task of retrieving information later. This is possible to do in the present version, but only by direct substitution of words. There should be some way of substituting complete phrases.

9.3 Grammar

The grammar is far from perfect and therefore needs to be continuously enhanced and amended. One of the currently most prominent problems is nouns acting adjectively on another nouns, for instance "family member" and "protein complex". There is theory which describes how to deal with these compositions. Unfortunately, the methods outlined will probably not work well without impeding the logic foundation of our language.

9.4 TUC

The focus of the GeneTUC project has been the input module. Making the system understand as many types of sentences as possible is key when compiling a knowledge base from free text. Although the grammar and semantics are shared between the input and query module, the query module can be said to be a bit less mature than its counterpart. (For examples, see [And00].) Some work should hence be devoted to making TUC (and, subsequently GeneTUC) able to answer more types of questions.

9.5 What lies ahead

It is important to understand that this effort is only a first step along the way for the TUC architecture. The single-sentence interpretation suffices for BusTUC operation, but a more sophisticated system allowing for story summarisation requires block-level understanding and the ability to group facts together. It also needs a much better notion of common sense than the one found in TUC today. Be that as it may, the employed strategy and the general architecture does allow for moderate optimism on the part of the TUC developers. With substantial research and development effort, TUC might very well rise to become such a system one day.

GeneTUC's potential as a general-purpose NLP interface to other data and knowledge bases should also be investigated. TQL permits paraphrasing into other languages, e.g., SQL, making it ideal as a front end for existing and future knowledge bases.

References

[Amb94] Tore Amble. *Topics in Knowledge-based NLP systems*, chapter Domain Modelling and Automated Natural Language Interfaces. Samfundslitteratur, DK 1970 Frederiksberg C, 1994.

[Amb99] Tore Amble. Logic, events and natural language understanding. In S. Storøy, editor, *Norsk Informatikkonferanse NIK'99*, pages 234–245, 1999.

[Amb00a] Tore Amble. BusTUC - A Natural language Bus Route Oracle. In *ANLP-NAACL 2000 - Applied Natural Language Conference*, May 2000.

[Amb00b] Tore Amble. The Understanding Computer - Natural Language Understanding in Practice, 2000. Lecture Notes.

[And00] Anders Andenæs. GeneTUC /j&'-ne-tük/. Technical report, Department of Computer and Information Science, Norwegian University of Science and Technology, 2000.

[Ask] AskJeeves.com. http://www.askjeeves.com.

[BCK+00] Kenneth Baclawski, Joseph Cigna, Mieczyslaw M. Kokar, Peter Mager, and Bipin Indurkhya. Knowledge representation and indexing using the Unified Medical Language System. In *Pacific Symposium on Biocomputing*, 2000.

[Bra97] Jon S. Bratseth. Bustuc - a natural language bus traffic information system. Master's thesis, Norwegian University of Science and Technology, 1997.

[Cal99] Robert Callan. *The Essence of Neural Networks*, chapter 7-8. Essence of Computing. Prentice Hall Europe, 1999.

[Cas92] Denise Casey. Primer on Molecular Genetics. http://www.ornl.gov/hgmis/publicat/primer/intro.html, 1992.

[CCT] The Cell Cycle and Mitosis Tutorial. http://www.biology.arizona.edu/cell_bio/tutorials/cell_cycle/main.html.

[(DA98] Defence Advanced Research Projects Agency (DARPA). In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*, 1998.

[FTTT98] K. Fukuda, T. Tsunoda, A. Tamura, and T. Takagi. Toward Information Extraction: Identifying protein names from biological papers. In *Pacific Symposium on Biocomputing*, 1998.

[GHb] Writer's Workshop - Grammar Handbook. http://www.english.uiuc.edu/cws/wworkshop/grammarmenu.htm.

[HGP] The Science Behind the Human Genome Project. http://www.ornl.gov/hgmis/resource/info.html.

[HJL98] Hilde Hasselgård, Stig Johansson, and Per Lysvåg. *English Grammar: Theory and Use*. Universitetsforlaget, 1998.

[Hov00] Eivind Hovig. e-Mail, July 2000. Private communication.

[JM00] Daniel Jurafsky and James H. Martin. *Speech and Language Processing*. Prentice Hall, Inc, 2000.

[MED] Medline. http://www.nlm.nih.gov/databases/freemedl.html.

[MT00] David Milward and James Thomas. From information retrieval to information extraction. In *ACL Workshop on Recent Advances in NLP and IR*, 2000.

[Nil98] Nils J. Nilsson. *Artificial Intelligence: A New Synthesis*. Morgan Kaufmann Publishers, Inc, 1998.

[NLP] Natural Language Processing FAQ. ftp://rtfm.mit.edu/pub/usenet-by-hierarchy/comp/ai/nat-lang.

[Pat98] Kevin Paterson. How do readers work out the syntactic structure of sentences? http://ibs.derby.ac.uk/~kpat/R&L/syntax.htm, March 1998.

[Per83] Fernando Pereira. Logic for natural language analysis. Technical report, SRI International, 1983. Revised PhD Thesis.

[RRH00] Thomas C. Rindflesch, Jayant V. Rayan, and Lawrence Hunter. Extracting Molecular Binding Relationships from Biomedical Text. In *ANLP-NAACL 2000 - Applied Natural Language Conference*, 2000.

[RTWH00] Thomas C. Rindflesch, Lorraine Tanabe, John N. Weinstein, and Lawrence Hunter. Edgar: Extraction of Drugs, Genes And Relations from the biomedical literature. In *Pacific Symposium on Biocomputing*, 2000.

[SPT00] Takeshi Sekimizu, Hyun S. Park, and Juniichi Tsjujii. Identifying Interaction between Genes and Gene Products Based on Frequently Seen Verbs in Medline Abstracts. In *Pacific Symposium on Biocomputing*, 2000.

[SR99] Tom Strachan and Andrew P. Read. *Human Molecular Genetics*. BIOS Scientific Publishers, Inc, 2 edition, 1999.

[SS99] Don R. Swanson and Neil R. Smalheiser. Implicit text linkages between Medline records; using Arrowsmith as an aid to scientific discovery. http://kiwi.uchicago.edu/libtrends.html, March 1999.

[TMO+00] James Thomas, David Milward, Christos Ouzounis, Stephen Pulman, and Mark Carroll. Automatic extraction of protein interactions from scientific abstracts. In *Pacific Symposium on Biocomputing*, 2000.

[Wit16] Ludwig Wittgenstein. Tractatus logico-philosophicus, 1914-1916.

A Section 4.1.3 in TQL

The section regarding genes in chapter 4.1.3 is written in GeneTUC-comprehensible NRL. The language may seem naive and simple for the human reader, but it shows how NRL may pass as a "Nearly Reasonable Language". Furthermore, there is no way of knowing whether this effort pushes GeneTUC to its limits; there may still be a large potential yet to be unleashed.

A.1 GeneTUC's output

E: genes are the specific sequence of nucleotide bases in the dna .

.....

```
Unresolved reference:  dna

sk(1)isa gene
sk(1)isa sequence
adj/specific/sk(1)/A
sk(2)isa base
adj/nucleotide/sk(2)/A
sk(3)isa dna
nrel/in/base/dna/sk(2)/sk(3)
nrel/of/sequence/base/sk(1)/sk(2)
be1/sk(1)/sk(4)
event/real/sk(4)
srel/in/sequence/sk(1)/sk(4)
```

.....

E: genes carry the information about heredity .

```
.....................................................................
Unresolved reference:  information

sk(6)isa gene
sk(7)isa information
sk(8)isa heredity
nrel/about/information/heredity/sk(7)/sk(8)
carry/sk(6)/sk(7)/sk(9)
event/real/sk(9)
.....................................................................
```

E: the gene contains the information which is required to construct proteins .

```
.....................................................................
Unresolved reference:  information

sk(11)isa information
adj/required/sk(11)/sk(12)
event/real/sk(12)
srel/in_order_to/thing/sk(13)/sk(12)
sk(15)isa protein
construct/sk(11)/sk(15)/sk(14)
event/real/sk(14)
srel/being_the/reason/sk(13)/sk(14)
contain/sk(6)/sk(11)/sk(16)
event/real/sk(16)
.....................................................................
```

E: the human genome is estimated to comprise 50000 to 120000 genes .

```
.....................................................................
Unresolved reference:  genome

sk(18)isa genome
adj/human/sk(18)/A
adj/estimated/sk(18)/sk(20)
event/real/sk(20)
srel/in_order_to/thing/sk(19)/sk(20)
sk(22)isa set
has/set/cardinality/sk(22)/120000
has/set/member/sk(22)/A=>A isa gene
has/set/member/sk(22)/A=>comprise/sk(18)/50000/sk(21)
has/set/member/sk(22)/A=>srel/to/thing/A/sk(21)
has/set/member/sk(22)/A=>event/real/sk(21)
srel/being_the/reason/sk(19)/sk(21)
.....................................................................
```

E: the gene consists of exons (protein - coding regions) and introns (non - coding regions) .

.....

```
sk(24)isa exon
consist/sk(6)/sk(25)
srel/of/thing/(sk(24),sk(26))/sk(25)
event/real/sk(25)
sk(26)isa intron
```

.....

E: the introns are eliminated from the gene through rna processing.

.....

```
Unresolved reference:  thing

sk(28)isa intron
sk(29)isa agent
eliminate/sk(29)/sk(28)/sk(30)
srel/from/thing/it/sk(30)
event/real/sk(30)
sk(31)isa processing
adj/rna/sk(31)/A
adj/gene/sk(28)/sk(30)
srel/nil/activity/sk(31)/sk(30)
srel/through/place/nil/sk(30)
```

.....

E: an average - sized processed gene spans 3000 base pairs .

.....

```
sk(33)isa gene
adj/average/sk(33)/A
adj/sized/sk(33)/A
adj/processed/sk(33)/A
sk(34)isa set
has/set/cardinality/sk(34)/3000
has/set/member/sk(34)/A=>A isa pair
has/set/member/sk(34)/A=>adj/base/A/B
has/set/member/sk(34)/A=>span/sk(33)/A/sk(35,A)
has/set/member/sk(34)/A=>event/real/sk(35,A)
```

.....

E: genes serve as templates for the synthesis of proteins .

```
.....................................................................
Unresolved reference:  synthesis

sk(37)isa gene
sk(38)isa template
sk(39)isa synthesis
sk(40)isa protein
nrel/of/synthesis/protein/sk(39)/sk(40)
nrel/for/template/synthesis/sk(38)/sk(39)
serve/sk(37)/sk(41)
srel/as/thing/sk(38)/sk(41)
event/real/sk(41)
.....................................................................
```

E: three bases called codons manage the process .

```
.....................................................................
sk(43)isa set
has/set/cardinality/sk(43)/3
has/set/member/sk(43)/A=>A isa base
has/set/member/sk(43)/A=>sk(44,A)isa codon
has/set/member/sk(43)/A=>be_named/A/sk(44,A)/sk(45,A)
has/set/member/sk(43)/A=>event/real/sk(45,A)
has/set/member/sk(43)/A=>manage/A/sk(39)/sk(46,A)
has/set/member/sk(43)/A=>event/real/sk(46,A)
.....................................................................
```

E: this is done indirectly through the use of amino acids and mrna .

```
.....................................................................
Unresolved reference:  set


Unresolved reference:  use

it isa set
sk(48)isa mrna
sk(49)isa use
sk(50)isa amino_acid
nrel/of/use/amino_acid/sk(49)/sk(50)
adj/done/it/sk(51)
event/real/sk(51)
srel/nil/agent/sk(48)/sk(51)
srel/in/activity/sk(49)/sk(51)
srel/through/place/nil/sk(51)
srel/indirectly/mode/nil/sk(51)
.....................................................................
```

E: The rna which is transcribed from the dna in the cell's nucleus resembles a single strand of dna .

.....

```
Unresolved reference:  rna


Unresolved reference:  cell

sk(53)isa rna
sk(58)isa dna
sk(56)isa nucleus
sk(55)isa cell
has/cell/nucleus/sk(55)/sk(56)
nrel/in/dna/nucleus/sk(58)/sk(56)
adj/transcribed/sk(53)/sk(54)
srel/from/agent/sk(58)/sk(54)
event/real/sk(54)
sk(57)isa strand
adj/single/sk(57)/A
nrel/of/strand/dna/sk(57)/sk(58)
resemble/sk(53)/sk(57)/sk(59)
event/real/sk(59)
```

.....

E: this mrna moves from the nucleus to the cytoplasm .

.....

```
Unresolved reference:  cytoplasm

sk(61)isa cytoplasm
move/sk(48)/sk(62)
srel/from/place/sk(56)/sk(62)
srel/to/thing/sk(61)/sk(62)
event/real/sk(62)
```

.....

E: the synthesis of proteins is performed using ribosomes which translate the mrna to proteins .

.....

```
sk(64)isa agent
perform/sk(64)/sk(39)/sk(66)
event/real/sk(66)
srel/during/time/sk(65)/sk(66)
sk(65)isa time
```

```
sk(68)isa ribosome
sk(70)isa protein
translate/sk(68)/sk(48)/sk(69)
srel/to/thing/sk(70)/sk(69)
event/real/sk(69)
use/sk(64)/sk(68)/sk(67)
srel/in/time/sk(65)/sk(67)
event/real/sk(67)
```

.....

E: the genetic code is a series of codons which specify amino acids required to make specific proteins .

.....

```
Unresolved reference:  code

sk(72)isa code
adj/genetic/sk(72)/A
sk(72)isa series
sk(74)isa codon
sk(76)isa amino_acid
sk(77)isa agent
require/sk(77)/sk(76)/sk(78)
event/real/sk(78)
specify/sk(74)/sk(76)/sk(75)
event/real/sk(75)
nrel/of/series/codon/sk(72)/sk(74)
sk(81)isa protein
adj/specific/sk(81)/A
make/sk(80)/sk(81)/sk(79)
srel/in/thing/sk(72)/sk(79)
event/real/sk(79)
event/real/sk(73)
```

.....

E: in laboratories mrna has been isolated .

.....

```
sk(84)isa mrna
sk(85)isa laboratory
adj/isolated/sk(84)/sk(86)
srel/in/place/sk(85)/sk(86)
event/real/sk(86)
```

.....

E: this mrna serves as a template when synthesising cdna .

.....

```
sk(89)isa template
serve/sk(84)/sk(90)
srel/as/thing/sk(89)/sk(90)
event/real/sk(90)
srel/during/time/sk(88)/sk(90)
srel/when/mode/nil/sk(90)
sk(88)isa time
sk(92)isa cdna
synthesise/sk(84)/sk(92)/sk(91)
srel/in/time/sk(88)/sk(91)
event/real/sk(91)
```

.....

E: the cdna may be used for locating genes on a map of chromosomes .

.....

```
may isa month
sk(94)isa agent
sk(95)isa gene
sk(96)isa map
sk(97)isa chromosome
nrel/of/map/chromosome/sk(96)/sk(97)
nrel/on/gene/map/sk(95)/sk(96)
locate/sk(94)/sk(95)/sk(98)
srel/with/thing/may/sk(98)
event/real/sk(98)
adj/cdna/may/A
```

.....

A.2 Some interpretation

By manually examining the TQL code generated by GeneTUC, one can find what it has actually learned from the text. Primarily, we can extract all direct relations (transitive verbs) of interest from the output. For the sake of readability, names have been substituted for Skolem constants, and event references have been removed.

```
carry/gene/information
construct/information/protein
contain/gene/information
eliminate/agent/intron
resemble/rna/strand
perform/agent/synthesis
translate/ribosome/mrna
```

```
use/agent/ribosome
require/agent/amino_acid
specify/codon/amino_acid
make/agent/protein
synthesise/mrna/cdna
locate/mrna/gene
```

The statements should be read as "<Verb>/<Subject>/<Direct Object>, as "the gene carries the information". This way of storing relations makes it simple to later extract information from the semi-permanent database.

Such direct relations may have modifiers, these are marked as "srel" i TQL, e.g.,:

```
move/mrna
srel/from/place/nucleus
srel/to/thing/cytoplasm
```

The mRNA moves (this is the intransitive version, hence no direct object) from the nucleus to the cytoplasm. The modifiers can be cascaded, or joint as in:

```
consist/gene
srel/of/thing(exon,intron)/
```

The gene consists of both exons and introns.

Nouns may also be modified, either through adjectives or through prepositional phrases:

```
resemble/rna/strand
adj/single/strand
nrel/of/strand/dna/strand/dna
```

The noun modifier syntax is "nrel/<Preposition>/<Class 1>/<Class 2>/<Instance 1>/<Instance 2>". The strand is single and it is a strand of dna.

Relations containing sets are almost as simple, although the output is a bit more verbose.

```
sk(34)isa set
has/set/cardinality/sk(34)/3000
has/set/member/sk(34)/A=>A isa pair
has/set/member/sk(34)/A=>adj/base/A
has/set/member/sk(34)/A=>span/gene/A/sk(35,A)
has/set/member/sk(34)/A=>event/real/sk(35,A)
```

This means: The set given the name sk(34) has a cardinality of 3,000, it contains 3,000 elements. If A is a member of the set, then A is a base pair and the gene spans A. In effect, the gene spans 3,000 base pairs.

A.3 Anaphoric reference

The architecture supports resolution of simple anaphoric references, both internal and external. An example from the preceding sentences is:

E: in laboratories mrna has been isolated.

and

E: this mrna serves as a template when synthesising cdna.

The second sentence clearly refers to the mRNA mentioned in the previous sentence. This reference is preserved in the TQL code:

```
sk(84)isa mrna
sk(85)isa laboratory
adj/isolated/sk(84)/sk(86)
srel/in/place/sk(85)/sk(86)
event/real/sk(86)
.
.
.
sk(89)isa template
serve/sk(84)/sk(90)
srel/as/thing/sk(89)/sk(90)
event/real/sk(90)
```

B Strict vs. shallow parsing

This appendix is a very short introduction to two of the most common parsing techniques. First, CFG-based, or *strict* parsing is described, and a small grammar is presented. Second, a *shallow* parsing technique based on HMMs and a small example, is shown.

There have also been some efforts to create parsers using neural networks. The basic methodology is described in [Cal99], but lies beyond the scope of this chapter.

B.1 Strict parsing

The strict parsing regime is founded in logic. Using context-free grammars (CFGs), it is easy to specify the grammatical structure of sentences. A very basic CFG for a subset of English is shown in Figure B.1.

The leaf nodes “Determiner”, “Noun”, “ProperName” and “Verb” unifies with words of the corresponding classes. Note that this grammar does not consider tense or transitivity class for the verbs, and numbers for the nouns.

With our simple grammar, we might find some sentences and try to create valid *parse trees* for the sentences. We will use the following sentences:

John walks.
Mary saw the dog.
Mary saw two dogs.

The two first sentences can be created using the parse trees shown in Figure B.2. The third sentence cannot be made from our grammar, since it does not allow for numbers.

As the term *strict* suggests, this technique places rigorous demands on the input conforming to the parser’s grammar. All possible grammatical structures has to be specified, a very laborious task. But the well-made strict parser is readily extended to larger vocabularies, and produces parses of the highest quality.

```
Sentence      →  NounPhrase VerbPhrase [.]

NounPhrase    →  Determiner Noun
              |  Noun
              |  ProperName

VerbPhrase    →  Verb NounPhrase
              |  Verb
```

Figure B.1: A very simple CFG.

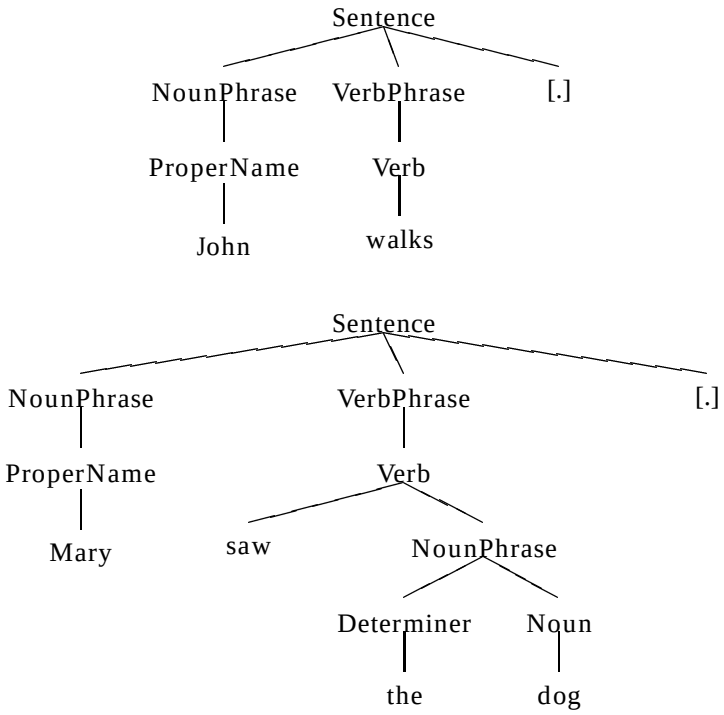

Figure B.2: Parse trees for “John walks” and “Mary saw the dog”.

Lexical likelihood		Tag sequence probability	
P(John|ProperName)	0.01	P(ProperName|_)	0.3
P(Mary|ProperName)	0.01	P(Noun|_)	0.3
P(Dog|Noun)	0.001	P(Verb|_)	0.1
P(walks|Verb)	0.01	P(Verb|ProperName)	0.4
P(walks|Noun)	0.017	P(Noun|ProperName	0.01

Figure B.3: Sample probabilities for a small subset of English. The numbers presented here are fabricated just for this example, and is not based on analysis of actual texts. The underscore (_) represents start of sentence.

B.2 Shallow parsing

Shallow parsers use stochastic part-of-speech taggers, most commonly taggers based on Hidden Markov Models (HMMs). The basis for stochastic taggers is simply to pick the most likely series of tags for a specific sentence. This kind of taggers uses Bayes rule, which has been used successfully in NLP since the late 1950s.

HMMs maximises the probability of a tag being a specific word, given the previous tags in the sentences. Note that the method does not ask which tag is most likely for a given word, it asks "given a word class, what is the probability that it is the specified word":

P(word|tag)*P(tag|previous n tags)

Using the Markov property, the second probability can be rephrased into P(tag|previous tag).

Obviously, this technique requires a large list of probabilities to be readily available to be successful. Some probabilities are shown in Figure B.3.

Let's direct the attention once again to the first sample sentence, and concentrate on the second word, namely *walks*. *Walks* can both be a verb, i.e., the act of walking, and a noun, as in "go for walks". Using the list in Figure B.3, the bigram probability can be computed:

P(walks|Verb)P(Verb|ProperName) = .01 * .4 = .004
P(walks|Noun)P(Noun|ProperName) = .017 * .01 = .00017

Although the noun sense of *walks* is more likely than the verb sense, the low probability of a proper name preceding a noun makes the stochastic tagger disambiguate *walks* correctly.

Compiling a list of such probabilities requires minute analysis of large corpora of text. Compared to the strict parser, the shallow parser will often recognise a larger number of sentence structures, although not necessarily as correctly as the former. This because of the non-zero probability of less common tag sequences. However, the shallow parser may allow sentence structures not necessarily considered grammatically correct, making it more suitable for spoken and informal language.