# Automatic Definition Extraction Using Evolutionary Algorithms

Claudia Borg

Department of Intelligent Computer Systems

University of Malta

Submitted in partial fulfillment of the requirements for a degree of

*Master of Science*

June 2009

**Faculty of Information**
**and Communication Technology**

**Declaration**

I, the undersigned, declare that the dissertation entitled:

> Automatic Definition Extraction Using
> Evolutionary Algorithms

submitted is my work, except where otherwise acknowledged and referenced.

Claudia Borg
June 2009

# Acknowledgements

Several people have crossed their paths along this piece of work. Throughout, I thank Mike Rosner for his guidance and supervision. Once again, it has been a pleasure and I hope it won't be the last.

Within the LT4eL project, I must thank Paola Monachesi for her support, encouragement and patience, and for being an inspiration to how a project should be managed and how people can be motivated. I also thank the Iasi group in Romania, for hosting me during my Erasmus exchange. Thank you Dan Cristea, Corina Forascu, Ionut and Laura Pistol, Adrian Iftene, Diana Trandabat, for offering me support, friendship and office space. I must also thank Mădălina Ioniţă for specifically directing me towards Genetic Programming and giving me the necessary focus. Thanks also to Lothar Lemnitzer, Eline Westerhout, Rosa del Gaudio, Łukasz Degórski, and all those who participated in the machine learning group. Thanks to my Maltese colleagues, Reuben Bartolo and Jimmy Borg, who had to put up with mundane task, and in the process came up with creative tools to ease the way. I cannot mention all the LT4eL people since I would surely leave someone out, so I give a collective thanks to all the project members. I truly enjoyed my work experience on the project, even if it delayed the process of completing this piece of work due to more pressing deadlines!

Within the departments of Computer Science and Artificial Intelligence I thank Sandro Spina, Joe Cordina and Gordon J. Pace for giving birth to this idea and for the several discussions held thereafter. Kevin Vella for allowing me access to the cluster computers which have allowed me to run a large number of experiments.

I thank also my friends, Duncan Paul Attard, Christian Colombo, Steven Caruana, Josef Galea, Anthony Mercieca, Frankie Inguanez, Edward Grech and Stephen Fenech for running some of my initial experiments which required a number of processing days to complete! Thanks for having to put up with a slower PC during those days, it was a significant contribution towards my level of sanity.

During my employment with the University of Malta, I started my days in an empty lone office up in the attic. Slowly, colleagues trickled in. Duncan Paul Attard, Christian Tabone, Christian Colombo, Stephen Fenech, thanks to you all for sharing not only the space, but the despair of being lost. I final see the light at the end of the tunnel, just as I know you have all found yours.

As always, through such times, family is the first stop to vent out frustration. Thanks for putting up with all of that, for understanding that I was too busy at times or for having to accommodate my inflexible timetable. To my father, Dennis, always prepared to point-out my mistakes, both in work and in life, followed with lengthy explanations of how, when, where and why, thanks for the pointing. The lengthy explanations I could easily live without. Thanks to my lovely friends, Kler, Claudine, Sarah and Fiona. You might not have understood one single bit of my work, but you have understood the pressure, and offered the necessary support.

Last, and certainly most, in no way that words can justify or even express, I thank Gordon. Without you this work would not have even started, let alone be complete. Thanks for your support during each hurdle encountered, your unequivocal encouragement, your belief that I am getting somewhere before I even knew where I was. Thanks.

# Abstract

Learning texts contain implicit knowledge such as definitions which provide an explanation of a particular term, or how it relates to other terms. Students assimilate new knowledge about a new topic by referring to such definitions to help them understand and conceptualise new ideas. To help the learning process, definitions could be presented in the form of a glossary which could be queried when new terms are encountered.

Tutors could identify definitions present in their learning material manually, and place them in a glossary for easy reference. However, it is a laborious task to create such glossaries. In this thesis we look at automatic definition extraction from eLearning texts using machine learning techniques. We carry out two main experiments. The first uses Genetic Algorithms to learn weights of a fixed set of features used to identify definitions. These weights give an indication of the level of importance to the respective features, and when used in a definition extraction tool, they can also be used to rank definitions according to the level of confidence. In the second experiment, Genetic Programming is used on a training corpus of definitions and non-definitions, and attempts to learn rules which could be used for automatic classification of sentences in these two classes.

The results achieved are promising, and we show that it is possible for a Genetic Program to automatically learn similar rules derived by a human linguistic expert and for a Genetic Algorithm to then give a weighted score to those rules so as to rank extracted definitions in order of confidence.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

A milestone for any natural language is the creation of its dictionary. A dictionary's importance is in its function, allowing people to look up the meaning of terms in a quick and effective way. A person can expand his knowledge and understanding of a topic by conceptualising new terms encountered and relate them to the knowledge previously gained. Both a general purpose dictionary or a domain specific glossary can be considered as indispensable references. However many introductory texts or books to a topic rarely come with an accompanying glossary since it requires substantial effort to produce, even though most of the definitions would be present in the text itself. In this dissertation we address definition extraction using an approach which extracts definitions from learning material to be presented to a tutor, who would then consolidate the definitions into a glossary to supplement study materials.

## 1.1 Definitions in eLearning

eLearning is becoming more widespread, offering a learning service not only to students who are at different physical locations from the teaching institution, but also to those in a traditional learning environment by providing them with additional support and learning aids to facilitate their learning process. One of the most popular gateways to eLearning is online via the Internet, often through a learning management system (LMS), which allows both students and tutors to manage the learning materials and track their progress. But how helpful is an LMS? Consider the following scenario:

John is a University student, enrolled in several courses spread over the year. Lecturers upload the material covered during a lecture on an LMS, supplemented with additional material such as case study articles or documents which cover technical issues in more detail. Over the weeks, the amount of information continues to increase and, by the end of the course, John is overwhelmed by the material that has been covered. Luckily an LMS allows the tutor to organise the material in sections or topics, so it is organised in a structured manner. John simply has to go through each topic and cover the material available. However sometimes he encounters problems in understanding technical words, or is not sure of the concept being discussed. He is sure he read the word, together with an explanation, in some previous document, but he has forgotten its meaning. How can the LMS assist John in pinpointing the definition he had read?

The process could be quite simple. As the tutor uploads new material, the LMS could propose candidate definitions that it extracts automatically from the text. The tutor can then approve, or modify the definitions for the LMS to save in a glossary. The glossary can be accessed by learners before they attempt to read new material, or can be retrieved by the LMS's search facility when a learner is looking for the meaning of a particular term.

### 1.1.1 What is a Definition?

In this thesis we look at automatic definition extraction from eLearning texts with the aim of assisting learners to conceptualise new terms and help the understanding of new concepts encountered in learning material. So, what is exactly a definition?

There are several ways a term can be defined. One way is by offering a synonym, similar to the function of a thesaurus. For instance, the word *feature* can be also expressed as a *characteristic* or an *attribute.* We can thus conceive the meaning of a word by another one that can replace it, and we understand its meaning by saying that "a feature is an attribute of an object". We can also offer a definition by its *genus et differentia*, where *genus* explains the general class it belongs to, and *differentia* gives the distinguishing factors from other similar classes. For instance,

> A poodle is a dog with thick curling hair.

The first part of the sentence defines a poodle belonging to the class dog, whilst the second part of the sentence gives the distinguishing factors of how a poodle differs from other dogs. Terms can also be defined by enumeration of hyponyms, such as:

> A dog is an animal with four legs which barks, e.g. the poodle, terrier or greyhound.

In this sentence, we are provided with specific examples of things which fall under the class 'dog'. Terms can also be defined by what they are not, such as:

> An animal is a living creature, not a plant, that has senses and is able to move.

In this case we are specifying that an animal is not a plant. Sometimes definitions could be the glossary items of acronyms, such as:

> XML is an eXtensible Markup Language.

Although it is useful to know the full wording of an acronym, it does not necessarily mean that a proper explanation of the term has been provided. In this example, if a person does not know what a markup language is, the full term still does not provide much meaning as to what XML is. Words can also be defined by providing an example of how it is used in a sentence. So in the case of XML, the explanation might have included a sample of XML and an explanation of how it is used and why.

### 1.1.2 Definitional Properties

From the above examples we can see that definitions have several properties, and that the way that a term is explained can differ substantially. Shaw (1922) provides six different ways of constructing definitions, which in themselves highlight the properties that definitions should have.

A definition could be based on the etymology of a word, by showing its derivation; this is done by breaking the word into its original components and deriving the meaning from each of those components. In English this is usually done by tracing the word to Greek or Latin. For example, the word agnostic it made up of two components: the prefix *a-*, meaning without, and gnostic, related to the Greek word *gnōsis* meaning knowledge. The word agnostic thus refers to a person who is non-committal about something (usually used in the context of religious belief) because of lack of knowledge for or against the case.

In order to reduce ambiguity, a definition may have to provide context to the term, providing added meaning through surrounding words within the same sentence. For example, the term *case* has several definitions, some of which are:

- a patient under treatment in medical jargon,
- a legal action under law,
- the lower or upper case of letters in printing.

We clearly require the context of the term within a sentence to understand the particular meaning being referred to.

A definition can be based on the analysis of a term by bringing out the characteristics that it represents. In this way a term can be explained through a description of its appearance, its parts and its purpose. Along with analysis, a term can be described by exclusion, stating what the term is not. This technique is used to alleviate ambiguity for when an erroneous meaning is associated to a word. A term can also be described by providing an example of what its function is and what it can be applied to or not. And finally, a definition by analogy explains the meaning of a term by comparing it to somewhat similar (synonymous) terms. Various combinations of these techniques are commonly used to provide a definition by *genus et differentia* described in the previous section, where a term is first described in its general form and then the definition goes on to describe what sets is apart from other similar terms.

However, it is clear that definitions are not always clear-cut cases, and sometimes we are presented with information that gives some overall meaning of a term, without actually setting out to define it. For example, the following sentence does not really constitute a definition:

> Notebook computers are complete computers with full functionality and with all major devices of a desktop computer.

The problem with such a sentence is that it attempts to define what a notebook computer is by using the term computer, which is itself present in the term being defined. This type of definition is referred to as a circular one where a term is defined by the term itself. Such a definition is vague and does not tell us much about notebook computers. It also does not differentiate a notebook computer from a desktop computer. Definitions must also not be too broad, such as:

> A chair is an object you sit on.

There are several other objects you can sit on (table, floor, sofa, bed) and which are definitely not a chair. Finally, definitions should use more familiar terms to explain the meaning to the term itself. If a definition is aimed towards people with certain technical knowledge, then it is acceptable for it to contain terms used frequently is such specialised domains.

### 1.1.3 Motivation for Automatic Definition Extraction

The identification of what constitutes a good definition is itself a challenge. At times, humans themselves do not agree when they are classifying sentences as definitions, because of different views of what a definition should consist of and how detailed or complete it should be. In automatic definition extraction the challenge goes one step further — how does a human classify a sentence as a definition or a non-definition? And can we produce some process to automatically extract definitions from text?

With so many forms of how a definition is expressed, it is certainly a challenging task to identify the way humans classify definitions. We are able to classify a sentence as a definition if we recognise that the sentence contains information that explains a term. So we are able to conceptualise that a sentence contains definitional knowledge, without necessarily having an understanding of the knowledge expressed. However there are some tips that humans use implicitly. For instance, a sentence which contains the phrase structure *'something is a . . . '* would, usually, classify as a definition. Sometimes the structure of a sentence or its position could also indicate if a sentence could be a definition or not, such as having a colon in the sentence, *Term: explanation*; or the first sentence of a Wikipedia article. Such indicators used by humans could facilitate the automatic process of extracting definitions.

Automatic definition extraction can be a useful tool not only in the eLearning domain, but in other domains such as automatic dictionary creation, question answering systems and ontology engineering. Manual identification of definitions is a time-consuming task. John's tutor may have already spent several hours preparing the necessary materials for a course and planning its structure. It is unlikely that she is prepared to spend more time creating a glossary, unless this is presented to her in an automatic manner. A definition extractor tool can be plugged into a LMS so that such a functionality can be made available to the tutor. In the scenario of a question answering system, we can automatically extract information on a particular question, such as 'What is XML?' The utility of such a tool is the extraction of knowledge, and possibly the ability of manipulating that knowledge for further use, such as in ontology engineering where terms and their meanings are used for automatic ontology creation. The example seen previously, 'A poodle is a dog with thick curling hair.', could result in the creation of two concepts, dog and poodle, where the relation between the two is defined as *type of*, and the concept poodle has an attribute *hair* which is *thick and curling.*

Although there are several end uses for definition extraction, we focus on eLearning where definitions are used to gain knowledge and information on specific topics. Our scenario is to create an easy and usable definition extraction tool which could provide specific and additional knowledge to learners.

### 1.1.4 The Thesis in Context

This thesis began in the context of a larger project, LT4eL — Language Technologies for eLearning, Monachesi *et al.* (2007). LT4eL was a $6^{th}$ Framework Programme project with the aim of facilitating the retrieval of Learning Objects

(LOs) through the use of language technologies and semantic knowledge. One of the tasks within the project involved creating grammar rules that could be used to extract definitions. This proved to be a laborious and tedious task, requiring specialist linguistic knowledge, and the results achieved were discouraging. To try and improve the results, a machine learning group was created to apply different machine learning techniques to the task of definition extraction. Different approaches were taken with the aim of evaluating the potential of different techniques used. This thesis presents the proposed approach taken by the University of Malta, presenting a novel experiment using a particular class of machine learning techniques for the purpose of definition extraction.

## 1.2 Proposed Approach and Results Achieved

Automatic definition extraction presents a challenging situation where we are trying to identify specific knowledge, and there is no hard and fast rule as to how that knowledge is represented or how to encode the machinery to extract that knowledge. One possible approach could be to use linguistic information pertaining to a sentence and try to identify the composition of how a definition is structured through linguistic rules. Another approach could try to attempt machine learning techniques and allow a computer program to automatically learn such rules by providing it with specific examples of both definitions and non-definitions.

### 1.2.1 Existing Solutions for Definition Extraction

Different corpora and techniques have been applied to the task of definition extraction. The most common approach is rule-based, creating rules based on either part-of-speech information, noun phrase chunking or cue phrases, or a mixture of some or all of these components. The work carried out using technical or medical texts achieved far better results than non-technical texts, with the average best precision[1] varying between 60% and 87%. These results are further improved by

[1]Precision is a metric used in Information Retrieval showing the percentage of correctly classified items of those being proposed as candidate solutions through the automatic system.

machine learning techniques, where the highest accuracy[1] rate achieved was by Fahmi & Bouma (2006) on Wikipedia medical texts at 92%. It is obvious from the early stages of our experiment that the corpus from which definition extraction is attempted does influence the results. For instance, sentence position is considered as an important feature in Wikipedia articles, whereas in an eLearning corpus this feature is not always applicable.

In the domain of Question Answering (QA) results vary, but similarly the best results are those that employ machine learning techniques. In the QA domain, one very differing aspect is that the system is only looking for definitions for a particular word or term. In such a context, a system could easily identify all sentences containing that specific word, and thus limiting the search space. In work by Blair-Goldensohn *et al.* (2004), they further apply summarisation techniques so as to present an article format to 'what is' questions. Our work differs in that we attempt to identify all definitions in a particular text, and we have no information as to what head words might possibly be present.

### 1.2.2 Approach Taken by LT4eL

In the context of the LT4eL project, the work carried out and the problems there encountered influenced the direction of the proposed thesis. A set of 450 definitions were annotated to be used as a 'learning corpus' by observing the structure of definitional sentences together with linguistic information (mainly part-of-speech), trying to identify common aspects between the sentences. Grammar rules were executed against a training corpus and a program automatically evaluated each change in the manual grammar rules. The metrics used were similar to other work in this domain, being precision, recall and f-measure. Precision is the percentage showing the quality of the automatically classified definitions, whilst recall is a percentage that shows the proportion of definitions that have actually been captured. F-measure is a formula that gives a weighted average between precision and recall, with the facility to favour one more than the other.

It was noticed that restricting the rules to a few sentences gave higher precision since the rules would capture very few, but correct sentences. On the other hand,

[1]Accuracy is the percentage of correct classifications over the whole corpus.

recall would be low since other definitional sentences were being left out. By generalising the grammar rules to try to capture more sentences immediately resulted in a higher recall (more of the definitional sentences being captured), but a low precision since many of the captured sentences now were not definitional sentences. It is hard to decide where the middle line should be drawn.

Within an eLearning context, where definitions are presented to a tutor and thus the extraction process is semi-automatic with human expert validation, a higher recall was preferred albeit with a low precision.

In LT4eL it was observed that:

1. Identifying grammar rules manually which can be used to detect definitions is difficult and requires experimentation. Results were discouraging, where, for example, in definitional sentences containing the phrase 'is a', a precision of 17% was obtained.

2. The rules contained no ranking knowledge, and thus the candidate definitions were presented to the tutor in the order found in the text, rather than ranking them according to the likelihood of a sentence being considered as a definition or not. This problem is quite crucial since it affects the usability of the system. A tutor who is presented with definitions in a ranked order is more likely to continue using such a system.

Faced with these problems, LT4eL set up a group to attempt machine learning techniques to improve results. The University of Malta, through the work carried out as part of this thesis, proposed a solution which applies evolutionary algorithms in order to solve these two problems.

### 1.2.3 Proposed Solution

We are presented with two specific problems in the process of extracting definitions automatically. The first problem is that of ranking definitions according to a level of certainty in their classification. A possible solution could attempt at attaching some form of likelihood to the rules capturing those definitions, thus resulting in rules containing confidence information on the way they classify sentences. The second problem is to discover automatically candidate rules which could be used to classify definitions. By exploring various rule-possibilities automatically against a training corpus, we might be able to identify such new rules without relying on linguistic expertise. The approach to these two problems taken in this thesis is to apply evolutionary algorithms, namely Genetic Algorithms (GA) and Genetic Programming (GP), in the attempt of solving rule confidence and rule discovery respectively.

The simplest of the two experiments is learning how likely a rule is to capture a definition. For this problem we propose to use a GA which could learn to allocate weights to a fixed set of predefined features. We can allow a GA to explore different sets of weights and discover which ones would achieve the best results. One of the advantages of learning weights is that we can learn both positive and negative weights, the first indicating that a rule can be used to detect a definition, and the latter indicating that a rule can detect a non-definition. Through these weights we would like to create the ideal dividing line between definitions and non-definitions, capturing and classifying sentences as correctly as possible. Since this takes the form of an experiment, we decided to test different possible fitness functions and selection methods to be able to judge which would fare better than others.

In order to attempt learning new possible grammar rules, we need an experimental setup which could come up with new grammar rules automatically and test them on the training data available. When it finds a rule which can correctly match a sentence, it keeps it as a possibly good rule to be used as a definition identifier. A GP is ideal for this type of experiment, as similarly to GAs, it allows the exploration of possible grammar rules automatically, through which process, it might discovering rules which are able to classify definitions from non-definitions correctly.

The two experiments can be viewed as the learning process whose output will finally be used by a definition extraction tool. Figure 1.1 shows how these algorithms could fit into a complete tool that provides the user with a learning module, and an actual classification module. In phase 1, learning objects are automatically annotated with linguistic features and human annotators tag definitions to produce a training set. Phase 2 would then consist of the GP deriving rules which could capture definitional sentences. The rules are based on linguistic

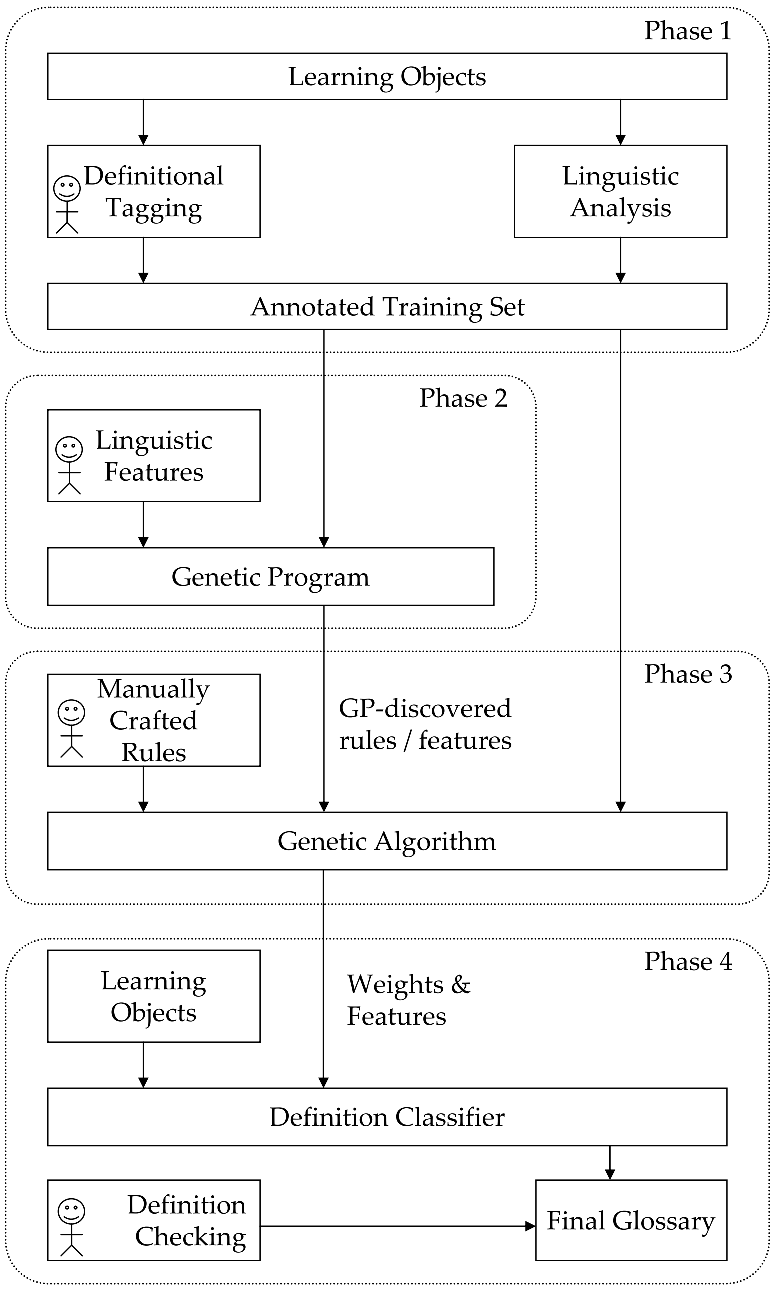

Figure 1.1: Evolutionary algorithms as part of a definition extractor

features inputted by the human annotator, and depend on the type of linguistic tagging carried out in the first phase. For instance, if a part-of-speech tagger is used, then the tags it uses would probably end up as part of the linguistic feature set used by the GP over which the rules are learnt. The rules produced by the GP can then be fed into the GA in phase 3, together with manually crafted rules (optional), so that weights can be associated to the rules. The final phase represents the actual end-user definition extraction tool, whereby together with the rules and weights learnt, it is possible to classify and rank definitions in order of confidence. In an eLearning context, a tutor would check the output of the definition extractor tool and select those sentences which will form part of the final glossary.

The learning phase is a one-time process with the objective of learning a weighted grammar which could then be applied to any document within the same domain of the training corpus. Once the learning phase has been completed, the definition classifier can be used independently. The different phases do not represent a single tool, but rather different components. This thesis focuses primarily on the second and third phase.

### 1.2.4 Results Achieved

The experiments were carried out as two separate tasks. The GA aims at learning weights of a set of features which can be used to identify definitions. When applied to the extraction process, the weights act as a confidence indicator, ranking definitions by allocating a score to the sentence. The first experiment focuses on testing different possible settings for the GA, specifically trying out different selection methods and fitness functions. Overall, through the application of the weights on a basic set of simple features, we successfully improve precision from 17% (achieved in the manually crafted rules) to over 62%. Whilst carrying out a qualitative evaluation on the classification of definitions, we discover certain errors in the manual annotation of definitions within the corpus which affect our results. Some sentences were wrongly annotated as definitions, whilst other sentences which are clearly definitions have been missed during the annotation process, and are thus considered as non-definitions. Notwithstanding this problem, and despite having utilised only a small and simple feature set for the GA, the improvement obtained in precision was very promising.

The GP resulted in a more difficult experiment setting. It is faced with a large search space and the task to learn grammar rules is not a trivial one. Notwithstanding this, the GP did succeed to learn rules that had been identified in the manually crafted rules, and thus could be used as a tool to replace the human expert or to supplement the knowledge required to bootstrap a grammar for extracting definitions. In one of the definitional categories (the punctuation category[1]) the GP succeeded in learning rules that achieved better results than those achieved from the manually crafted rules, raising the f-measure from 17% (manually crafted rules) to 30% (GP-learnt rules).

For other types of definitions, the GP manages to learn rules similar to those already present in the manually crafted rules, but achieves lower results due to lack of linguistic detail in the rules learnt. For instance, the GP learnt (within a few generations) the sequence *noun-is-a-noun*. However, this rule is slightly too generic. By comparison, the manually crafted grammar rules are slightly more complex taking into consideration various surrounding possibilities and phrases. The results achieved by the GP show that it is not a final or stand-alone solution for rule discovery, and indicate that further experimentation could lead to better results. It might also be the case that the features chosen for the learning algorithm were themselves based on what was already observed in the process of the manually crafted rules, thus influencing the rule-learning process towards that direction.

In a final experiment, we ran the GA on a set of rules learnt by the GP in the is-a category[2] in order to associate weights to these rules. The rules on their own, without any confidence measure, obtain an average of 26% f-measure, which is very similar to that obtained through the manually crafted rules. By learning weights to these sets of rules, the f-measure increased to 68%, obtaining

[1]Definitional sentences that contain a punctuation mark such as a colon or a dash as a characteristic.

[2]The is-a category represents those sentences which contain the verb *to be* followed by a determiner, typically being the phrase *is a*.

a 100% precision and a 51% recall. These figures mean that by applying weights to the rules learnt by the GP, we managed to correctly classify just over half of the definitions, without capturing a single sentence incorrectly. The 51% recall indicates that the other half of the definitions were considered as non-definitions by our classifier. This possibly might be because the rules learnt by the GP do not cover the full set of definitions. Considering that this type of result was achieved through a practically fully automatic[1] learning algorithm, with minimal human intervention, this result is a very positive one which requires further exploration.

The results achieved have directed us towards various new questions, giving space for further experimentation. An evolutionary approach is not so frequently applied in Natural Language Processing, and this experiment has shown that it is possible to achieve positive results using such an approach. Further experiments with different corpora will place us in a better position to compare our results with other work reviewed where definition extraction is attempted on a different domain, such as a medical one, where documents and hence definitions are well-structured and might be easier to identify with our techniques. We also propose how these techniques can be used to successfully bootstrap both a corpus of definitions and grammar rules to extract definitions and carry out further experiments.

## 1.3 Outline of Dissertation

In chapter 2 we look at the work carried out in the task of definition extraction for different applications such as glossary creation or question answering systems. We compare various results achieved and methods used for this task, and propose to use an alternative machine learning technique which was not previously applied to definition extraction. We also review work carried out using genetic algorithms and genetic programming in domains containing similarities to our problems, such as grammatical inference, in order to be in a position to address potential problems in the proposed experiments. We provide a tutorial on GAs and GPs in chapter 3, giving the reader enough background to understand the

[1]The only human input the GP requires is a set of linguistic objects to use for the production of the rules, and a training set based upon which it would learn grammar rules.

implications of our decisions when discussing in greater detail the choices made for our experiments. Chapter 4 provides the description of the work carried out within the LT4eL project, which served both as a motivation and a foundation to this work. We then describe in detail the proposal of our experiment and the decisions made for the design and implementation of both the GA and GP in chapter 5. Chapters 6 and 7 describe the different experiments carried out and the results achieved for the GA and the GP respectively. We also evaluate the results from both a qualitative and an quantitative perspective. Chapter 8 describes an additional experiment where the GA is applied to the rules learnt by the GP, in order to evaluate both algorithms as a possible combined tool for definition extraction. Finally, we conclude in chapter 9 by presenting a discussion on the work carried out and observations made which motivate future work in definition extraction.

# Chapter 2

# Background

In this chapter we review definition extraction approaches reported in the literature, and applied to different domains such as glossary creation or question answering systems. We compare different results achieved, types of corpora used and methods applied to this task. We review how results from rule-based techniques are improved through the application of machine learning techniques, and how the corpus can affect the results of the definition extraction task. Since in our experiment we propose to use GAs to learn how to rank the correctness of classified definitions, and GPs to learn new grammatical rules to capture definitions, we also review work carried out in similar areas using these algorithms.

## 2.1 Definition Extraction

The literature review presented in this section looks at various attempts in definition extraction, some of which are aimed at specific applications, such as the automatic creation of glossaries or question answering techniques. When attempting definition extraction this is generally done in a specific domain, with few works trying to identify definitions over generic texts. We analyse the various techniques used, from grammar patterns, cue-phrases, statistical methods to support vector machines, and present a discussion on the different approaches at the end of this section. Compared to some other NLP areas, definition extraction remains relatively unexplored. The review below highlights the issues and difficulties present in the task of automatic definition extraction.

Several of the works reviewed use precision and recall as their metrics in order to evaluate the quality of their results. Precision is the percentage of all sentences being proposed as definitions correctly classified as definitions. Recall is the percentage of all the set of definitional sentences manually marked in the training data correctly classified as definitions[1]. If one looks at the sentences a definition extraction algorithm identifies in a text, one can calculate its precision. However, for every sentence identified by the algorithm, there are typically many other sentences in the corpus which may or may not be definitions. To calculate recall one has to check whether each of the sentences in the corpus is actually a definition. Given the large number of such sentences in a corpus, this is a much more labour intensive task than that of calculating precision[2]. For this reason, some of the results discussed in this chapter uses only precision as their main measure of success. The main problem with this is that an over-restrictive filter may be designed to extract only definitions but resulting in many other definitions remaining unnoticed. This would give a high precision even though other definitions would not be captured.

Another metric used is accuracy, which considers the correct classifications not only of the positive examples, but also of the negative ones. Since in definition extraction we are presented with a rather large set of non-definitions and a relatively small set of definitions, accuracy is not the ideal metric since generally a rather large number of non-definitions will be correctly classified (e.g. classifying 20,000 non-definitions correctly, with 600 of these wrongly classified as definitions would not have a great impact on accuracy. However classifying 600 non-definitions as definitions, when we only have a 100 definitions in our corpus, would have a serious impact on the resulting definition extractor.). Manning *et al.* (2008) argues that the advantage of using precision and recall is that you can trade-off one for the other by using f-measure, which gives a weighted harmonic mean of the two metrics, according to the emphasis required for the retrieval process to be considered a successful one.

[1]Both precision and recall are defined in mathematical terms in section 5.1.1.3 where we describe how the formulae will be used in our work.

[2]In some cases recall might not even make much sense to calculate, such as when the corpus is not a fixed one e.g. the Internet.

### 2.1.1 Grammar and Rule-based Extraction

Work carried out on automatic creation of glossaries usually tends to be rule-based, taking into consideration mainly part-of-speech (POS) information as the main linguistic feature. Park *et al.* (2002) extract glosses[1] from technical texts (such as computer manuals), using additional linguistic information added to the texts to enable them to identify possible glosses. For this purpose they use several external tools, including POS tagging and morphological analysis, to add the linguistic information to the texts. They manually identify the linguistic sequences which could constitute glosses, and describe the sequences as cascade finite-state transducers. Since sequences sometimes also capture non-gloss items, Park *et al.* include post-filtering rules to discard certain forms from the candidate set. These include person and place names, special tokens such as URLs, words containing symbols (except for hyphens and dashes) and candidates with more than six words in length.

Variants are identified and grouped, with one gloss set as the canonical form and others listed as variants (including misspelt items and abbreviations). Finally all glossary candidates are ranked and presented to human experts who select which glosses will be part of the final glossary. In evaluating the candidate glosses, three human experts accepted 228 (76%), 217 (72%), and 203 (68%) out of the top 300 items as valid glossary items. The evaluation does not consider missed glosses which ranked lower than 300 and inter-annotator agreement between the three experts is not discussed. The choice of the top 300, rather than say 200, glosses seems to be arbitrary, possibly expecting experts to go through glosses only until non-gloss items become more frequent in the list. They also do not state the length of the texts and how many gloss items on average are present in their texts.

Muresan & Klavans (2002) propose a rule-based system, DEFINDER, to extract definitions from technical medical texts, which can then be fed into a dictionary. The corpora used consist of consumer-oriented texts, where a medical term is explained in general language words in order to provide a paraphrase.

[1]A gloss provides a short meaning of a term, usually expanding an acronym or specialised terms.

They approach the task by identifying linguistic features present in definitions or synonyms through manual observation. They point out that the structure of definitions might not always follow the *genus et differentia* model and that different styles of writing can be a challenge for the automatic identification of definitions.

DEFINDER first identifies candidate phrases by looking for cue-phrases such as "is called a", "is the term for", "is defined as", or a specific set of punctuation marks which are deemed important for this task. A finite state grammar is then applied to extract the definitions, basing its rules on POS and noun phrase chunking to help with the definition identification process. DEFINDER then uses statistical information from a grammar analysis module to improve the results, which the authors claim that this takes into account the styles for writing of definitions (apposition, relative clauses, anaphora). In this work we see that the automatic identification of definitions is mainly based on the primary identification of certain phrases, and then further filtered through certain rules that reinforce a sentence being a definition (such as its POS structure).

The evaluation of DEFINDER focuses on comparing the classification of definitions extracted from nine articles against those marked by human annotators, obtaining a precision of 86.95% and a recall of 74.47%. Since the end-use of DEFINDER is to provide alternative definitions to medical jargon, a qualitative evaluation focuses on the usefulness and the readability of the definitions proposed in comparison to other online medical dictionaries.

Malaisé *et al.* (2004) attempts definition extraction with the purpose of extracting the semantic relation present in definitions. In this work the extraction is carried out from an anthropological corpus consisting of different formats, and the evaluation corpus is in the field of dietetics, a medical type of resource. They apply lexico-syntactic patterns in addition to cue phrases, focusing on hypernym and synonym relations in sentences. They obtain a precision of 66% and a recall of 36% (recall is calculated over a small random sample of the corpus). The authors conclude that the lexical markers used to extract definitions and their relations are sometimes limited to domain-specific contexts, and thus can be reused only within a particular domain. It is possible to apply the rules to a new domain to try and discover possible new pairs of definitions and relations, however it might not necessarily be as effective as in the domain the rules were created for.

Storrer & Wellinghoff (2006) report on work work carried out in definition extraction from technical texts by using valency frames. Valency frames contain linguistic information indicating what arguments a verb takes, such as object, subject, position and prepositions. Frames are used to match structures of sentences, and thus extracting definitions using rules which are centred around verbs. This is a rule-based expert-driven approach, with all information being provided by human experts (valency frame, definition categorisation). Definitions in technical texts tend to be well-structured, frequently matching crisp rules and thus making such an approach viable. Classifying definitions into categories according to their valency frame allows Storrer & Wellinghoff to concentrate on specific patterns per rule, taking a fine-grained approach rather than a generic 'catch-all' definition extractor. The evaluation was carried out on a corpus containing manually identified definitions, and achieved an average of 34% precision and 70% recall over all the rules created.

Walter & Pinkal (2006) work on definition extraction from German court decisions using parsing based on linguistic information. Their corpus consists of court decisions, restricted to environmental law due to domain specific terminology in legal texts. They identify two styles of definitions: (i) normative knowledge is described as text that "connects legal consequences to descriptions of certain facts", (ii) terminological knowledge consists in "definitions of some of the concepts used in the first descriptions". They discuss issues specific to legal text, where although the text is fairly structured, the decisions themselves could constitute in controversy as often the terms used in legal texts need clarification.

Based on a survey of definitional sentences, Walter & Pinkal observe common structural elements which constitute a definition. They identify them as follows:

1. the element that is defined (the term);
2. the element that fixes the meaning to be given to the term (the definition);
3. the connector — indicating a relation between the term and the definition;
4. a qualification specifying a domain area; and,
5. signal word that serves to mark a definition (`dann` in German, for which there is no equivalent term in English).

The first three structural elements are applicable to several definitions. The fourth one is domain specific being more applicable to legal texts, and the fifth element is present in German but not English, thus language specific.

Walter & Pinkal also observe that definitions contained broad linguistic variations and that simply using keyword or pattern matching would not suffice to extract such sentences. On the other hand, through the recognition of the similarities between the structural elements, and with added linguistic knowledge, the task of definition extraction becomes more effective. Based on these observations, they manually craft 33 extraction rules which retrieve 4,716 sentences as candidate definitions from their corpus. Filtering rules are then applied to bring the number down to 1,342 sentences (28% of the initial 4,716 candidate sentences). In order to evaluate the extracted definitions, 473 sentences were given to two annotators to decide on the correctness of the classification, reaching a precision of 44.6% and 48.6% respectively. Precision improves considerably (over 70%) when the best 17 rules are used out of the 33. Recall is not calculated since the corpus is not annotated with definitions.

Przepiórkowski *et al.* (2007) attempt definition extraction for a Slavic group of languages (Bulgarian, Czech and Polish). They use eLearning texts and apply rule-based techniques to extract definitions. They classify sentences according to linguistic types, such as:

- `NounPhrase` is `NounPhrase`,
- `NounPhrase`: `NounPhrase`,
- `NounPhrase` i.e. `NounPhrase`,

and derive three sets of grammar rules for each language (sharing similarities between them). The best results achieved are for the Czech language, with a precision of 18.3% and a recall of 40.7%. The result for the Czech group is higher through the use of a technique which enables the capture of multi-sentence definitions. The authors claim that since there is no established evaluation methodology for definition extraction, they decided to include sentences which were partially captured as definitions. Thus a grammar rule does not need to capture a definitional sentence in full, but can capture only part of the sentence, and the whole sentence is presented as a definition. By classifying such sentences as well, the results for Czech increase to a precision of 22.3% and a recall of 46%, and for the other languages the improvement margin is higher, but still achieving lower results than Czech. A discussion on the challenges faced from a linguistic point of view is presented, where it is argued that Slavic languages have certain characteristics which result in difficult encoding of simple patterns such as `NounPhrase` *verb* `NounPhrase`. As a result the authors conclude that more generic grammar rules should be applied to increase recall, and employ machine learning techniques to filter out non-definitions at a later stage, thus improving precision.

### 2.1.2 Machine Learning Techniques

Fahmi & Bouma (2006) use medical pages from the Dutch Wikipedia as their corpus, starting with a rule-based approach to definition extraction and then turning to machine learning techniques to try and improve their results. They start by extracting all sentences containing the verb *to be*, together with some other grammatical restrictions, as the initial set of possible definition sentences. These sentences are then manually categorised into definitions and non-definitions. From these sentences they observe which features can be important to distinguish the set of definitional sentences from non-definitional ones.

Features identified include (i) text properties (n-grams, root forms, and parentheses), (ii) sentence position, (iii) syntactic properties (position of the subject of sentence and whether the subject contains a determiner) and (iv) named entities (location, organisation and person). These features, and different combinations of them, are then categorised into attribute configuration files which are fed into three supervised learning methods, namely naïve Bayes, maximum entropy and support vector machine. This work evaluates the improvement achieved in definition extraction by using these machine learning methods and the different feature combinations. The accuracy achieved over the different methods and configuration files varies from 77% to 92%, with the best resulting method being the maximum entropy on the configuration using the first three features described above.

Westerhout & Monachesi (2007a, 2008) experiment with both rule-based and machine learning techniques. They use an eLearning corpus in Dutch which is partially annotate with manually identified definitions. They argue that since the definition extractor is for an eLearning setting and the learning objects tend to be small in size, both precision and recall need to be given importance unlike similar work where only precision is considered. The manually annotated definitions are divided into five different categories that have been identified through observation. Linguistic rules are used to capture a large number of definitions, after which machine learning techniques similar to Fahmi & Bouma (2006) are applied as a filtering technique. Their final results are slightly lower than those of Fahmi & Bouma, with accuracy at 88%. This is to be expected since the corpus used is less structured than Wikipedia articles.

Degórski *et al.* (2008) focus on Polish eLearning material with the purpose of extracting definitions to be presented to a tutor for glossary creation. Initial attempts using manually crafted grammar rules achieve an f-measure of 28%, and thus the authors attempt several machine learning classifiers available in the Weka toolset (Witten & Frank, 2005) to improve results. The techniques used are naïve Bayes, decision trees (ID3 and C4.5), lazy classifier IB1, AdaBoostM1 with Decision Stump and AdaBoostM1 with nu-SVC. In these experiments they report an increase of f-measure to a maximum of 31% (precision at 20% and recall at 69%) with the best result obtained by the ID3 decision tree classifier. Further experiments on the Polish language use Balanced Random Forest (BRF) in Kobyliński & Przepiórkowski (2008). BRF is a machine learning technique for classification using decision trees, where decisions are based on a subset of attributes which are randomly selected and the best attribute for the current tree is then chosen. This technique increases f-measure to 32% (precision at 21% and recall at 69%), with an accuracy of 85%. Both techniques have improved results over manually crafted rules, achieving close results to work by by Fahmi & Bouma; Westerhout & Monachesi.

### 2.1.3 Definition Extraction from the Internet

Klavans *et al.* (2003) look at the Internet as their corpus, focusing mainly on large government websites, and try to identify definitions by genus by using linguistic information. They use POS and noun phrase chunking as part of the linguistic tools to annotate documents, and also use cue phrases to identify possible important information. In this task, several problems are identified, including the format of the definitions and the content in which they are present. Definitions on the Internet can be ambiguous, uncertain or incomplete. They are also being derived from heterogeneous document sources. Another problem encountered is that the Internet is a dynamic corpus, and websites may change over time.

An interesting discussion is presented in how to evaluate a definition extractor, proposing a gold standard for such a type of evaluation, based on qualitative evaluation apart from the standard quantitative metrics. In their evaluation, human annotators were given 25 definitions and are asked to identify the head word and the defining phrase. The inter-annotator agreement can then be calculated according to what was marked. The authors remark that although there is a high agreement on the marking of the head words, there is lower agreement on the defining phrase, with 81% agreement if considering partial phrases (e.g. one annotator marks a full sentence, and another marks only part of it), and even lower if exact matches were only considered. This shows the challenges as to how humans differently perceive sentences to be definitions.

Liu *et al.* (2003) apply a rule-based approach to extract definitions of concepts for learning purposes from the Internet, focusing mainly on computer science topics. They aim to assist learning by presenting learners with definitions of concepts, and the sub-topics or salient concepts of the original topic. Their system queries search engines with a concept, and the top 100 ranked results are retrieved. In order to discover salient and sub-topics, they look at layout information presented in the HTML tags for features such as headings, bold and italic. A rule-based approach is then applied to filter out items which are generally also highlighted in webpages (such as company names, URLs, lengthy descriptions) and stopword removal. They use term frequency to rank the proposed salient or sub-topics.

Definition extraction is then attempted for the concepts and sub-concepts identified in the first phase of their work. The identification is carried out through rule-based patterns, (e.g. {concept} {refers to | satisfies}. . . ). Webpages containing definitions are attached to the concept, and presented to the user in ranked order. The ranking is based on how many concepts and definitions are present in a webpage (the more being available, the higher the ranking as it is considered more informative).

Liu *et al.* also propose a way of dealing with ambiguity, where the term being learnt is too generic and may appear in different contexts (e.g. classification may be used in library classification, product classification, data mining classification, etc.). Again, to differentiate between different contexts, the term is allocated a parent topic, and through the use of document layout structure, a hierarchical structure of topics is built accordingly. Mutual reinforcement is also used to provide further evidence of the hierarchy built, by further searching for sub-topics under a particular concept. This usually identifies information about other salient topics under that same concept, which thus continues to re-ensure that the sub-topics are related and belong to the parent concept. Their evaluation compares the definitions retrieved by their system (achieving 61.33% precision) to the results provided by Google (precision 18.44%) and AskJeeves (precision 16.88%).

It is interesting to note that the definition extraction phase is preceded by a phase of concept identification, simplifying the task of definition extraction. However, the search for definitions is carried out on particular concepts, and not for all definitions contained in a given text. This might result in definitions of other concepts present in parsed documents being lost. The work also does not make any use of linguistic information, since at their level of processing simple pattern matching on particular keywords is sufficient. The web is a very large corpus and thus can provide many documents containing definitions. However, this can also be a disadvantage affecting the quality of definitions found, similar to the problems encountered by Klavans *et al.* (2003) described earlier — ambiguous, uncertain or incomplete definitions. This is partially addressed by providing several resulting definitions, and documents containing more definitions (which might be more authoritative) are presented first. Quality of definitions is important in any learning phase, and can determine the learner's understanding of a concept — providing wrong or incomplete definitions can be detrimental to a learner's experience.

### 2.1.4 Answering Definitional Questions

Question Answering (QA) is considered part of information retrieval, attempting to provide answers to questions posed in natural language by trying to identify the reply from a collection of documents. There are several areas in which QA systems are applied to, including definitional questions on which the following review will focus. QA systems use the question posed as a starting point to searching for information, and apply linguistic techniques to filter out the necessary content. Some systems use summarisation techniques to provide specific answers in paragraph format rather than short snippets. In either case, we are interested in analysing the techniques used to identify sentences containing the possible definitional information.

Prager *et al.* (2001) attempt a solution to definitional questions to form part of a full QA system. They identify those hypernyms[1] which co-occur with the definition term, and then check with Wordnet how closely related the two terms are. Co-occurring terms with the highest occurrence count and closest relation in Wordnet are ranked highest. They argue that highest ranked co-occurring terms are more likely to be considered acceptable hypernyms and can be presented as a definitional answer. However, not all definitional questions can be answered in this way, since sometimes Wordnet does not contain the term being defined, or does not contain useful hypernyms which could be presented as definitions. Also, in some cases, a hypernym is not the ideal answer as a user might not know what the answer means, and the system risks falling into a cyclic pattern of definitional answering.

Joho *et al.* (2001) observe particular patterns which could extract definitional information or hypernyms not present in Wordnet. Their method extracts all the sentences where the query noun matches, and then rank the sentences according to (i) the sentence position — the earlier it occurs in the document, the more likely it is to be a definition, (ii) word count of the highest occurring words in all candidate answers (excluding stop words), and, (iii) a weight to the pattern itself.

[1]A term is said to be a hypernym of a group of terms where the former denotes a class of objects of the latter. The term ‘animal’, for example, is a hypernym of the terms cat, dog, bird, etc.

Miliaraki & Androutsopoulos (2004) extend on Prager *et al.* and Joho *et al.*'s work just described, and apply it on TREC QA data[1]. Their learning-based method uses a support vector machine (SVM) and they attempt to learn classification of definitions based on four different configurations. An SVM is given a training set of terms to be defined and all 250-character-snippets in which the term occurs in the middle, retrieved from the information retrieval engine. Each snippet is converted into a training vector according to the configurations being learnt. The snippets are manually classified into acceptable definitions or not (3,004 definitions vs. 15,469 non-definitions). The SVM learns to generalise the attributes to be able to classify unseen vectors.

1. The first configuration is based on work by Joho et al. The SVM learns the attributes of (i) sentence position, (ii) word count, based on the most words appearing in all snippets, and (iii) a weight to each manually crafted patterns, 13 in all.

2. The second configuration takes the above attributes, but also includes an additional binary attribute to show whether the snippet contains one of the best hypernyms or not.

3. The third configuration adds extra binary attributes to the second, each corresponding to an automatically acquired pattern. The patterns considered are the n-grams extracted from documents where the query term occurs before or after. The patterns are ranked according to precision, and the best 100 to 300 patterns are used. The advantage of using n-grams is that domain specific indicators in definitions are also learnt.

4. To evaluate whether the use of Wordnet facilitates finding definitional answers, an additional experiment was done discarding the information of whether or not a snippet contains a hypernym.

The third configuration obtained the best results with 84% of the questions being answered with acceptable snippets.

[1]TREC is the yearly Text REtrieval Conference, with a specific track on QA.

Blair-Goldensohn *et al.* (2004) apply both pattern matching and machine learning techniques towards answering definitional questions from generic corpora containing a range of topics. They then apply summarisation techniques to provide a long format answers. They work towards extracting all possible sentences that contain definitional information, based on a pre-defined predicate set as follows:

1. Genus — a generic description of the category the term belongs to;
2. Species — specific information which distinguishes the term from other terms in the same category;
3. Target partition — dividing the term into sub-parts;
4. Cause — the cause of something;
5. History — historical information about the term;
6. Etymology — information on the term's origin;
7. Non-specific definitional — any type of information related to the term (this can be seen as a superset of all the above categories, and more).

In order to learn the above predicates, 14 terms were selected and 5 documents were retrieved from the web for each term. The texts were manually annotated according to these predicates and used as training data. Two approaches were considered to learn the categorisation of sentences according to the predicates. The first uses machine learning techniques to learn feature-based classification and the second uses manually extracted lexico-syntactic patterns over the training dataset.

The machine learning technique uses incremental reduced error pruning (IREP) learning algorithm, achieving an 81% accuracy on the training data. The rules learnt take into consideration term concentration and position within a document, and the accuracy is considered sufficient since these sentences will not be presented directly to the user, but rather summarised at a later stage. The second technique concentrates on identifying definitions by genus et differentia using lexicosyntactic patterns. Using the 18 manually crafted rules a precision of 96% over the training set, however recall is not calculated. Since summarisation is being applied to the extracted sentences, the final evaluation focuses on the readability and quality of the summaries provided as answers.

Sang *et al.* (2005) develop two QA strategies that look specifically at medical texts about RSI (Repetitive Strain Injury) in Dutch, based on layout information and syntactic parsing. By restricting the domain to medical texts, the type questions asked and the type of answers expected are identified as follows: treatment, symptom, definition, cause, diagnosis, prevention and other. Type of answers required are also identified as named entities (what does RSI mean?), list (a list of causes), paragraph (a more detailed explanation), and a yes/no type of answer.

The first strategy was a layout-based information extraction system, which extracts the first sentence of each article from a medical encyclopaedia. Through this exercise it was observed that in larger descriptions in encyclopaedias, information is divided into sections with heading containing keywords similar to the question types recognised (treatment, symptoms, causes), and thus further definitions could be extracted from such sections. The second strategy uses syntactic parsing based on dependency relations, with the corpus annotated and stored as dependency parse trees. Patterns were then defined manually to extract information pertaining to definitional, cause and symptom questions. Definitional answers were based on the concept being present as the head word followed by the verb *to be*, and achieved a precision of 18%. Symptom answers were based on the presence of particular phrases (a sign of, an indication of, is recognizable, points to, manifests itself in, etc.), with a precision of 76%. Cause answers were based on the presence of certain phrases (causes, cause of, results of, arises, leads to), extracting 6600 definitions and achieving 78% precision.

The extracted sentences are then made available to a QA system, and if no answer is found within that set, it uses a generic QA system as a fallback option. The results show that these systems do not perform well, and the precision of the combined correct and incomplete answers range between 14% and 12% for the respective systems. Sang *et al.* argue that three main reasons for these results are that (i) these strategies do not contain question analysis, (ii) the corpus is too small (e.g. does not contain any or enough information on the treatment of RSI) and (iii) the fall back generic QA system is not appropriate for medical texts.

### 2.1.5 Summary

In this literature review, rule-based approaches rely heavily on expert knowledge at hand crafting and improving rules for definition extraction. Post-filtering rules are also used to improve results, as described by Park *et al.*, Muresan & Klavans, and Walter & Pinkal. With the exception of Liu *et al.*, all rule-based or machine learning techniques exploit linguistic information such as POS or named entity recognition in order to abstract away from specific terms and instead focus on the linguistic structure of definitional sentences. Whilst most focus their corpora around technical, legal or medical texts where definitions are generally well-structured, Blair-Goldensohn *et al.*; Kobyliński & Przepiórkowski; Westerhout & Monachesi manage to achieve very positive results in extracting definitions from more generic domains. However, QA systems already have the query term as a starting point and thus need only consider sentences containing the query term. This feature of QA systems could explain why identification of definitional sentences produces more positive results.

Identifying glosses in the work of Park *et al.* can be seen as a subset of identifying full definitions. Whilst achieving an average precision of 72%, their system is aimed at providing semi-automatic glossary creation, suggesting possible glosses to a human expert. For their task, they consider this result as satisfactory. Muresan & Klavans point out various difficulties that are encountered when attempting definition extraction, especially since writing styles can differ. DEFINDER works on medical texts and the evaluation is limited to a set of 53 definitions, which is considerably small and might explain the high precision and recall obtained. On the other hand, the results obtained by Storrer & Wellinghoff; Walter & Pinkal; Westerhout & Monachesi indicate that relying solely on grammatical rules is not enough. This is further substantiated in various works carried out, including that of Muresan & Klavans who use statistical techniques over and above grammatical information, and Fahmi & Bouma; Kobyliński & Przepiórkowski; Westerhout & Monachesi who apply machine learning to learn rules for definition extraction.

Fahmi & Bouma manage to increase accuracy substantially from 54% to an average of 90% by introducing an element of machine learning to rule-based definition extraction. One must take into account that the corpus used (Wikipedia medical articles) are well-structured, and definitions tend to be contained within the first sentence or paragraph of the article. This facilitates the identification task of definitions, especially since sentence position is a feature used by the machine learning techniques to identify definitions. They also concentrate on sentences containing the verb *to be*, thus limiting the learning task. Westerhout & Monachesi also succeed in emulating similar results on an unstructured corpus achieving a slightly lower accuracy, but nonetheless re-enforcing the effectiveness of the techniques used by Fahmi & Bouma. Other machine learning techniques such as ID3 in Degórski *et al.* and BRF in Kobyliński & Przepiórkowski also showed an improvement in the classification of definitions. This could indicate that the structure of definitional sentences is paramount to how successful the extraction task is. In the case of medical Wikipedia articles it is relatively easier even for an untrained human to identify definitions. Whilst in eLearning texts definitions might be found in various locations throughout the text, and might not have a well-structured format, as expected in technical and medical texts.

Extracting definitions from the Internet presents further challenges, in that such a system would have to deal with the faults of the web. The content is dynamic, likely to change or become unavailable, and not necessarily authenticated. The last point can cause problems if definitions extracted are incorrect because they state erroneous facts. The problem here is not the definition extractor, but rather the origin of such text, which cannot be detected with automatic methods.

Answering definitional questions has always be a challenge in QA systems. Such systems are usually quite large catering for various types of questions, and containing a module concentrating on definitional type of questions. The advantage in this type of work is that the query phrase contains the starting point for the search of the definition. Thus from an early stage of the search, the system need only consider sentences that contain the term being searched for, or its synonyms.

Sang *et al.* approach QA by extracting all first sentences contained in medical articles on RSI. They consider these sentences as definitions which could possibly answer definitional questions. This system does not perform well when compared to other systems since it limits its answers to this extracted set of ‘definitions’.

It does not try to extract other definitional sentences contained within the rest of the texts which might provide more correct answers for their QA system.

In work by Blair-Goldensohn *et al.* and Miliaraki & Androutsopoulos, we see that approaching definition extraction with machine learning techniques increases precision considerably to over 80%. This is a positive result, reflecting also the increase in accuracy by Fahmi & Bouma (92%). Of particular interest is the work by Blair-Goldensohn *et al.* where they are interested in a wider spectrum of definitions which will finally be summarised into a long format answer.

From the above we can conclude that improving results achieved by rule-based techniques can be done by including machine learning techniques. However, since definition extraction is still relatively unexplored, not many different machine learning techniques have been attempted on this type of task. The purpose of this work is to explore the use of evolutionary algorithms, specifically using Genetic Algorithms and Genetic Programming, for the task of learning rules and ranking of candidate definitions. Thus the next section will review work carried out in areas similar to the task of definition extraction using these techniques.

In table 2.1 we present a summary of the results reviewed above. In the table we abbreviate accuracy to A, precision to P, and recall to R. Languages are also abbreviated (EN — English, DE — German, DU — Dutch, PL — Polish, CZ — Czech). The table shows the corpora used, the techniques employed and the results achieved (if several results are reported, only the best one is shown in the table).

Table 2.1: Summary of Results

| | | | | Metric | | | |
|---|---|---|---|---|---|---|---|
| **Author** | **Corpus** | **L.** | **Technique** | **A** | **P** | **R** | **F** |
| Park *et al.* (2002) | Technical | EN | Rule-based, POS sequence | | 76% | | |
| Muresan & Klavans (2002) | Medical | EN | Cue phrases, rule-based, POS sequence | | 87% | 74% | 80% |
| Malaisé *et al.* (2004) | Medical | EN | Cue phrases, rule-based, POS | | 66% | 36% | 47% |
| Storrer & Wellinghoff (2006) | Technical | EN | Valency frames | | 34% | 70% | 46% |
| Walter & Pinkal (2006) | Law | DE | Rule-based | | 70% | | |
| Fahmi & Bouma (2006) | Medical | DU | Features augmented with ML | 92% | | | |
| Westerhout & Monachesi (2007b) | eLearning | DU | Features augmented with ML | 88% | 80% | 78% | |
| Przepiórkowski *et al.* (2007) | eLearning | CZ | Rule-based, POS sequences, noun phrases | | 22% | 46% | 30% |
| Degórski *et al.* (2008) | eLearning | PL | ML — ID3 | | 20% | 69% | 31% |
| Kobyliński & Przepiórkowski (2008) | eLearning | PL | ML — BRF | 85% | 21% | 69% | 32% |
| Klavans *et al.* (2003) | Internet | EN | POS sequences, noun phrases | 81% | | | |
| Liu *et al.* (2003) | Internet | EN | Rule-based, concept identification | | 61% | | |
| Miliaraki & Androutsopoulos (2004) | QA TREC | EN | SVMs with different settings | 84% | | | |
| Blair-Goldensohn *et al.* (2004) | QA Internet | EN | ML – IREP algorithm | 81% | | | |
| Blair-Goldensohn *et al.* (2004) | QA Internet | EN | Manually crafted rules | | 96% | | |
| Sang *et al.* (2005) | QA Medical | DU | Layout, POS | | 76% | | |

A: Accuracy; P: Precision; R: Recall; F: F-measure; EN: English; DE: German; DU: Dutch; CZ: Czech; PL: Polish

## 2.2 Evolutionary Algorithms

A Genetic Algorithm (GA) (Goldberg, 1989; Holland, 1975) is a search technique which emulates natural evolution, attempting to search for an adequate solution to a problem by mimicking natural selection. By simulating a population of individuals represented as strings (with a particular interpretation) GAs try to evolve better solutions by selecting the best performing individuals (through the use of a *fitness function*), allowing them to survive and reproduce. This is done using two operations called *crossover* and *mutation*. Crossover takes two individuals (parents), splits them at a random point, and switches them over, thus creating two new individuals (children, offspring). Mutation takes a single individual and modifies it, usually in a random manner. The fitness function measures the performance of each individual, which is used by the GA to decide which individuals should be selected for crossover and mutation, and which individuals should be eliminated from the population. This process mimicks the survival of the fittest, with the better performing individuals remaining within the population, and poor performing individuals being removed.

Genetic Programming (GP) (Koza, 1992) is based on GAs, with the difference that the individuals are represented as programs (stored as trees), as opposed to strings as in GAs. In this manner, 'programs' can be evolved through the use of GPs. Although not as widespread in use as GAs, GPs provide an excellent way to retain the evolutionary techniques of GAs and at the same time take advantage of tree representation which may be more appropriate for certain problems. The same types of operations are carried out in GPs, using sub-trees for creating new individuals in the crossover phase, and changing one node randomly during the mutation phase. The fitness function performs the exact same function as in GAs, in that it decides which individuals will be selected to proceed further into the evaluation process.

Concerns in GAs and GPs are similar. The fitness function is generally difficult to derive, and experiments are required to determine the effect of different fitness functions. Other parameters, such as population size and how often mutation and crossover are carried out, also have an effect on the results. With respect to GPs, the tree representation translates into having a larger search space and it is more difficult to implement crossover and mutation. These factors have to be considered, and the literature review below will analyse how such decisions have been taken before.

The task of learning rules is closely related to grammatical inference, in which machine learning techniques are used to learn grammars from data. GAs, and less frequently GPs, are two such techniques which have been applied to grammatical inference. We will thus review work in this area to observe the different techniques used and results achieved. A full overview of GAs and GPs is given in chapter 3, where a detailed explanation of the two algorithms is given.

### 2.2.1 Genetic Algorithms

Lankhorst (1994) describes a GA used to infer context-free grammars from positive and negative examples of a language. New, possibly better performing, grammar rules may be discovered by combining parts of different grammar rules. A discussion is presented on the choice of gene representation, between a binary representation and a high-level representation. It is argued that a bit string can represent many more schemata than higher level representations, yielding to a better coverage of the search space. A bit representation is chosen, with the lower order bits encoding grammar rules on the right hand side of the rule, whereas higher order bits are encoding the left-hand side symbol.

Selection is based on a stochastic universal sampling algorithm, which helps to prevent premature convergence by 'holding back' super-individuals from taking over the population within a few generations. The best individual is also always allowed to survive to the next generation. Mutation allows for a bit in a chromosome to be mutated. However, this operation is given a low probability so as not to change the population randomly. Reproduction is affected by the schema chosen. The crossover point is influenced by how the representation of the rule is expressed. Lankhorst chooses a two-point crossover[1] system, thus allowing right-hand side rules to crossover more easily.

The GA runs different types of experiments, each time attempting to learn a grammar for a particular target language, with the final aim to use the rules learnt

[1]A two-point crossover splits each individual in three parts rather that the standard two.

by the GA to classify positive and negative examples over a language correctly. In total, ten different languages are specified, ranging from matching brackets to a micro natural language. The latter includes simple grammar rules such as a sentence consists of a noun phrase and a verb phrase, a noun phrase can consist of a determiner followed by a noun or else a noun on its own. These grammar rules are kept at a rather simple level, since they do not contain the detailed linguistic information usually used when parsing is carried out.

Since the GA uses both positive and negative examples to learn possible rules, Lankhorst defines a fitness function that uses the correct classification of positive and negative examples. In general, the first fitness function proposed by Lankhorst allows the population to converge into a solution with reasonable results (the number of generations the GA needed to converge to the best solution vary from 17 to over 1000, depending on the grammar being learnt). However, for the micro natural language, the fitness function did not result in a correct grammar. A further adjustment to the fitness function was required, and the GA was trained incrementally, initially training it using only noun phrases, and after the grammar was correctly inferred, verb phrases were introduced into the training data. This procedure resulted in correct classifications of substrings to influence the fitness score, allowing the GA to converge correctly as well for this grammar, converging after just under 2000 generations.

Losee (1996) applies a GA to learn the syntactic rules and tags for the purpose of document retrieval by providing linguistic meaning to both documents and search terms. Individuals in the GA represent a syntactic rule for parsing or a rule containing the sequence of part-of-speech (POS) tags, or a combination of the two. Whilst sequences of POS tags are not attached to particular words (e.g. noun–verb–determiner–verb), syntactic rules attach linguistic information to each separate word (e.g. dog–noun–subject, in ‘dog bites man’). Losee limits the number of rules for each non-terminal symbol to five to allow the experiment to achieve acceptable results given the computational restraints at the time. He also restricts the GA to produce one offspring during the crossover phase. He maintains that, apart from improving computational performance since only one child has to be evaluated, this also introduces superior genes into the population more quickly as it increases the rate of learning. As a fitness function, the GA

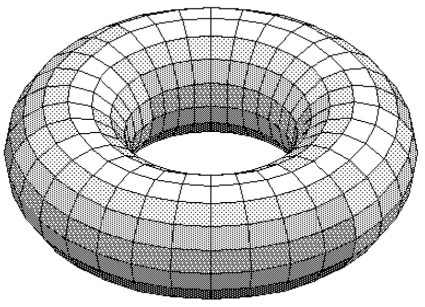

Figure 2.1: Torus formation

uses a weighted function of the resulting ranking of the document and the average maximum parse length, thus measuring performance in terms of how effectively the added linguistic information facilitates document retrieval. The evaluation measures which documents are retrieved and the length of the parse required to retrieve such documents. The GA is reported to improve the quality of the initial randomly generated syntactic rules and POS tags.

Belz & Eskikaya (1998) attempt grammatical induction from positive data sets in the field of phonotactics. The grammar is represented with finite state automata. In this paper two results are produced, one for German syllables and the other for Russian bisyllabic words. The GA is described in detail, including the type of methods selected, and chromosome representation. The GA used is a fine-grained one, where individuals are on a fixed size matrix and are allowed to mate with immediate neighbours. The neighbourhood function is defined as though the individuals lie on the surface of a torus (representation of which is shown in figure 2.1). They argue that an important issue is the representation used for individuals. They present two alternatives:

1. production rules of the form $s_1 \rightarrow as_2$ and $s_1 \rightarrow a$ (where $s$ is a nonterminal symbol and $a$ is a terminal symbol); or
2. a state transition matrix.

They argue that production rules produce more fine-grained genotype representations since the terminals and nonterminals can be represented individually. On the other hand, state transition matrices can be only seen as a whole, each represented by a single cell. The final representation chosen for this work is that of transition matrices. Each individual represents a possible transition matrix which in turn represents the grammar being induced. In order for GAs to be used, the matrix is flattened into one string (one row after the other).

The chosen representation has direct implications on the rest of the GA operations. Crossover and mutation cannot be carried out in the traditional sense of GAs, and certain knowledge must be present in the GA so as to maintain a sound structure of this flat matrix. Belz & Eskikaya manage to discover automata for German syllables and on average for 94% of the Russian noun dataset, achieving rather promising results.

Keller & Lutz (1997) attempt learning context-free grammars through the use of GAs, by learning probabilities to all possible grammar rules. The initial population of the GA is made of all possible combinations of terminals and nonterminals of the form $A \rightarrow BC$ and $A \rightarrow a$, where $A$, $B$ and $C$ are nonterminals and $a$ is a terminal. This guarantees that, although the grammar is a large one, it is finite. There is also no loss of generality, as all possible rules are present in the initial grammar (including those that will not be part of the final solution).

The type of GA chosen for this work also uses a 2D grid representation in torus formation (figure 2.1), where mating occurs only with immediate neighbours. Each individual is encoded as a set of weights, each weight relevant to one parameter. The weight is represented as an n-bit block, and an individual can be viewed as consisting of $M$ blocks of n-bits, where $M$ is the number of all possible rules. Since the grammar is considerably larger than the final solution, Keller & Lutz try to give more importance to zero probability assignment. Thus, the initial bits of the n-bit block is seen as a "binary-switch" as to whether the rest of the bits should be taken into consideration or not.

As for crossover, Keller & Lutz achieved better performance by using a novel genetic operator which they call *and-or crossover* instead of the classical crossover operation. And-or crossover looks at the parents bit by bit, with one child taking the bits produced using conjunction (conservative), and the other child takes the bits produced using disjunction (liberal).

Their experiments focus on 6 grammars (equal number of a$s$ and b$s$, $a^n b^n$, balanced brackets, balanced brackets with two bracket symbols, palindromes over

$\{a,b\}$, and palindromes over $\{a,b,c\}$). They ran the GA for each grammar 10 times, except for the last language which ran only for 3 times due to processing time required. The results show that in the majority of runs the GA was able to learn a grammar which contained all the necessary rules to generate the language. In some instances it would also learn an additional rule with a near zero probability. In other cases where the GA does not converge to a suitable grammar, it is terminated after a number of generations. In such cases Keller & Lutz suggest that this is due to the presence of local maxima around which the population would have converged.

Spasić *et al.* (2004) aim at classifying biomedical terms into specific classes, which represent concepts from an ontology in the biomedicine domain. In order to derive the possible class of a term, they look at the surrounding context of that term. This context is learnt through data mining, extracting the contextual patterns surrounding the terms. The patterns contain morpho-syntactic and terminological information and are represented as generalised regular expressions. Each contextual pattern is then given a value indicating its statistical relevance. They use this to remove the top and bottom ranked features since they are considered too general or too rare to play a role in term classification. Class selection of a term is then learnt using a GA. For a particular class, the GA tries to learn which of the contextual patterns are relevant. Each individual in the GA is a subset of contextual patterns and its fitness corresponds to the precision and recall of using these patterns on the training data. Eventually the GA learns a good subset of features which can be used to identify terms in that class.

In their evaluation, Spasić *et al.* manage to obtain higher precision and recall than three other baseline methods used (random rule, majority rule and naïve Bayes), with f-measure standing at 47%. This experiment shows that by introducing linguistic knowledge to the problem of classification, they were able to improve results considerably. They suggest that by exploiting further orthographic and lexical terms (such as suffix), it would be possible to improve precision even further.

### 2.2.2 Genetic Programming

Smith & Witten (1995) propose a GP that adapts a population of hypothesis grammars towards a more effective model of language structure. They discuss grammatical inference using statistical methods, and the problems encountered in their work. They point out that probabilistic n-gram models allow frequent, well-formed expressions to statistically overwhelm infrequent ungrammatical expressions. There is also the problem with allowing probability for unseen data. The zero-frequency problem entails in assigning a small probability to all unseen data, resulting in both ungrammatical n-grams becoming as probable as unseen grammatical ones.

In their work, Smith & Witten aim at finding a context-free grammar which is able to parse and generate strings in English. They consider the choice of representation of the grammar between logical s-expressions or Chomsky Normal Form (CNF), and base their discussion on similar work by Koza (1992) and Wyard (1991). Koza successfully trained a GP using s-expressions to learn a context-free grammar to detect exons of length 5 within DNA strands. Wyard on the other hand, failed to learn $a^n b^n$ using CNF grammars. Smith & Witten decide to base their work on Koza's representation and the population is represented as Lisp and-or s-expressions. Initial experiments showed that certain constraints were required in order for the GP to evolve, including a maximum depth for nesting and a grammar-generator to allow the GP to evolve towards more suitable grammars. With these constraints in place, the GP evolves simple grammars even within two generations, forming simple sentences such as 'the dog saw a cat'. However, the GP is left to run over more generations to achieve a broader exploration of the search space and result in a more efficient grammar. In order to evaluate the performance of the grammar being learnt, after every 25 generations, they introduce new sentences to see if a similar word grouping is achieved. For example, the sentence 'the dog bit a cat' evolved the same grammar as 'the dog saw a cat' within the next cycle, as the authors expected.

### 2.2.3 Grammar Representation

Through the work reviewed in grammatical inference, we note that grammar representation is an important choice in the expression of our problem. In context-free grammars we specify rewrite rules which for a given symbol on the left hand side, to a string in a specified language on the right hand side. Another grammar representation is tree-adjoining grammars (TAG) Joshi & Schabes (1997), where rather than having symbols and strings, trees are used as the representation structure. Modifications to a tree are carried out either by substitution or adjoining. The first specifies that a leaf node is changed with another tree only if its root is the same as the leaf node. The latter inserts an auxiliary tree instead of an internal node in the original tree, replacing that one node by the auxiliary tree itself. TAG is used to derive grammars, and the type of operations specified are similar to those that can be carried out by GPs. Thus such a representation could allow us to specify the required language over which the GP will learn the grammar rules. Being similar to context-free grammars, it might be too rich for our requirements, where a regular language might serve well. It would also result in an increased search space, which could effect the performance of the GP. However, it is still interesting to review the operations defined on TAG which could influence those in the GP.

### 2.2.4 Summary

We have reviewed work using GAs and GPs in the area of grammar induction, which is closely related to our problem of learning grammars which can be used for definition extraction. Work by Lankhorst highlights the importance of the fitness function, effecting the success (convergence) of the algorithm. Lankhorst uses the positive and negative classifications of a grammar to calculate its performance. He eventually arrives at a function that also considers partially correct classification and context of a sentence. The different functions were derived through the various experiments carried out, and for different types of grammars. Fine-tuning a fitness function will be necessary for any experiment with GAs. Losee also basis the fitness function on what is parsed correctly, creating a average parse metric in order to be able to evaluate the fitness of the grammar being learnt. In cases where the problem being learnt is a grammar, the fitness function must take into account what is being parsed correctly, or classified wrongly. This type of measure is very close to precision and recall metrics used in information retrieval, and also in the work reviewed in section 2.1. These types of metric could be reflected in the choice of fitness function when implementing our work.

The representation of the individual is an important aspect in GAs and GPs. In the reviewed survey we observed that certain authors choose to flatten a tree structure and impose restrictions on the algorithm in order to use a GA. It is not clearly stated whether this is a well informed choice or whether the possibility of using a GP is not considered. By imposing restrictions on the GA to treat a string structure as a tree structure could affect the learning process, since these restrictions affect crossover and mutation. The issue of how to represent the problem being learnt, i.e. the encoding of the individual, seems to be overlooked at times, or 'easier' representations chosen. In the work reviewed above, we often note that rather than opting for a tree representation and using a GP for learning a grammar, a flattened bit-type representation is chosen and restricting where a GA can dissect the individual for crossover and mutation operations.

## 2.3 Conclusion

In this chapter we presented a review of work in definition extraction and GAs and GPs. Several aspects were highlighted which could affect our results and the subsequent evaluation process. The first issue is that of the type of corpora used, where most often technical and medical texts, being more structured, result in better results. The application of machine learning techniques tends to improve results considerably, and although evolutionary algorithms have not been applied to definition extraction, we expect to achieve some improvement over the results achieved using manually crafted rules. Within the QA domain we find that since the head word is present in the question, definition extraction has a head start. This can be seen by the results achieved in QA. In our work we focus on extracting all definitions present in a text, and not those pertaining to a particular term. Comparison between the two tasks is difficult, however it is still important to note the techniques used and if these could be applied to definition extraction in general. Another issue is that GPs have not been used as extensively as GAs, and when it was applicable to use GPs authors would select to flatten a tree structure and use a GA with certain conditions rather than keeping the tree structure and using a GP algorithm. Reasons for such a choice are not provided. This also means that literature on GPs is not as extensive as GAs, and focuses more on learning of computer programs.

# Chapter 3

# Genetic Algorithms and Genetic Programming: An Introduction

This chapter provides an overview of the techniques used in Genetic Algorithms and Genetic Programming, presenting the necessary concepts used in the rest of the dissertation. Different aspects of these algorithms are presented including choices and their affect on the overall performance of the algorithms.

## 3.1 Evolutionary Algorithms

Artificial Intelligence (AI) aims at creating systems which are able to react intelligently to the environment. There are as many different definitions for intelligence as there are researchers in the field. However one prevalent view due to McCarthy (2007) is that of intelligence as the "ability to achieve goals in the world." Researchers in this field have been experimenting with several approaches to compute programs which are able to achieve specific goals, including amongst others, formal logic, brain simulation and evolutionary computation. The last is a technique that mimics biological evolution in nature and simulates a process akin to this. An evolutionary algorithm is a population-based optimisation process using operations corresponding to biological reproduction, genetic mutation, and fitness to survive, so as to advance the quality of the candidate solutions.

Genetic Algorithms and Genetic Programming are two strands of evolutionary algorithms differing mainly on the way they represent candidate solutions (individuals of the population). In this chapter we give an overview of these two approaches, presenting concepts and techniques which will be used in our approach.

## 3.2 Properties of Evolutionary Algorithms

An Evolutionary Algorithms (EAs) is a search technique which emulates natural evolution, attempting to search for an optimal solution to a problem by mimicking natural selection. By simulating a population of individuals (candidate solutions), EAs try to evolve better solutions by *selecting* the better performing individuals, allowing them to survive and reproduce. Performance is measured through a *fitness function*, which provides some value of how good a solution is when compared to another. Reproduction is done using two operations called *crossover* and *mutation.* Crossover takes two individuals (parents), splits them at a random point, and combines the first part of the first individual with the second part of the second individual to create the first child. Then combines the second part of the first individual and the first part of the second individual to produce the second child. These children become new individuals in the next generation of the population, and are sometimes referred to as offspring. Mutation takes a single individual and modifies it slightly, usually in a random manner. The fitness function is a measure of performance, through which we quantify how fit an individual is. The process of selection uses the fitness measure to pick those which will be allowed to reproduce and thus progress into the new generation of the population. This process mimics the survival of the fittest, with the better performing individuals advancing to a new generation, and poorer performing individuals being eliminated. An outline of the process described here is presented below in algorithm 1. The key elements of the EA are:

**Search Space** is a plane of unknown size, and at times infinite, consisting of possible different solutions and there is an element of distance between candidate solutions.

**Gene representation** is the way of encoding a proposed solution (individual) to the problem being attempted by the EA.

**Crossover** is the mating process through which two individuals are recombined to produce two new individuals, thus permitting search space exploration by looking at new solutions.

**Mutation** is a disruptive process through which an individual is modified slightly in a random manner, again intended to explore different solutions. Due to its disruptive element, mutation is usually set with a low probability of occurring.

**Fitness** is a function which measures how well an individual fares in attempting to solve a problem. This function is detrimental to the EA's performance since if we are not measuring an individual's strength in the right manner, we might be attempting to solve a different problem than the one actually specified.

**Selection** is a process through which individuals are chosen to reproduce. This process can be discriminatory by allowing only individuals of a certain fitness level to reproduce. Selection is an important technique since it endorses which genes (characteristics of an individual) will move on to the next generations.

**Population convergence** is reached when the population consists of closely similar individuals or solutions, usually having the same fitness value throughout the population. When such a situation is reached it is difficult for individuals to improve further unless it is through some disruptive force such as mutation. This is referred to as reaching a local maximum, where a best solution has been found with the information currently present in the population, but this is not necessarily the best overall solution to the problem, and other better solutions probably exist.

**Termination** is a decision where we consider whether the algorithm should terminate after a fixed number of generation or after the population has converged. In reality, evolution does not terminate and is a continuing process over millions of years. The 'improvement' of individuals can take thousands of years and in general evolution is slow. In EAs we are presented with limited computation and time, and thus several techniques aim at hastening the evolution process.

There are various considerations to be taken when implementing an EA. In this section we will discuss some of the possible implementations and techniques available which are of more importance to our work.

**Algorithm 1** Outline for the Evolutionary Algorithm

```
Generate Random Population
for $i = 0$ to MaxGenerations do
  calculate FITNESS of all individuals
  SELECT individuals for crossover
  carry out CROSSOVER
  carry out MUTATION
  if population converges then
    exit loop
  end if
end for
```

## 3.2.1 The Individuals of a Population and Their Fitness

We stated that an EA consists of a population of individuals which represent candidate solutions. Thus the encoding of an individual (how it is actually programmed) will depend on what type of problem we are attempting to solve. Choosing the right type of encoding is important to the algorithm and its performance, and ideally should present an intuitive way of understanding the problem being tackled by the EA. Different encodings will be explained in further detail later in this chapter.

Since the individual represents a candidate solution to a problem, the EA should have a method to evaluate the individual's performance. This method is referred to as the fitness function. There is no singular fitness function that can be applied to different problems as each fitness function is problem specific.

If we were trying to learn the solution to the linear equation $3x + y = 384$, our individuals would consist of candidate solutions of $x$ and $y$ that when applied to the equation would result in the value 384. In this case the fitness function would be to actually evaluate the equation. However, when EAs are applied to real world problems, we usually have no indication of what the ideal solution is, we only know what the problem is. Therefore, we might not have the ideal fitness function that would result as the desired solution. Often the fitness function is a form of metric which, when applied to a good individual, the result would be close to the sought solution (in our example if $x = 127$ and $y = 2$, applying the equation would give us the result of 383, which is close to the solution being sought; however it is usually unlikely that we use GAs to solve this type of 'easy' problem.). Choosing the most appropriate fitness function for the problem attempted is one of the most difficult choices when implementing a GA — what type of measurement should we use to decide how well an individual fares? The choice made might affect how successfully the algorithm performs.

### 3.2.2 Selection Method

An important feature of EAs is the selection method which is responsible for identifying which individuals which will be chosen to reproduce, and thus which genes will advance to following generations. Ideally, we would like a selection technique which will allow the fitter individuals into the next generation of the population, whilst at the same time continue exploring possible solutions with weaker individuals which might grow stronger in future generations. An inappropriate section method would permit weaker individuals to reproduce on an equal basis as the fitter individuals, thus slowing down the evolution of the population; allowing only fit individuals to reproduce limits the diversity of the population, resulting in an early convergence to the current best individual. There are various selection techniques which have been proposed for EAs which attempt to select individuals in a fair manner to maintain a mix of fitness in the population. The choice of the selection method applied to a particular problem is still an open-ended question (Mitchell, 1998). Thus, we will evaluate some of the different selection methods proposed in EAs, and analyse in which way they facilitate the growth of a population and possible pitfalls they might have.

#### 3.2.2.1 Roulette Wheel

The roulette wheel selection method is a fitness-proportionate technique which was used in Holland's original work on GAs (Holland, 1975). A 'roulette wheel' is divided into *n-slices*, with each slice being proportionate in size to the fitness of each individual in the population (with n-individuals in a population). The 'wheel' is spun $N$ times to select $N$ individuals which will be used as parents for the next generation. One of the problems with this technique is that selected individuals are placed back into the pool for selection, and can be selected again. In a worst case scenario an unfit individual can be selected $N$ times, resulting in the loss of all better individuals and the next population constituting of the same individual. In such a situation, it is difficult for improvement to occur any further as there is no divergence within the new population. The population is said to have converged to a local maximum, with no better solution possible.

#### 3.2.2.2 Stochastic Universal Sampling and Sigma Scaling

In order to avoid such a situation, Baker (1987) proposed an extension of the roulette wheel by giving each individual an *expected value* based on its fitness, at each generation. Each individual is represented proportionately according to the expected value over the wheel. In this method however, the wheel is spun only once, but with $N$ equally spaced pointers to select $N$ individuals for parenthood. Although this method still allows less fit individuals to be selected, fitter individuals will multiply more quickly since their expected value is higher than those of less fit individuals. This can still result in premature convergence and might not allow the GA to search for other possible solutions.

To address such a problem Forrest (1985) experimented with the expected value, with the aim of giving each individual a closer chance of being selected while there is greater difference in the overall fitness of the population. The expected value is calculated using sigma scaling, which is based on the fitness of the individual, the average fitness of the population and the standard deviation of the population fitness. As the population begins to converge, and the average fitness and standard deviation reduce, fitter individuals will have a higher expected value than less fit individuals.

$$\text{ExpVal}(i,t) = \begin{cases} 1.0 + \frac{f(i)-\bar{f}(t)}{2\sigma(t)} & \text{if } \sigma(t) \neq 0 \\ 1.0 & \text{otherwise,} \end{cases}$$

where ExpVal$(i, t)$ is the expected value of individual $i$ at time $t$, $f(i)$ is the fitness of $i$, $\bar{f}(t)$ is the mean fitness of the population at time $t$, and $\sigma(t)$ is the standard deviation of the population fitnesses at time $t$. In the beginning of an EA, the standard deviation will be high since the population will contain individuals with varying fitness. As the population begins to converge towards a solution, the deviation decreases as the individuals become similar in fitness values.

#### 3.2.2.3 Elitism

So far, the selection methods reviewed all allow the possibility of losing the fittest individual in a particular generation. Elitism, introduced by De Jong (1975), forces the selection process to keep an exact copy of the top fittest individuals from the population at each generation. It also allows these individuals to be selected for reproduction. Through this technique we can ensure that the best individuals are not lost through reproduction or mutation. Many researchers have found that this technique improves the performance of an EA substantially, as the best individuals are always kept (Mitchell, 1998). If the best individuals discovered by an EA are kept, we reduce the possibility of converging to a local maximum. Although it is still possible that the search space contains better solutions that the EA still has not discovered.

#### 3.2.2.4 Boltzmann Selection

Boltzmann selection uses a “temperature” variable, which controls the rate of selection according to a preset schedule. The idea of this technique is to vary the weakness or strength of the selection method at different stages of the evolution. The temperature in the beginning of the evolution is set to high, resulting in the selection method to be weak, allowing less fit individuals to reproduce at a close rate of the fitter individuals. Throughout the lifetime of the algorithm, the temperature is gradually lowered and the selection strengthens, becoming stricter and giving fitter individuals a higher chance to reproduce. An implementation is to assign to each individual $i$ at time $t$ an expected value,

$$\mathrm{ExpVal}(i,t) = \frac{e^{f(i)/T}}{\langle e^{f(i)/T} \rangle_t}$$

where $T$ is the temperature and $\langle g(i) \rangle_t$ is the average of function $g$ over the population at time $t$. This allows for search-space exploration which might lead to fitter individuals and aims at avoiding having individuals stuck in a local maximum. However, fitter individuals could be lost in the beginning of the EA lifecycle, when selection pressure is weak.

#### 3.2.2.5 Rank Selection

Proposed by Baker (1985), this selection method considers the rank rather than the fitness value of an individual. Through this technique, if there is a 'super-individual' with a very high fitness value, this difference will not influence the selection as its rank value is one position away from the next ranked individual. In this way we avoid giving a larger share of offspring to a small group of highly fit individuals and allow for a slower convergence of the population. Each individual is ranked in increasing order of fitness, from 1 to $N$. The user chooses the expected value $Max$ for the fittest individual at rank order $N$, where $Max$ must be $\geq 0$. The expected value of individual $i$ at time $t$ is given as

$$\mathrm{ExpVal}(i,t) = Min + (Max - Min)\frac{\mathrm{rank}(i,t) - 1}{N - 1}$$

where $Min$ is the expected value of the individual with rank 1. Baker recommends $Max = 1.1$ and showed this scheme to be favourable over other values. Similarly to the Boltzmann selection, Rank allows for a weak selection technique which could result in a slower process in finding highly fit individuals. However this technique increases the preservation of diversity and could lead to a quicker convergence than fitness-proportionate selection.

#### 3.2.2.6 Tournament Selection

In the fitness-proportionate techniques described above, the EA has to pass twice through the population to calculate the average fitness and then to compute the expected value of each individual. Tournament selection chooses two individuals at random from the population, and takes a random value $r$ between 0 and 1. If $r > k$ ($k$ is a constant), then the fitter of the two individuals is chosen as a parent, otherwise the less fit individual is chosen. Both individuals are placed back into the population and can be chosen again. Although this technique is computationally more efficient, the selection process is still taking the fitness into consideration. If the constant $k$ is set to 0.5, this will result in half the time choosing the fitter individual and half the time the less fit one. However in a worst case scenario, the fittest individual is never chosen to be able to compete with another, and this might result in a loss of gene information.

### 3.2.3 Crossover and Mutation

Crossover is a genetic operation that takes two individuals (parents) from the population and mates them to create new individuals (offspring, children). The idea of this operation is to keep the genes present in a population, but at the same time try new combinations of these genes. The simplest form of crossover, shown in figure 3.1, is the single-point crossover, which takes a random point and switches the parents over to produce two new individuals. The disadvantages of this technique it when the individuals are long in length, since it limits the way different parts in the parents are recombined. To deal with this problem, it is possible to carry out a two-point crossover technique, where the parents' parts are switched at two random points rather than one. Still, the recombination can be restricted, and researchers experimented with various numbers of crossover points to try to maximise recombination possibilities. In implementations where the individuals are of varied length, the crossover point can be at a different location for the two parents, thus producing offspring of different length. Another technique is to perform bit-crossover, where the crossover occurs at every bit (as shown in figure 3.2). In this technique, the children will constitute of bits selected

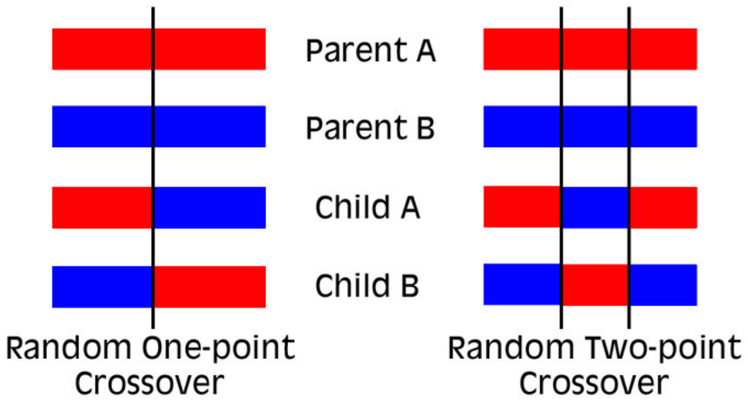

Figure 3.1: One- and two-point crossover of two individuals

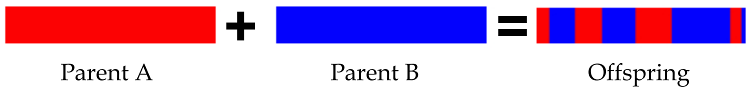

Figure 3.2: Bit crossover of two individuals

randomly from both parents, but these bits remain at the same position in the children as they were originally in the parents.

Mutation is the process through which an individual changes slightly in a random manner, and thus introducing a controlled randomness into the evolution of the population. This slight modification is designed to explore the search space better, considering slightly different solutions to our problem. However, due to the randomness of the mutation process, it introduces a certain disruptiveness within the population and an individual's fitness might improve as well as worsen. In general, the probability of mutation occurring in every generation is kept low, to control this injected disruptiveness which could have a negative impact on a well-performing population.

### 3.2.4 Convergence

A population is said to converge when all individuals in the population reach a similar fitness and no further improvement is observed. The EA should ideally permit the exploration of the search space by letting less fit individuals to reproduce and by finding a correct balance between crossover and mutation. A high mutation rate might reduce the fitness of the best individuals by randomly changing their genes. As we saw above, the selection method also affects the convergence of the population. Strict fitness-proportionate techniques which give more importance to the fitter individuals will lose the diversity of the population fitness and converge prematurely. Premature convergence does not give enough time for search space exploration, and we might reach a local maximum, which would be the best solution for the population we currently have, but might still not be the best solution over the whole search space.

## 3.3 Genetic Algorithms

A Genetic Algorithm (GA) (Goldberg, 1989; Holland, 1975) is based on EAs and it is designed to offer a generic algorithm which can be applied to several problems. An important element of GAs is the representation of the individual, since this should provide a flexible setup to solve a variety of problems.

### 3.3.1 Encoding the Individual

Deciding upon the way an individual is encoded is important as it will determine the way a GA and its operations are implemented. Typically, most GA implementations choose a fixed-length binary encoding since Holland's work concentrated on such a representation and thus is more well-documented. Other possible representations include many-character encodings (such as a protein structure, commonly found in work in the domain of bioinformatics) and real-value encodings (example to learn weights for a neural network).

The encoding is a representation of the solution to the problem we are attempting. Different encodings can represent different types of problems. If we are trying to learn simple and correct computer programs, we might try to represent the individuals as strings, made up of the syntax of possible programs. However we would then have to encode additional information, such as which parts of the individual can be changed by which in operations such as crossover and mutation. For example, we can substitute keywords with other keywords, but not with equations. Another representation could be a tree structure, which is a more natural representation as we can specify node types, and allow operations according to their type, without imposing restrictions on the encoding itself. Tree representation will be discussed in more detail in section 3.4, where Genetic Programming is described.

### 3.3.2 Genetic Operators

The genetic operations are affected by the encoding of the individual chosen and by implementation considerations. In a bit or character representation crossover and mutation can occur at any point in the gene. In real number representation, crossover could either take into consideration the length of each number and select the crossover point at such a particular interval, or it could select any point in the individual, thereby lower order values could become higher order, thus changing the value. Same applies for mutation where it could occur at a bit level (thus changing the value of the real number), or replacing a real number with a randomly generated one. Figure 3.1 shows how one-point and two-point crossover operations produce new individuals whilst figure 3.2 shows the result of the bit crossover technique.

## 3.4 Genetic Programming

Genetic Programming (GP) was introduced by Koza (1992), with the idea of applying evolutionary programming to automatically generate computer programs. The main distinguishing factor of GPs is the encoding of the individual where the encoding of an individual is represented as a tree. This change affects the way genetic operations are implemented in GPs.

### 3.4.1 Encoding the Individual

The design of a GP begins at a high order level, where one must define what problem the candidate solution (computer program) must solve. Once this is defined, it is possible to decide what set of terminals and primitive functions (non-terminals) will be used in the encoding of the individuals. Koza *et al.*

(2005) provides the following example to illustrate this process. The goal of the GP is to find a computer program whose output is equal to the values of the quadratic polynomial $x^2 + x + 1$ in the range between $-1$ and $+1$. This is referred to as symbolic regression where the aim is to discover the target functions and the necessary coefficients (or an approximation to them). In the given example, the non-terminal set of symbols (primitive functions) is set as $F = \{+, -, *, \%\}$ and the terminal set includes $x$ and a fixed set of constants in the range of $-5.0$ to $+5.0$. Figure 3.3 displays individuals which were randomly generated under the conditions described here.

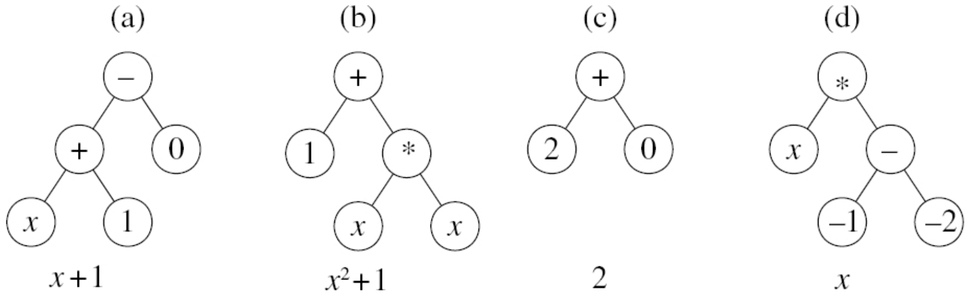

Figure 3.3: Randomly generated computer programs

In more complex problems, it is usually necessary to restrict the syntactic structure of the individual by specifying certain ‘conditions’. For instance, such a specification could restrict that one type of node or leaf must be followed only by another of a particular type. When such constraints are placed, all GP operations must comply with these specifications.

### 3.4.2 Genetic Operators with Trees

Crossover in GPs takes two selected individuals and recombines them at randomly selected nodes, as though sections of different programs are being switched. Figure 3.4 displays a possible crossover between two individual, where the crossover point for the first individual was selected at the + node, and at the $x$ node for the second individual. The resulting offspring from this crossover operation are also shown.

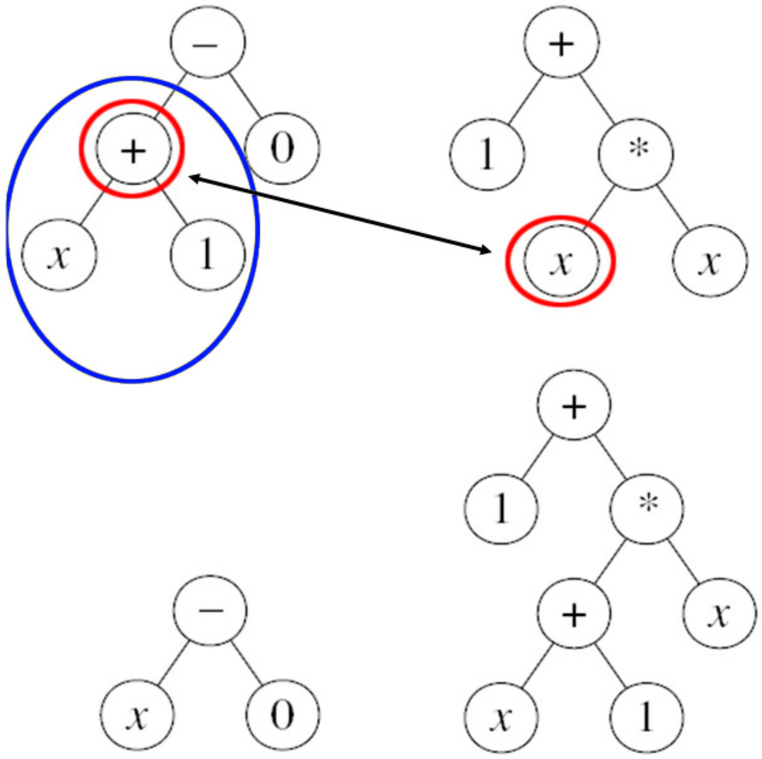

Figure 3.4: Crossover of two individuals

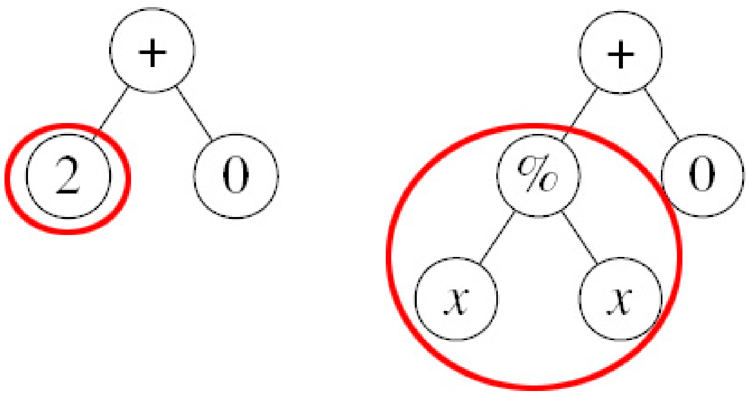

Figure 3.5: Mutation of an individual

In the case of mutation, a random node is chosen and the subtree from that node onwards is deleted. A new subtree is generated randomly and placed at that point, replacing the original subtree. This is shown in figure 3.5.

Similarly to GAs, the probability of mutation is kept low to around 1% of the population. Another genetic operator referred to by Koza is reproduction, whereby an individual is copied into the new population as is. In GAs, this process occurs when the Elite selection technique is implemented, maintaining the top scoring individuals within the next population to ensure that the best individuals are kept. Koza *et al.* suggest that around 8% of the population can be retained at each generation, chosen randomly through the process of reproduction and around 90% of the population are selected for crossover.

### 3.4.3 Fitness and Selection Techniques

The fitness of an individual in GPs is calculated by executing the program and comparing the result to the evaluation of the target program. Figure 3.3 displays the formulas that the trees represents, and according to the parameters of the problem specification, $x$ is evaluated in the range from $-1.0$ to $+1.0$. The value of the function can be compared to the value of the target function, thus giving an indication of how close a solution is. The perfect fitness value is that of 0 indicating that a solution has been found which gives the exact same result as the solution being sought.

The most common selection techniques in GPs are tournament selection and fitness-proportionate selection, both giving importance to the relatively fit individual. These selection methods are preferred as they are not greedy, meaning that weaker individuals still have a chance of being selected and stronger individuals are not guaranteed selection.

## 3.5 Conclusion

In this chapter we have given an overview of GAs and GPs and at the same time we touched upon certain algorithm choices which will influence our implementation described in chapter 5. This tutorial is by no means a complete overview, and readers who are interested in further detail should refer to more complete literature, especially (Holland, 1975; Koza *et al.*, 2005; Mitchell, 1998). Our purpose was to provide a sufficient explanation of the features found in GAs and GPs to understand the solution we propose for definition extraction in the rest of the dissertation.

# Chapter 4

# Language Technologies and Definition Extraction for eLearning

This thesis is a result of work carried out under LT4eL, a project where definition extraction from eLearning texts was attempted through manually crafted rules. The experience gained within the project, together with experiments carried out by different partners, gave us an insight to the difficulties encountered in the manual task of creating grammar rules based on observation. In this chapter we present the work we carried out within the project that served as a basis for this thesis.

## 4.1 Introduction

eLearning is the process of acquiring knowledge, information or skill through electronic means. One of the most popular gateways to eLearning is online via the Internet, often through Learning Management Systems (LMS). LMS allow tutors to manage collections of learning materials and monitor students' progress, whilst providing students with a structured way to access data. However, given the huge amount of static and dynamic learning content created for eLearning tasks, it becomes necessary to improve the effectiveness of retrieval and the accessibility of such documents through the LMS.

Language Technology can support eLearning, especially when used to enhance LMS. From a content perspective, it would be ideal if learning objects (LOs) would contain additional information to facilitate the retrieval of such documents. Content creators would want to emphasis their efforts on the learning task, rather than manually selecting and entering metadata. In the project Language Technologies for eLearning (LT4eL, Monachesi *et al.* (2007)) aims at enhancing eLearning through the use of keyword extraction, definition detection and concept annotation in LOs. Through these functionalities, embedded within a LMS, a tutor can provide additional information to a LO, such as a keyword list and definition compendiums present in the text, enabling the student to pin down the important topics present in a particular collection of documents. The implementation of these functionalities was based on the presence of implicit linguistic information available in texts. Nowadays, reliable linguistic tools such as part-of-speech taggers are available and provide linguistic analysis of text. Thus LOs can be annotated with linguistic information as meta-data (hidden from the user) which is available to this suite of tools.

The LT4eL definition extraction tool was based on grammar rules derived through manual observation of definitions and their linguistic properties. This proved to be a challenging job and the results achieved in the time-frame of the project were poor when compared to peer work. The project decided to split the definitions into distinct categories with the aim to improve results. This section covers the work carried out under the LT4eL project by the University of Malta in which we were heavily involved.

## 4.2 Corpus Preparation

Within the LT4eL project, we collected a corpus of English LOs of IPR-free (Intellectual Property Rights) documents in the areas of ICT and eLearning. The target for the corpus was of 200,000 words for each language, with the final English corpus consisting of over 1.2 million words. The majority of the LOs were written in a proprietary format, which does not allow easy manipulation and addition of metadata. In order to standardise the formats across the corpus, all LOs were converted into HTML (Hypertext Markup Language) to retain layout information (such as bold, italic, table format), and then automatically translated into an XML-based (extensible markup language) format conforming to the XCES (corpus encoding standard using XML) DTD, a specification for linguistically annotated corpora (Ide & Suderman, 2002). We created a linguistic annotator for English, which uses a part-of-speech tagger (Toutanova & Manning, 2000) and named-entity recogniser (Finkel *et al.*, 2005) as plug-in tools, to provide documents with the necessary additional linguistic metadata.

Once the corpus was linguistically annotated, we manually identified and annotated a set of 450 definitions within our corpus to assist in the creation and evaluation of the tools required for the project. Through this schema, all the information within our corpus becomes easily extractable and machine readable. Figure 4.2 show an annotated sentence which includes markings for a definition, keyword, named entity and other linguistic information for each word in the final XML format.

## 4.3 Rule-Based Definition Extraction

Initially, we set out to have 450 manually annotated definitions, split into three sets: (i) a training set, (ii) a testing set, and finally (iii) an evaluation set, each consisting of 150 definitions. The training set was used to manually observe possible patterns that can be commonly found in definitions; the testing set was to test the created rules and on the basis of the results tweak the rules; and the evaluation set was used to carry out a final evaluation of the grammar rules.

The rules were created through manual observation of these sentence definitions, representing mainly the POS sequences noticed. An XML transducer, *lxtransduce* (Tobin, 2005), was used to match the defined patterns and a rewrite rule is then applied to the matched cases. In our case, definitions are left intact, and surrounded with `definingText` tags. Figure 4.3 shows an example of a grammar rule which looks for a determiner at the beginning of a sentence followed by a noun.

A program to evaluate new grammar rules was created to immediately evaluate any changes to the rules. Thus is was possible to monitor the effect of the slightest change. Often, the result was that when using more specific rules, we

```
<s id="s81">
<definingText id="dt1" def_type1="is_def" def="m1">
<markedTerm id="m1" dt="y" kw="y">
<chunk id="c1a" type="NAME">
<tok id="t2172" class="word" sp="y" ctag="NNP" base="latex"
                                  msd="N,SG,proper,vrbl">LATEX</tok>
</chunk>
</markedTerm>
<tok id="t2173" class="punc" sp="y" ctag="(" base="("msd="">(</tok>
<tok id="t2174" class="word" sp="y" ctag="VBN"
           base="pronounce" msd="V,PAST,ED,finite">pronounced</tok>
<tok id="t2175" class="other" sp="y" ctag="JJ"
                                 base="lah-tek" msd="">Lah-tek</tok>
<tok id="t2176" class="word" sp="y" ctag="CC" base="or"
                                                  msd="CJ">or</tok>
<tok id="t2177" class="other" sp="y" ctag="JJ"
                                 base="lay-tek" msd="">Lay-tek</tok>
<tok id="t2178" class="punc" sp="y" ctag=")" base=")"msd="">)</tok>
<tok id="t2179" class="word" sp="y" ctag="VBZ" base="be"
                                    msd="AUX,PRES,S,finite">is</tok>
<tok id="t2180" class="word" sp="y" ctag="DT" base="a"
                                             msd="DT,SG,wh">a</tok>
<tok id="t2181" class="word" sp="y" ctag="NN"
                      base="collection" msd="N,SG">collection</tok>
<tok id="t2182" class="word" sp="y" ctag="IN" base="of"
                                                  msd="PP">of</tok>
<tok id="t2183" class="word" sp="y" rend="i" ctag="FW"
                   base="macro" msd="N,PL,proper,vrbl">macros</tok>
...
</definingText>
</s>
```

Figure 4.1: A sample of an annotated sentence

obtained a higher precision but lower recall. On the other hand, if we attempted

```
<rule name="det_S_noun_phrase">
 <seq>
  <query match="s/*[1][name()='tok'][@ctag='DT']"/>
  <ref name="noun_group" mult="+"/>
 </seq>
</rule>
```

Figure 4.2: A sample of a manually crafted rule

to generalise the rules slightly, we would gain a higher recall, but precision would lower. The task of actually discovering these grammar rules manually was tedious and required linguistic knowledge.

Another issue observed during this phase was how to deal with definitions that are spread over multiple sentences. Human annotators were instructed to annotate multi-sentence definitions as constituting of parts. However, the rules in lxtransduce were not able to cover multi-sentences and thus affected our results. It was decided that if a rule would match one sentence of a multi-sentence definition, it would be considered as a match, even if the whole definition is not captured. In an eLearning context this is acceptable since the tutor is presented with the context of the extracted definitions (surrounding sentences) and it would still be possible for the tutor to include a second sentence as part of the definition.

## 4.3.1 Categorising Definitions

On the basis of the initial results and difficulties encountered, it was decided to adopt a divide-and-conquer approach by splitting definitions into different categories observed with the hope that working on the categories separately would improve the quality of our grammars. This also allowed to concentrate on grammar rules within a particular category which allowed to focus on specific characteristics in that category. The types of definitions observed in our texts have been classified as follows:

**Is-a** Definitions containing the verb "to be", generally followed by a determiner.
E.g.: "A joystick is a small lever (as in a car transmission gearshift) used

mostly in computer games."

**Verb** Definitions containing other verbs as connectors such as "means", "is defined", "is called". E.g.: "the ability to copy any text fragment and to move it as a solid object anywhere within a text, or to another text, usually referred to as cut-and-paste." In this case the term being defined is at the end of the sentence, and it is classified so by the use of 'refer to'.

**Punctuation** Definitions containing punctuation features, usually separating the term being defined and the definition itself. This category is also referred to as 'punct' for short. E.g.: "hardware (the term applied to computers and all the connecting devices like scanners, modems, telephones, and satellites that are tools for information processing and communicating across the globe)." where the definition is contained within brackets.

**Layout** Definitions containing particular layout style, similar to the punctuation feature, but separated through the use of a table or the defining term is a heading and the definition is the sentence below it (similar to the punctuation definition, however the term and definition would be placed in separate cells).

**Anaphora** Definitions containing a pronoun, usually referring to the defining term which would be placed outside the definitory context. This is common in cases where the definition is over more than one sentence, and the second sentence would refer to the defining term using a pronoun. E.g.: "This (Technology emulation) involves developing techniques for imitating obsolete systems on future generations of computers ."

**Other** Other definitions to capture those which do not fall in the above categories. E.g.: "information skills, i.e. their ability to collect and process the appropriate information properly in order to reach a preset goal." where the defining term and the definition are separated by 'i.e.'.

Table 4.1 shows the distribution of the manually annotated sentences according to the described categories above. Work carried out on the manually crafted rules for the LT4eL project focused on the first three categories where is was

Table 4.1: Distribution of definitions by category

| **Category** | **Number** | **Percentage** |
|---|---|---|
| Is-a | 111 | 24% |
| Verb | 141 | 30% |
| Punctuation | 127 | 27% |
| Pronoun | 28 | 6% |
| Layout | 8 | 2% |
| Other | 49 | 10% |

viable to concentrate efforts in, since the majority of definitions fell within these categories. The forth category, based on layout style, depends on the conversion process to HTML, and how well the information is retained. For instance, in the conversion carried out in LT4eL, table information was not retained. The fifth category would have required anaphora resolution to be applied, where words such as 'this' would have been resolved to what original word it refers to. For example "There is a TEX system called LATEX. This is a typesetting system.". The second sentence would have been marked as definitional sentence under the fifth category. Had we applied anaphora resolution, the word "This" would have been replaced by "LATEX". The last category was introduced as a catch-all one, where all remaining sentences were grouped together.

### 4.3.2 Results of Manually Crafted Rules

Although categorisation of definitions permitted focusing efforts to crafting rules over a smaller set of sentences, it was still hard to improve results. Table 4.2 presents the results achieved with the manually crafted rules delivered as part of the LT4eL project.

In general, the results are not very promising, especially if these had to be compared to other related work. These results were similar throughout the other languages within the LT4eL project, which motivated us to look into machine learning techniques for improvement.

Table 4.2: Results for definition extraction with manually crafted rules

| **Category** | **F-measure** | **Precision** | **Recall** |
|---|---|---|---|
| Is-a | 0.26 | 0.17 | 0.58 |
| Verb | 0.33 | 0.34 | 0.32 |
| Punctuation | 0.17 | 0.33 | 0.12 |

To discriminate between definitions and non-definitions in an automatic manner, it is important to identify features which are present in the definitional class but not in the non-definitional class. Such features may range from a simple rule stating *contains the verb to be*, to more complex rules, such as POS sequences. The proposed machine learning approach aimed at extending and improving the results achieved from the manually-crafted grammars.

## 4.4 Machine Learning Tasks

Various machine learning approaches were tried in order to improve the results of the definition extraction process. For the Dutch language, Westerhout & Monachesi (2007a) attempted several experiments based on work carried out by Fahmi & Bouma (2006), using different textual properties as features and then using three different supervised learning methods, namely namely naïve Bayes, maximum entropy and support vector machine. They managed to increase f-measure in the is-a category from 42% with manually crafted rules to 79% using a naïve Bayes classifier (with an accuracy of 88%). In an elearning context such a result is acceptable considering that a tutor would then verify the proposed sentences to be included in a final glossary.

For Polish, Degórski *et al.* (2008) experiment with well-known classifying algorithms (naïve Bayes, decision trees (ID3 and C4.5), lazy classifier IB1, AdaBoostM1 with Decision Stump and AdaBoostM1 with nu-SVC). From an f-measure of 28% with manually crafted rules, the best increase is achieved by the ID3 classifier with an f-measure of 32%. In a separate experiment, Kobyliński & Przepiórkowski (2008) use Balanced Random Forest (BRF) which is a machine learning technique for classification using decision trees, where decisions are based on a subset of attributes which are randomly selected and then the best attribute for the current tree is chosen. Here the f-measure also reaches 32%, with accuracy at 85%.

The various experiments carried out supported the idea that in the case of the corpora being used within LT4eL, machine learning techniques improved results, and in some cases, substantially. Thus, from our part we proposed to use evolutionary algorithms, which as yet had not been applied to the task of definition extraction and classification. A detailed proposal is described in the following chapter.

## 4.5 Conclusion

Through the categorisation of definitions, we were able to improve results for certain categories, such as the is-a category. However, having achieved a high recall, precision was considerably low. This meant that whilst good definitions were being captured, a high number of incorrect definitions were also being included in the result set.

Another problem was that there was no ranking of the results as the extraction method used was a simple yes/no classification. Definitions were presented to the user in the order in which they appeared in the texts. Since the system was intended to suggest definitions to a tutor for approval, having a few incorrect definitions was not deemed as a problem. However, it is desirable that the definitions are presented in a ranked order, so that those definitions with a higher confidence value are presented at the top of the results. We also observed that incorrectly classified definitions could be filtered out using post-processing filtering after the initial grammar was applied. We propose to use GAs and GPs to include these desired features into the process of definition extraction. Our focus remains on English non-technical eLearning texts.

# Chapter 5

# Experiment Design and Methodology

In this chapter we discuss the setup of the two proposed experiments for definition extraction from natural language texts — one using a GA and another using a GP. The GA aims learning weights to a set of features; thus we define what we mean by a feature, what an individual constitutes of, and how it will be represented in the GA. We also discuss different fitness functions and selection methods available, and justify why the purpose of the initial experiment is to find the ideal setup for the GA. The GP, although similar to a GA in structure, requires different individual representation, and thus we define the language of the individual's representation and how the individual will be interpreted. The description and discussion on both the GA and the GP serve as the basis of the implementation of the two experiments.

## 5.1 Experiment Overview

We are confronted with two specific problems in definition extraction. The first problem is that manually crafted rules have no information as to how effective they are as definitions classifiers. To solve this problem we propose our first experiment, whereby we use a GA which learns weights to a fixed set of features or rules, and these weights can then be used to represent a level of importance or confidence associated to respective features, allowing a definition extractor to rank classified definitions according to some form of score.

The second problem identified is that manually crafting grammar rules is not an easy task, requires certain expertise and is time-consuming. Our proposal attempts solving this problem by applying a GP to a training corpus of definitional and non-definitional sentences. The GP generates different grammar rules (by observing a certain process) and tests whether these rules are good definition classifiers. Thus, it is possible to learn rules which capture definitions in an automatic manner.

In this section we describe in detail the two proposed experiments and the choices made for the implementation of these two algorithms. Of particular importance in the following sections is the discussion of how we represent the individuals in both experiments, and the different fitness functions used.

### 5.1.1 Experiment One: Genetic Algorithm

From the experience gained through the manual crafting of grammar rules for definition extraction, we noticed that certain rules, or sub-parts, contain more specific or important information than other rules. This led to the idea of using a GA as a possible technique to learn the importance of the features that can recognise definitions. This can be done by assigning weights to each feature and allowing the algorithm to adjust the weights according to the performance.

#### 5.1.1.1 Feature Description

A feature is a test which, given a sentence $s$, returns whether a particular structure, word or linguistic object is present in the sentence — essentially characteristics that may be present in sentences. These could range from rendering information (bold, italic), to the presence of keywords, or part-of-speech sequences that could identify the linguistic structure of a definition. Consider the following two example sentences:

$s_1$ = "**vi** is a text editor that runs under Unix."

$s_2$ = "The Artificial Intelligence department *is part of* the ICT Faculty."

Consider also the following set of features:

$$\begin{aligned} f_1 &= \text{FW}\rightarrow\text{VBZ}\rightarrow\text{DT}\rightarrow\text{NN}\rightarrow\text{NN}\rightarrow\text{WDT}\rightarrow\text{VBZ}\rightarrow\text{IN}\rightarrow\text{NNP} \\ f_2 &= \text{hasItalic} \\ f_3 &= \text{hasBold} \end{aligned}$$

Feature $f_1$ represents a part-of-speech sequence showing the order of the tags expected in a particular sentence. Features $f_2$ and $f_3$ describe rendering information a sentence should contain. In our examples, sentence $s_1$ matches features $f_1$ and $f_3$, while sentence $s_2$ matches feature $f_2$. A match returns a value of 1, whereas a non-match returns a value of 0. Note that a feature does not necessarily have to match a whole sentences, but can match only a part. It is also possible that the feature is present more than once in the sentence (say, there is more than one bold word), however the return is still 1 in that a match has occurred.

Now, given a set of $n$ basic features, $\overline{f} = \langle f_1 \ldots f_n \rangle$, and $n$ numeric constants, $\overline{\alpha} = \langle \alpha_1 \ldots \alpha_n \rangle$, one can produce a compound feature combining these basic features in a linear fashion:

$$F_{\overline{\alpha}}^{\overline{f}}(s) = \sum_{j=1}^{n} \alpha_j \times f_j(s)$$

If we apply this to our example sentences above, taking, for instance, the numeric constants to be $\overline{\alpha} = \langle 4, -1, 3 \rangle$.

$$\begin{aligned} F_{\langle 4,-1,3\rangle}^{\langle f_1,f_2,f_3\rangle}(s_1) &= 4 \times 1 + (-1) \times 0 + 3 \times 1 = 7 \\ F_{\langle 4,-1,3\rangle}^{\langle f_1,f_2,f_3\rangle}(s_2) &= 4 \times 0 + (-1) \times 1 + 3 \times 0 = -1 \end{aligned}$$

There are different ways how these values may be interpreted.

- Classifying a sentence as a definition if and only if its score is greater than 0. In this case $s_1$ would be classified as a definitions, whereas $s_2$ would not.

- Using zero as the cut-off point is arbitrary and can be set to any particular value $\tau$. For example if we take $\tau$ to be $-5$, both $s_1$ and $s_2$ would be classified as definitions.
- A more elaborate interpretation is to use the value as the confidence by which one can categorise the sentences as a definition. In this case, $s_1$ is more likely to be a definition than $s_2$ since it has a higher score. Naturally this can be done both using zero or another arbitrary cut-off point.

Clearly the question now lies in how appropriate values for the vector of numeric constants $\overline{\alpha}$, and $\tau$ can be chosen.

#### 5.1.1.2 Learning Weights

The problem now is: given a fixed set of features $\overline{f}$, how can we calculate a good set of weights $\overline{\alpha}$, so as to maximise the effectiveness of the combined features as a definition classifier?

We use a GA with the different possible interpretations of a compound feature described above. The values learnt would thus correspond to the relative effectiveness of the individual features as classifiers of definitions. Before starting the experiment, a predefined set of features is adopted and remain static throughout the experiment. The individual will be encoded as a list of real numbers of length equal to the number of predefined features. Thus, the $i$th individual ($1 \leq i \leq \textit{populationsize}$) will have the structure:

$$g_i = \langle \alpha_{i,1}, \alpha_{i,2} \ldots \alpha_{i,n} \rangle$$

Note that $n$ corresponds to the number of predefined features. An individual $g_i$ scores a sentence $s$ using the compound feature formula given earlier:

$$\textit{value}_i(s) = F^{\overline{f}}_{\langle \alpha_{i,1}, \ldots, \alpha_{i,n} \rangle}(s) = \sum_{j=1}^{n} f_j(s) \times \alpha_{i,j}$$

The initial population will consist of genes with random weights assigned to each feature.

Table 5.1: Set of definitional sentences

| **Definition** | **Value** |
|---|---|
| $D_1$ | 3 |
| $D_2$ | 2 |
| $D_3$ | 2 |
| $D_4$ | 1 |
| $D_5$ | $-2$ |

Table 5.2: Set of non-definitional sentences

| **Non-definition** | **Value** |
|---|---|
| $ND_1$ | 1 |
| $ND_2$ | 0 |
| $ND_3$ | $-1$ |
| $ND_4$ | $-3$ |
| $ND_5$ | $-5$ |

#### 5.1.1.3 Fitness Function

Given that we have a training corpus available, we can define a fitness function that is based on how many definitions and non-definitions an individual manages to classify correctly. F-measure, precision and recall are popular metrics used in retrieval and classification domains. These metrics can be used as a fitness function to gauge the performance of an individual in the population (providing a way to rank individuals and produce a quantifiable fitness measure).

Given an individual $i$, and a value $\tau$ separating definitions from non-definitions, one can define the following values:

$$\begin{aligned}
truePositives_\tau(i) &= count\{s \mid s \in D \wedge value_i(s) \geq \tau\} \\
falsePositives_\tau(i) &= count\{s \mid s \in ND \wedge value_i(s) \geq \tau\} \\
trueNegatives_\tau(i) &= count\{s \mid s \in ND \wedge value_i(s) < \tau\} \\
falseNegatives_\tau(i) &= count\{s \mid s \in D \wedge value_i(s) < \tau\}
\end{aligned}$$

**Example:** To illustrate better, we will use the following example. Let $D$ be the set of definitions in the corpus with their associated $value_i(s)$ shown in table 5.1, and $ND$ be the set of the non-definitions with their associated $value_i(s)$ shown in table 5.2. For instance, taking zero as the value of $\tau$, we obtain the following values:

$$truePositives_0(i) = count\{D_1, D_2, D_3, D_4\} = 4$$

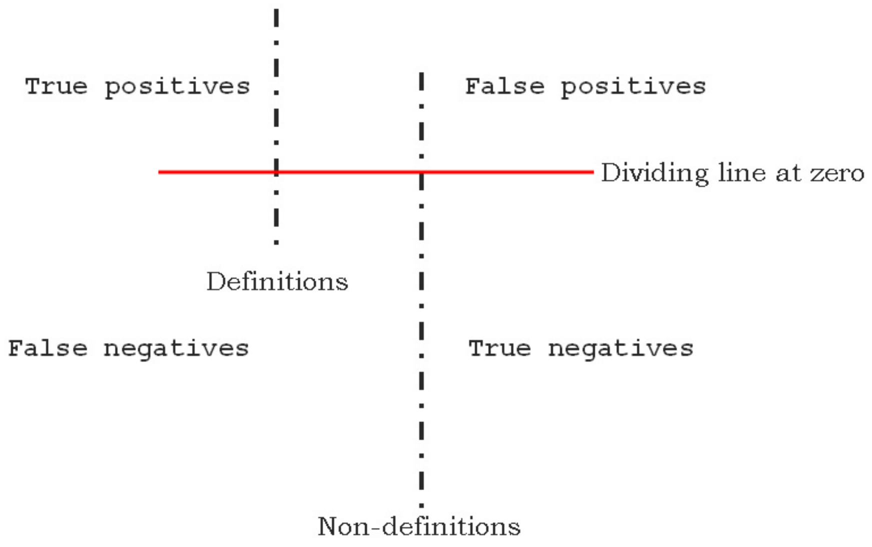

Figure 5.1: Classification of individuals

$$\begin{aligned} \mathit{falsePositives}_0(i) &= \mathit{count}\{ND_1, ND_2\} = 2 \\ \mathit{trueNegatives}_0(i) &= \mathit{count}\{ND_3, ND_4, ND_5\} = 3 \\ \mathit{falseNegatives}_0(i) &= \mathit{count}\{D_5\} = 1 \end{aligned}$$

If instead we take the value of $-2$ for $\tau$, we obtain the following results:

$$\begin{aligned} \mathit{truePositives}_{-2}(i) &= \mathit{count}\{D_1, D_2, D_3, D_4, D_5\} = 5 \\ \mathit{falsePositives}_{-2}(i) &= \mathit{count}\{ND_1, ND_2, ND_3\} = 3 \\ \mathit{trueNegatives}_{-2}(i) &= \mathit{count}\{ND_4, ND_5\} = 2 \\ \mathit{falseNegatives}_{-2}(i) &= \mathit{count}\{\} = 0 \end{aligned}$$

Figure 5.1 shows this in a graphic way, where sentences are plotted in each of the four quadrants representing the value described. $\diamond$

There are various ways of how these values could be used, however our focus will be on precision, recall and f-measure, metrics which are widely used in information retrieval and classification problems. Based on these counts, we calculate the following metrics:

*Precision* is the percentage of correctly classified definitions from all sentences being proposed as definitions by the learning system. This percentage measures the quality of the definitions being proposed.

$$Precision_\tau(i) = \frac{truePositives_\tau(i)}{truePositives_\tau(i) + falsePositives_\tau(i)}$$

*Recall* is the percentage of correctly classified definitions from all the set of positively marked sentences in the training data. This percentage measures how much of the actual definitions we managed to capture.

$$Recall_\tau(i) = \frac{truePositives_\tau(i)}{truePositives_\tau(i) + falseNegatives_\tau(i)}$$

**Example:** Building on the previous example we can now calculate precision and recall at $\tau = 0$:

$$Precision_0(i) = \frac{truePositives_0(i)}{truePositives_0(i) + falsePositives_0(i)} = \frac{4}{4+2} = 0.67$$

$$Recall_0(i) = \frac{truePositives_0(i)}{truePositives_0(i) + falseNegatives_0(i)} = \frac{4}{4+1} = 0.80$$

Similarly, the values of precision and recall for when $\tau = -2$ are:

$$Precision_{-2}(i) = \frac{truePositives_{-2}(i)}{truePositives_{-2}(i) + falsePositives_{-2}(i)} = \frac{5}{5+3} = 0.63$$

$$Recall_{-2}(i) = \frac{truePositives_{-2}(i)}{truePositives_{-2}(i) + falseNegatives_{-2}(i)} = \frac{5}{5+0} = 1.00$$

Note that in this example, by taking the classifying line as $\tau = -2$, we capture all the definitions in our corpus, indicated by recall = 1.00. However, more non-definitions are captured as definitions, affecting precision in a negative way. Still, the increase in recall has surpassed the decrease in precision. ◇

*F-measure* is a metric that uses precision and recall together with a *kappa*[1] value, which gives the relative importance to be assigned to precision or recall.

[1]In Information Retrieval, this value is usually referred to as *alpha*. However, in order to avoid confusion between the alpha's being learnt by the GA as features' weights, we refer to the f-measure's alpha value as kappa or $\kappa$.

$$F_{\tau}^{\kappa}(i) = \frac{(1+\kappa^2)\cdot(\mathit{Precision}_{\tau}(i)\cdot \mathit{Recall}_{\tau}(i))}{(\kappa^2\cdot \mathit{Precision}_{\tau}(i) + \mathit{Recall}_{\tau}(i))}$$

The kappa value is used to give more importance to either one of the two metrics. Our approach in this experiment is to use f-measure as a fitness function, with kappa being equal to one (no preference to either precision or recall).

**Example:** Developing our example further, we can calculate f-measure with $\kappa = 1.0$ and $\tau = 0$ as follows:

$$F_{0}^{1.0}(i) = \frac{(1+1.0^2)\cdot(\mathit{Precision}_{0}(i)\cdot \mathit{Recall}_{0}(i))}{(1.0^2\cdot \mathit{Precision}_{0}(i) + \mathit{Recall}_{0}(i))} = \frac{2\cdot 0.67\cdot 0.8}{1.0\cdot 0.67 + 0.80} = 0.73$$

Calculating f-measure with $\tau = -2$ we would have the following results:

$$F_{-2}^{1.0}(i) = \frac{(1+1.0^2)\cdot(\mathit{Precision}_{-2}(i)\cdot \mathit{Recall}_{-2}(i))}{(1.0^2\cdot \mathit{Precision}_{-2}(i) + \mathit{Recall}_{-2}(i))} = \frac{2\cdot 0.63\cdot 1.00}{1.0\cdot 0.63 + 1.00} = 0.77$$

From this example, we consider $\tau = -2$ to be a better definition classifier since it increased the overall f-measure from 0.73 to 0.77. $\diamond$

The f-measure metric presents an interesting problem found in all types of classification problems. Although we would like to achieve both a high precision and a high recall, this becomes difficult as either one improves considerably. As seen in the above example, a high recall (capturing all the definitions) usually results in capturing also a number of non-definitions, which affects precision negatively. On the other hand, a high precision reflects the correctness of those definitions being captured, which usually means that our rules are too specific and are not capturing the full set of definitions, resulting in low recall. In our experiments we can use the kappa value to emphasis which of the two values should be considered more important, precision or recall. This will allow the experiments to shift the learning preference towards either a higher recall or a higher precision. Of course, when carrying out such experiments we are aware that if one value gains, the other suffers. Yet it is interesting to note the results of what can be learnt under such conditions.

Thus, we propose the following two experiments for different value of $\kappa$:

1. Taking $\tau$ to be zero, thus using $F_0^{\kappa}(i)$ as the fitness score of individual $i$. We refer to this method as `CountZero`.

2. Choosing an optimal value of $\tau$ for individual $i$, thus giving the following score to individual $i$:
   $$\max\{F_\tau^{\kappa}(i) \mid \tau \in \mathbb{R}\}$$
   We refer to this method as `CountShifted`. In practice we calculate this not over all real numbers but only over the scores of the sentences which would give the same results.

An alternative way of using the formulae is by taking the sum of the squares of the distances above or below the threshold $\tau$ (or zero), rather than simply the count. Using the following alternative definitions:

$$\begin{aligned}
\Delta truePositives_\tau(i) &= \sum\{(value_i(s)-\tau)^2 \mid s \in D \wedge value_i(s) \geq \tau\}\\
\Delta falsePositives_\tau(i) &= \sum\{(value_i(s)-\tau)^2 \mid s \in ND \wedge value_i(s) \geq \tau\}\\
\Delta trueNegatives_\tau(i) &= \sum\{(value_i(s)-\tau)^2 \mid s \in ND \wedge value_i(s) < \tau\}\\
\Delta falseNegatives_\tau(i) &= \sum\{(value_i(s)-\tau)^2 \mid s \in D \wedge value_i(s) < \tau\}
\end{aligned}$$

**Example:** To understand the value of the formulae being proposed, we calculate these figures where at $\tau = 0$ we will have the following results:

$$\begin{aligned}
\Delta truePositives_0(i) &= \sum\{(3-0)^2,(2-0)^2,(2-0)^2,(1-0)^2\} = 18\\
\Delta falsePositives_0(i) &= \sum\{(1-0)^2,(0-0)^2\} = 1\\
\Delta trueNegatives_0(i) &= \sum\{(-1-0)^2,(-3-0)^2,(-5-0)^2\} = 35\\
\Delta falseNegatives_0(i) &= \sum\{(-2-0)^2\} = 4
\end{aligned}$$

And at $\tau = -2$ we will have the following:

$$\begin{aligned}
\Delta truePositives_{-2}(i) &= \sum\{(3-(-2))^2,(2-(-2))^2,(2-(-2))^2,\\
&\qquad (1-(-2))^2,(-2-(-2))^2\} = 66\\
\Delta falsePositives_{-2}(i) &= \sum\{(1-(-2)^2,(0-(-2))^2,(-1-(-2))^2\} = 14\\
\Delta trueNegatives_{-2}(i) &= \sum\{(-3-(-2))^2,(-5-(-2))^2\} = 10\\
\Delta falseNegatives_{-2}(i) &= \sum\{\} = 0
\end{aligned}$$

$\diamond$

Based on these, one can calculate precision ($\Delta Precision_\tau(i)$), recall ($\Delta Recall_\tau(i)$) and f-measure ($\Delta F^\kappa_\tau(i)$) just as before but using these values instead of the counts.

**Example:** Building further on our calculations, we can now obtain values for precision and recall at $\tau = 0$:

$$\Delta Precision_0(i) = \frac{\Delta truePositives_0(i)}{\Delta truePositives_0(i) + \Delta falsePositives_0(i)} = \frac{18}{18+1} = 0.95$$

$$\Delta Recall_0(i) = \frac{\Delta truePositives_0(i)}{\Delta truePositives_0(i) + \Delta falseNegatives_0(i)} = \frac{18}{18+4} = 0.82$$

Similarly, the values for precision and recall for when $\tau = -2$ are:

$$\Delta Precision_{-2}(i) = \frac{\Delta truePositives_{-2}(i)}{\Delta truePositives_{-2}(i) + \Delta falsePositives_{-2}(i)} = \frac{66}{66+14} = 0.83$$

$$\Delta Recall_{-2}(i) = \frac{\Delta truePositives_{-2}(i)}{\Delta truePositives_{-2}(i) + \Delta falseNegatives_{-2}(i)} = \frac{66}{66+0} = 1.00$$

We note that although these values are different from the count values calculated previously, the affect of the classification is the same. Taking $\tau = -2$ has increase recall to 1.00, but lowered precision to 0.83.

We now calculate f-measure, $\Delta F^\kappa_\tau(i)$, with $\kappa = 1.0$, at $\tau = 0$:

$$\Delta F^{1.0}_0(i) = \frac{(1+1.0^2)\cdot(\Delta Precision_0(i)\cdot \Delta Recall_0(i))}{(1.0^2\cdot \Delta Precision_0(i) + \Delta Recall_0(i))} = \frac{2\cdot 0.95\cdot 0.82}{1.0\cdot 0.95 + 0.82} = 0.88$$

Finally, we calculate f-measure at $\tau = -2$:

$$\Delta F^{1.0}_{-2}(i) = \frac{(1+1.0^2)\cdot(\Delta Precision_{-2}(i)\cdot \Delta Recall_{-2}(i))}{(1.0^2\cdot \Delta Precision_{-2}(i) + \Delta Recall_{-2}(i))} = \frac{2\cdot 0.83\cdot 1.00}{1.0\cdot 0.83 + 1.00} = 0.91$$

Again, we note that the affect of using the distance rather than the count effected our figures, but not the way the sentences are classified. ◇

From the above examples, we see that using the distance rather than the count, the scores increase considerably. In our experiments we want to investigate the use of this metric to see whether the learning of weights would result to giving a better separating line between definitions and non-definitions. We yield two experimental setups for a particular value of $\kappa$:

1. Taking $\tau$ to be zero, thus using $\Delta F_0^\kappa(i)$ as the fitness score of individual $i$. We refer to this method as `DistanceZero`.

2. Choosing an optimal value of $\tau$ for individual $i$, thus giving the following score to individual $i$:
$$\max\{\Delta F_\tau^\kappa(i) \mid \tau \in \mathbb{R}\}$$
We refer to this method as `DistanceShifted`. Again, in practice we calculate this not over all real numbers but only over the scores of the sentences which gives an adequate approximation.

#### 5.1.1.4 Other Configuration Aspects

There are different configuration aspects described in Chapter 3 with respect to selection methods, crossover techniques and settings such as population size and mutation rate. A GA is a machine that is coded to solve a particular optimisation problem. In our case, we seek to optimise weights for a set of features that could identify definitions. Since the ideal setting is not known in advance, part of the experiment will be to discover which of these configuration aspects are most effective for our problem.

### 5.1.2 Experiment Two: Genetic Programming

Using the GA as described above one can make do without human expertise to give relative importance to different features. However, the features would still have to be identified by human experts. In this second experiment we propose a GP approach to automatically learn new features to be used by the GA.

```
feature ::=
    simplefeature
  | simplefeature & feature

simplefeature ::=
    lobj
  | emptystring
  | any
  | simplefeature ?
  | simplefeature *
  | simplefeature . simplefeature
  | simplefeature + simplefeature
```

Figure 5.2: BNF specification for the representation of individuals

#### 5.1.2.1 Representation of the Individual

Individuals in a GP are represented as a tree, i.e. members of a language described as a context-free grammar (Koza, 1992). We thus have to define a language which can be used to describe features. Through observations of related work and the capabilities of *lxtransduce* (Tobin, 2005), regular expressions (extended with a number of constructs) should be sufficient to produce expressions that could correctly identify definitions.

We want to learn a combination of features that can be used to strengthen the confidence of definition classification. For this reason we specify two types of features which would be present in our language:

1. Feature — A unit which contains a conjunction of simple features
2. Simple feature — A unit which matches a simple object, such as a part-of-speech unit 'VB' (verb)

The grammar over which the individuals in the GP will range is given in figure 5.2, which essentially corresponds to regular expressions over simple linguistic objects, with the possibility of conjunction at the top level. This restriction over conjunction is used so as to ensure that the definition extractors are of reasonable size — allowing conjunction at any point in a regular expression would lead to a blow-up in the size of the transducers produced from such individuals.

Individuals of the GP will be interpreted as regular expressions by using the following semantics. Given a way of translating simple linguistic objects into regular expressions $re_l$ we can define a function $re$ translating features into regular expressions:

$$\begin{aligned}
re(sf) &\overset{df}{=} re_s(sf) \\
re(sf \texttt{ \& } f) &\overset{df}{=} (any^* \cdot re_s(sf) \cdot any^*) \wedge re(f)
\end{aligned}$$

$$\begin{aligned}
re_s(\texttt{emptystring}) &\overset{df}{=} \varepsilon \\
re_s(\texttt{any}) &\overset{df}{=} any \\
re_s(sf\texttt{?}) &\overset{df}{=} \varepsilon \mid re_s(sf) \\
re_s(sf\texttt{*}) &\overset{df}{=} re_s(sf)^* \\
re_s(sf_1\,\texttt{.}\,sf_2) &\overset{df}{=} re_s(sf_1) \cdot re_s(sf_2) \\
re_s(sf_1\texttt{+}sf_2) &\overset{df}{=} re_s(sf_1) \mid re_s(sf_2) \\
re_s(\texttt{lobj}) &\overset{df}{=} re_l(lobj)
\end{aligned}$$

We note that in the case of `lobj`, this is translated according to the actual linguistic feature being matched. For instance, when we look for the word *is*, the regular expression specifies that we are searching a part in the sentence where the part-of-speech is a verb and the base of the word is *be*. For each of the linguistic objects, we define a specific regular expression that represents the rules that have to be satisfied in order for a match to be made.

#### 5.1.2.2 Fitness of the Individual

In GPs the fitness function is the execution of the program where the result is then compared to the target program. In our case, we are dealing with a grammar representation that will be used to capture definitional sentences. Thus the fitness of the individual is the execution of the rule on our training corpus, and the result will be the classification of the sentences. Similarly to the GA, we will focus on collecting figures for:

$$\begin{aligned}
truePositives(i) &= count\{s \mid s \in D \wedge matches(s, re(i))\} \\
falsePositives(i) &= count\{s \mid s \in ND \wedge matches(s, re(i))\} \\
trueNegatives(i) &= count\{s \mid s \in ND \wedge \neg matches(s, re(i))\} \\
falseNegatives(i) &= count\{s \mid s \in D \wedge \neg matches(s, re(i))\}
\end{aligned}$$

where *matches* takes a sentence and a regular expression and returns whether the former matches the latter. Using these definitions one can define precision, recall and f-measure as defined in section 5.1.1.3. We propose to run this experiment using the f-measure as the fitness function, keeping Kappa at 1.0 since we want to learn rules without emphasising precision or recall.

## 5.2 Conclusions

In this chapter we described the experimental setup for different instances of GAs and GPs to learn how to identify definitional sentences. In the case of the GA the objective is to experiment with different selection methods, fitness functions and other configuration aspects which will influence the performance of the algorithm. This experiment will thus produce the setup of a final machine to be used with any set of features. The GP addresses a different question which is that of learning relevant features which can be used in the GA. The results of these experiments are reported in the following two chapters.

# Chapter 6

# Genetic Algorithms for Definition Extraction

In this chapter we present the setup of the GA experiment together with the results achieved over the different runs. The experiment focuses on trying out different fitness functions and selection methods to find an ideal GA configuration for definition extraction. We also experiment with different kappa values in the f-measure formula, showing that it is possible to influence the GA's learning process to favour either precision or recall. Finally, we analyse in a qualitative manner as to how the best individual classifies sentences. The evaluation brings forward errors in the human annotation of definitions in our corpus which have an impact on our results. Through the experiments we successfully show that even by using a a small set of features the GA improves the classification of definitional sentences.

## 6.1 Experiment Description

The purpose of the GA is to determine the weights to a set of predefined features which can classify definitions or non-definitions. The weights symbolise the importance or relevance a feature has according to the quality of the classification of the sentences. For instance, if we have a feature which is able to capture only sentences manually marked as definitions, this feature should have a high weighting indicating a level of confidence in its classification.

The purpose of the experiment is to find the ideal configuration set up for the GA to be able to be used in future as a 'fixed machine' for this single purpose. We have mentioned several design issues in section 3.3 which influence the performance of the GA. Additionally, we do not know in advance what the ideal settings should be (population size, number of generations required for the GA to converge, the best performing fitness function). Thus the first experiment focuses on using different settings and comparing the results achieved. Once the configuration is set, we can also focus on the kappa parameter of the f-measure which gives more weighting to recall or precision in the calculation of the formula. It is necessary to test with different kappas since different definition extraction tasks require different emphasis. We claim that in an eLearning scenario recall should be given more importance since definitions will be presented to a human expert who can then select which of the proposed definitions will make it into the final glossary. However, precision must not be a detriment since if an expert is presented with a large number of incorrect definitions, the system will not be usable.

The outcome of this experiment is to analyse the results obtained to by different runs, upon which we can then decide the final setup the GA should consist of for future experiments.

### 6.1.1 Experiment Settings

The corpus, detailed in section 4.1, is made up of eLearning objects and has been manually annotated with definitions. Furthermore, the definitions have been categorised into six different categories. In this experiment we focus only on the is-a category. In this category there are 111 definitions and 21,112 non-definitional sentences. These sentences are used by the GA as the learning corpus. The sentences were analysed against the predetermined set of features in order to obtain vectors of 0s and 1s indicating the presence or absence of a feature in a sentence.

The GA parameters were set to a population size of 100, maximum generations of 1000, and a mutation rate 1%. The initial population, although created randomly, was generated with the same seed so that the comparison between methods can be carried out in a more restricted form. This ensures that the experiments are not affected with a one-off super individual which might have been created randomly in one run, but not in another.

The selection methods used were SUS with sigma scaling, Roulette Wheel, Boltzmann, Elite and Rank. For Roulette, SUS, Boltzmann and Rank we implemented two variants for crossover. The first is the traditional crossover which selects a random point for crossover (one-point crossover) and the second is bit-crossover, where crossover occurs at every bit. To differentiate between the two crossover techniques, we append 'Bit' to the selection method; thus 'SUS' refers to SUS selection with one-point crossover, whilst 'SUS-Bit' refers to SUS selection with bit-crossover.

### 6.1.2 Feature Set Used

The experiment aims at testing different GA configurations to find the best GA setup for assigning weights to a predetermined set of features. For the purpose of this experiment, the tests are limited to the is-a category, using a feature set that need not be thoroughly complete or representative for the is-a category. A simple set of features is being used so as to place us in a better position to determine in an easy manner whether the weights learnt to the respective features are indeed realistic ones and reflect the situation in the corpus. The following is the feature set used:

1. contains the verb "to be"
2. has sequence "IS A" ("to be" followed by a determiner)
3. has sequence "FW IS" (FW is a foreign word - in the example "The process of bringing up the operating system is called booting", booting is tagged as an FW[1].)

[1]The tag FW, foreign word, is used by a tagger usually to indicate that the word is not known in the current position. In this case 'booting' is at a position where the tagger expects a noun, however it might not have the word in its lexicon and thus has no knowledge of its category.

4. has possessive pronoun (I, we, you, they, my, your, it)
5. has punctuation mark in the middle of the sentence (such as a hyphen or colon)
6. has a marked term (keyword)
7. has rendering (italic, bold)
8. has a chunk marked as an organisation
9. has a chunk marked as a person
10. has a chunk marked as a location

We expect the second, third and sixth features to gain higher importance over the other features, although not necessarily equal. The last five features have not been used in the manually crafted grammar rules since it is not possible to express them in lxtransduce rules due to the hierarchical structure of the corpus annotations.

## 6.2 Results

The first set of experiments run were with the following configuration parameters:

- Population size — 100
- Maximum generations — 1000
- Mutation rate — 1%
- Kappa (for f-measure) — 1.0 (no preference to either precision or recall)

### 6.2.1 `Count` Fitness Functions

The `Count` fitness functions is based on the f-measure metric described in section 5.1.1.3, which in this case is calculated according to the counts of how the sentences classify (true positive, true negative, false positive, false negative).

Table 6.1: Results for `CountZero` technique

| Method | f-measure | Precision | Recall |
|---|---|---|---|
| Roulette | 0.02 | 0.01 | 0.95 |
| Roulette-Bit | 0.02 | 0.01 | 0.94 |
| SUS | 0.03 | 0.01 | 0.90 |
| SUS-Bit | 0.03 | 0.01 | 0.95 |
| Boltzmann | 0.01 | 0.01 | 1.00 |
| Boltzmann-Bit | 0.01 | 0.01 | 1.00 |
| Elite | 0.03 | 0.01 | 0.96 |
| Rank | 0.03 | 0.01 | 0.95 |
| Rank-Bit | 0.03 | 0.01 | 0.89 |

#### 6.2.1.1 `CountZero` Technique

Table 6.1 presents the precision, recall and f-measure for the best individual obtained for the different selection techniques using the `CountZero` fitness function described in 5.1.1.3. The `CountZero` technique takes zero as the classifying separator between definitions and non-definitions; i.e. sentences obtaining a score above zero are classified as definitions.

The results clearly demonstrate that this technique fairs badly, and although recall is very high, precision is too low. This means that although it captures practically all those sentences manually marked as definitions, the technique also classifies a high number of non-definitions as definitions, thus resulting in such poor precision performance. We can also conclude that since recall practically reaches 100%, all definitional sentences score above zero with the present feature set.

#### 6.2.1.2 `CountShifted` Technique

Apart from using zero as the classifying line, we consider shifting the classifier to find the most favourable dividing line between definitions and non-definitions.

Table 6.2: Results for `CountShifted` technique

| Method | F-measure | Precision | Recall |
|---|---|---|---|
| Roulette | 0.50 | 0.46 | 0.55 |
| Roulette-Bit | 0.45 | 0.38 | 0.57 |
| SUS | 0.57 | 0.62 | 0.52 |
| SUS-Bit | 0.57 | 0.64 | 0.50 |
| Boltzmann | 0.37 | 0.42 | 0.32 |
| Boltzmann-Bit | 0.41 | 0.32 | 0.55 |
| Elite | 0.57 | 0.62 | 0.52 |
| Rank | 0.53 | 0.62 | 0.46 |
| Rank-Bit | 0.53 | 0.60 | 0.47 |

The position is shifted until the best yielding f-measure is found.

Table 6.2 presents the f-measure, precision and recall for the `CountShifted` technique. Using this technique we observe that the results have improved considerably over the previous technique, obtaining a maximum of 57% f-measure, with precision of around 63% and recall around 51%.

### 6.2.2 `Distance` Fitness Functions

One of the fitness functions we experiment with is to use the distance rather than the count so as to indicate a more accurate level of confidence in the classification of sentences. This means that if the classifying line is at zero, sentence $s_1$ scores 3, and sentence $s_2$ scores 10, then we conclude that the classification of $s_2$ has a higher level of confidence than that of $s_1$. Again, we experiment with two different techniques, one where zero is taken as the classifying line, and the second is where the classifying line is shifted to find the best value.

Table 6.3: Results for `DistanceZero` technique

| Method | F-measure | Precision | Recall |
|---|---|---|---|
| Roulette | 0.01 | 0.01 | 1.00 |
| Roulette-Bit | 0.01 | 0.01 | 1.00 |
| SUS | 0.02 | 0.01 | 0.75 |
| SUS-Bit | 0.02 | 0.01 | 0.75 |
| Boltzmann | 0.01 | 0.01 | 1.00 |
| Boltzmann-Bit | 0.01 | 0.01 | 1.00 |
| Elite | 0.02 | 0.01 | 0.76 |
| Rank | 0.02 | 0.01 | 0.96 |
| Rank-Bit | 0.02 | 0.01 | 0.77 |

#### 6.2.2.1 `DistanceZero` Technique

In table 6.3 we present the results achieved for the `DistanceZero` technique. The results obtained here are very similar to the results obtained using `CountZero` technique shown in section 6.2.1.1. Again we manage to achieve a high recall at the expense of poor precision, resulting in a very low f-measure.

#### 6.2.2.2 `DistanceShifted` Technique

In table 6.4 we present the f-measure, precision and recall obtained for the `DistanceShifted` technique. We note a similarity in results obtained in the `CountShifted` technique, the latter being slightly better. The best f-measure obtained is by the SUS, SUS-Bit, Rank, Rank-Bit and Elite selection methods, each reaching a value of 54%, with a precision of 59% and a recall of 50%.

## 6.3 Evaluation

From the results presented in section 6.2, we can conclude that zero is not an effective separating line between definitions and non-definitions, and yields very

Table 6.4: Results for DistanceShifted technique

| Method | F-measure | Precision | Recall |
|---|---|---|---|
| Roulette | 0.39 | 0.29 | 0.59 |
| Roulette-Bit | 0.36 | 0.25 | 0.59 |
| SUS | 0.54 | 0.59 | 0.50 |
| SUS-Bit | 0.54 | 0.59 | 0.50 |
| Boltzmann | 0.52 | 0.51 | 0.52 |
| Boltzmann-Bit | 0.41 | 0.32 | 0.55 |
| Elite | 0.54 | 0.59 | 0.50 |
| Rank | 0.54 | 0.58 | 0.50 |
| Rank-Bit | 0.54 | 0.59 | 0.50 |

poor results. Although in both `CountZero` and `ShiftedZero` the average recall is over 90%, precision is around 2%, resulting in a very low f-measure score for all selection methods. As expected, shifting the separator results in better results in our experiments. `CountShifted` performs slightly better on average than `DistanceShifted` given their f-measure, both able to achieve over 50%.

In this section we evaluate the performance of the GA and the individuals it has learnt. We also present a set of experiments where we tested different kappa values for f-measure, influencing the learning algorithm towards favouring either recall or precision. Finally, we take one of the best individuals and discuss the weights learnt, what they represent and how sentences are classified using this individual. Through this qualitative evaluation we uncover certain annotation errors in our corpus.

### 6.3.1 Convergence and Selection Technique Performance

In table 6.5 we present the number of generations required for a particular run to converge to the best individual. In the case where the GA did not converge this is indicated with the letters DNC. Looking at the different selection techniques, both the Roulette Wheel and the Boltzmann selection methods were unable to converge except when using the `CountZero` function. A documented problem with the Roulette Wheel technique is that the number of times an individual is selected for crossover is different from its expected value (either larger or smaller). Mitchell (1998) states that in "an extremely unlikely series of spins of the roulette wheel could even allocate all offspring to the worst individual in the population". As for Boltzmann selection, de la Maza & Tidor (1991) found that it outperformed fitness-proportionate functions for a small set of problems. In our case, when analysing the Boltzmann technique, throughout the life cycle of the GA the best individual fluctuates constantly and never settles at a fixed position (converges). In the case of the Roulette Wheel, the fact that two parents are chosen randomly also seems to produce a fluctuating best fitness in the experiments.

As for the shifted techniques where the best classifying line is chosen, we believe that when two well-performing individuals are selected, but with very different thresholds[1], then it would be difficult to produce equally good offspring from such varying parents. On the other hand, one notes an overall better performance with the Elite selection method since the best individuals are kept whole and are thus not lost to crossover.

### 6.3.2 Results Summary for Best Performing Techniques

The best performing techniques are `CountShifted` with SUS, SUS-Bit and Elite, and `DistanceShifted` with SUS, SUS-Bit, Elite, Rank and Rank-Bit obtaining similar results. Table 6.6 presents a summary of the best performing results described previously, showing precision, recall and f-measure for the best resulting experiments.

We note that since the features chosen are not a complete set of possible features the results might improve with a larger feature set for the is-a category of definitions. When comparing these results to those achieved with the manually crafted rules in the LT4eL project, and considering that the experiment is using a restricted selection of features, we not only have managed to retain a high

[1]For example $ind_1$'s best f-measure results at 0.2 being the dividing line, and $ind_2$'s best f-measure happens at 0.7.

Table 6.5: Convergence of selection techniques over the different fitness functions

| **Method** | `CountZero` | `CountShifted` | `DistZero` | `DistShifted` |
|---|---|---|---|---|
| Roulette | 100 | DNC | DNC | DNC |
| Roulette-Bit | 800 | DNC | DNC | DNC |
| SUS | 1,000 | 200 | 1,000 | 200 |
| SUS-Bit | 700 | 300 | 600 | 300 |
| Boltzmann | 800 | DNC | DNC | DNC |
| Boltzmann-Bit | 600 | DNC | DNC | DNC |
| Elite | 300 | 200 | 400 | 400 |
| Rank | 600 | 700 | 400 | 400 |
| Rank-Bit | 500 | 200 | 400 | 300 |

Table 6.6: Results for best experiments

| **Method** | **F-measure** | **Precision** | **Recall** |
|---|---|---|---|
| SUS (CS) | 0.57 | 0.62 | 0.52 |
| SUS-Bit (CS) | 0.57 | 0.64 | 0.50 |
| Elite (CS) | 0.57 | 0.62 | 0.52 |
| SUS (DS) | 0.54 | 0.59 | 0.50 |
| SUS-Bit (DS) | 0.54 | 0.59 | 0.50 |
| Elite (DS) | 0.54 | 0.59 | 0.50 |
| Rank (DS) | 0.54 | 0.58 | 0.50 |
| Rank-Bit (DS) | 0.54 | 0.59 | 0.50 |

recall, but also increased precision to over 62% from just 17% obtained using the manually crafted rules.

Table 6.7: Results for `kappa = 0.5`

| **Method** | **F-measure** | **Precision** | **Recall** |
|---|---|---|---|
| SUS CountShifted | 0.62 | 0.70 | 0.42 |
| SUS DistanceShifted | 0.60 | 0.72 | 0.37 |
| SUS-Bit CountShifted | 0.62 | 0.75 | 0.37 |
| SUS-Bit DistanceShifted | 0.60 | 0.72 | 0.37 |
| Elite CountShifted | 0.62 | 0.71 | 0.41 |
| Elite DistanceShifted | 0.61 | 0.68 | 0.45 |
| Rank CountShifted | 0.62 | 0.71 | 0.41 |
| Rank DistanceShifted | 0.59 | 0.69 | 0.37 |
| Rank-Bit CountShifted | 0.60 | 0.68 | 0.41 |
| Rank-Bit DistanceShifted | 0.54 | 0.58 | 0.43 |

### 6.3.3 Different Values of Kappa in the F-measure Metric

So far the experiments described always used f-measure with kappa equal to 1.0 as its fitness function. By changing the kappa value it is possible to manipulate the f-measure in bias of either precision or recall. The advantage of using f-measure as the fitness function is that this bias can be heavily influenced to one side without altering the internal functions of the fitness function. Apart from experimenting with the traditional kappa values of 0.5 (favouring precision) and 2.0 (favouring recall), we also wanted to test with exaggerated values of 0.1 (practically the f-measure is equal to precision) and 32.0 (where f-measure is nearly equal to recall). These experiments aim at showing whether it is possible to influence the learning of weights that would favour recall or precision accordingly. If so, then it would be possible to focus the learning path of the GA according to the system's requirements, thus providing a flexible approach to learning weights of features.

Table 6.8: Results for `kappa = 0.1`

| **Method** | **F-measure** | **Precision** | **Recall** |
|---|---|---|---|
| SUS CountShifted | 0.90 | 1.00 | 0.08 |
| SUS DistanceShifted | 0.65 | 1.00 | 0.02 |
| SUS-Bit CountShifted | 0.87 | 1.00 | 0.06 |
| SUS-Bit DistanceShifted | 0.85 | 1.00 | 0.05 |
| Elite CountShifted | 0.83 | 1.00 | 0.05 |
| Elite DistanceShifted | 0.83 | 1.00 | 0.05 |
| Rank CountShifted | 0.83 | 1.00 | 0.05 |
| Rank DistanceShifted | 0.83 | 1.00 | 0.05 |
| Rank-Bit CountShifted | 0.83 | 1.00 | 0.05 |
| Rank-Bit DistanceShifted | 0.87 | 1.00 | 0.06 |

#### 6.3.3.1 Experiments with Kappa Set to 0.5

Table 6.7 presents the f-measure, precision and recall for the experiments with kappa set to 0.5. This value favours precision slightly more than recall. Previously our best results stood at f-measure equal to 0.57, precision 0.63 and recall at 0.51. We note that setting kappa to 0.5 meant that the GA learnt weights which give a higher precision, resulting in a higher f-measure. The average figures have managed to improve precision to over 0.70, whilst recall was lowered to 0.42. This shows that it is impossible to improve precision without affecting, even slightly, recall. Although precision increased by 7 points, recall has been reduced by 10 points.

#### 6.3.3.2 Experiments with Kappa Set to 0.1

In table 6.8 we show the results for kappa set to 0.1. In this case f-measure is completely biased towards precision, emphasising that we would like to classify only correct definitions. The results show that indeed the experiments did learn to classify only correct definitions reaching complete precision, however the values for

Table 6.9: Results for `kappa = 2.0`

| **Method** | **F-measure** | **Precision** | **Recall** |
|---|---|---|---|
| SUS CountShifted | 0.54 | 0.46 | 0.56 |
| SUS DistanceShifted | 0.51 | 0.51 | 0.51 |
| SUS-Bit CountShifted | 0.53 | 0.39 | 0.59 |
| SUS-Bit DistanceShifted | 0.52 | 0.59 | 0.50 |
| Elite CountShifted | 0.54 | 0.46 | 0.56 |
| Elite DistanceShifted | 0.52 | 0.60 | 0.50 |
| Rank CountShifted | 0.53 | 0.41 | 0.58 |
| Rank DistanceShifted | 0.52 | 0.44 | 0.55 |
| Rank-Bit CountShifted | 0.53 | 0.40 | 0.57 |
| Rank-Bit DistanceShifted | 0.50 | 0.40 | 0.54 |

recall show that only few of the manually annotated definitions have actually been captured. If the experiment were to be applied to a fully automated definition extraction, it might be the case that such a setting would be preferred.

#### 6.3.3.3 Experiments with Kappa Set to 2.0

Table 6.9 presents the f-measure, precision and recall for the experiments with kappa set to 2.0. This value favours recall slightly more than precision. In the experiments with kappa set to 1.0, our best results stood at f-measure equal to 0.57, precision 0.63 and recall at 0.51. From the results of this experiment we see that at most, recall improved by 8 points to 0.59. However, in that case precision decreased by 24 points to 0.39. It is clear that in this case, the emphasis on recall reflects quite negatively on precision — something which in general we would like to avoid.

Table 6.10: Results for `kappa = 32.0`

| **Method** | **F-measure** | **Precision** | **Recall** |
|---|---|---|---|
| SUS CountShifted | 0.94 | 0.02 | 0.98 |
| SUS DistanceShifted | 0.94 | 0.06 | 0.95 |
| SUS-Bit CountShifted | 0.94 | 0.08 | 0.95 |
| SUS-Bit DistanceShifted | 0.94 | 0.08 | 0.95 |
| Elite CountShifted | 0.95 | 0.02 | 0.98 |
| Elite DistanceShifted | 0.94 | 0.08 | 0.95 |
| Rank CountShifted | 0.94 | 0.08 | 0.95 |
| Rank DistanceShifted | 0.94 | 0.04 | 0.96 |
| Rank-Bit CountShifted | 0.94 | 0.07 | 0.95 |
| Rank-Bit DistanceShifted | 0.94 | 0.06 | 0.95 |

#### 6.3.3.4 Experiments with Kappa Set to 32.0

In table 6.10 we show the f-measure, precision and recall obtained for the experiments with kappa set to 32.0. This setting gives a definite bias to recall, with f-measure resulting to practically the same value of recall. We note that precision is effected negatively by this setting and although we successfully capture all definitions, we also capture a high number of non-definitions.

#### 6.3.3.5 Conclusion on Results with Kappa Values

It is clear from the experiments that it is indeed possible to influence the learning of weights to features in such a way as to bias for either precision or recall. As has been pointed out previously, it largely depends on the system requirements the technique will be applied to. If there will be human expert intervention in approving the list of definitions (such as in an eLearning scenario where a tutor would select the definitions that will constitute the final glossary), then it is worthwhile considering an f-measure that favours recall. Notwithstanding, even in such a scenario precision remains an important factor, as a system which presents

Table 6.11: The best individuals under the `CountShifted` technique

| **Feature** | **SUS** | **SUS-Bit** | **Elite** |
|---|---|---|---|
| (1) contains the verb “to be” | 0.60 | 0.38 | 0.15 |
| (2) contains “is-a” | 0.84 | 0.92 | 0.86 |
| (3) contains “FW-is” | 0.07 | 0.55 | 0.19 |
| (4) has possessive pronoun | −0.21 | −0.21 | −0.21 |
| (5) has punctuation | 0.08 | −0.27 | 0.10 |
| (6) has keyword | 0.63 | 0.97 | 0.58 |
| (7) has rendering | −0.30 | −0.30 | −0.30 |
| (8) has organisation chunk | −0.38 | −0.38 | −0.38 |
| (9) has person chunk | 0.25 | 0.00 | 0.15 |
| (10) has location chunk | 0.11 | 0.20 | −0.08 |

too many incorrect ‘definitions’ will be unusable. It is difficult to judge where the ideal setting lies, as one would have to carry out a qualitative assessment of such a system from a usability perspective. Such an assessment is outside the scope of this thesis, and would be applicable not just to this technique, but to any technique being implemented for definition extraction.

### 6.3.4 The Meaning of an Individual

The experiments produced different sets of weights for the features used described in section 6.1.2. Table 6.11 presents the weights learnt by the best performing techniques for the `CountShifted` fitness function.

It is interesting to note that the features which received negative weights (possessive pronoun, rendering, and organisation chunk) from each experiment resulted in the GA learning the same weight for these features. This demonstrates a level of confidence in that these particular features are not helpful in detecting definitions. We also note that the first feature *contains the verb “to be”* is a subset of the feature *contains “is-a”* and that the weight learnt for the first feature is always smaller than the weight learnt for the second feature. This is because there are far more generic sentences having the verb "to be" than there are sentences with the sequence "is a". The latter is, as expected, more common in our category of definitional sentences. In fact the weights learnt do reinforce the perception that the sequence "is a", even if commonly found in all types of sentences, is a very important feature in our definitional set of sentences. Another important feature is the *has keyword* feature which generally has a high weighting. In section 6.1.2 we stated that we expected the second, third and sixth features to be recognised as important features. In fact these have been given positive weights by the three individuals. Another feature, *has person chunk*, has also been identified as a positive feature, although not as important and the other three features. We also note that there is sometimes a variance between the weights learnt for each feature. For instance, the feature *has punctuation* was recognised as a positive feature by the SUS and the Elite techniques, and as a negative feature by the SUS-Bit technique. It must be pointed out that the GA learns the best set of weights rather than individual weights, so this fluctuation is expected since in one experiment the importance of a feature might have been shifted to another. These different experiments produced very similar f-measure, precision and recall and thus shows that the variants in weights, although might affect the classification of certain sentences, do not affect the overall classification. Thus, if one set of weights managed to classify ten sentences, and another set of weights classified a different set of ten sentences as definitions, the overall result still remains the classification of ten definitional sentences.

In table 6.12 we show the weights of the best individuals learnt under the `DistanceShifted` technique. With this technique it transpires that only *has rendering* and *has organisation chunk* result as negative features. The other features are somewhat positive. Again the second, third and sixth features have the highest weights. The remaining features all have weights of around or below 0.10. The idea behind the distance technique is to create a certain level of confidence in the classification of sentences where, rather than just taking the count of the classifications, we use the distance of the score. The weights learnt in this setting also reflect this type of classification where features like four, five, nine and ten are not features which can clearly separate non-definitions from definitions.

Table 6.12: The best individuals under the `DistanceShifted` technique

| **Feature** | **SUS** | **SUS-Bit** | **Elite** |
|---|---|---|---|
| (1) contains the verb “to be” | 0.04 | 0.04 | 0.00 |
| (2) contains “is-a” | 0.92 | 0.79 | 0.94 |
| (3) contains “FW-is” | 0.35 | 0.45 | 0.39 |
| (4) has possessive pronoun | 0.05 | 0.00 | 0.01 |
| (5) has punctuation | 0.02 | 0.02 | 0.01 |
| (6) has keyword | 0.95 | 0.88 | 0.97 |
| (7) has rendering | −0.30 | −0.30 | −0.30 |
| (8) has organisation chunk | −0.38 | −0.38 | −0.38 |
| (9) has person chunk | 0.11 | 0.02 | 0.01 |
| (10) has location chunk | 0.07 | 0.02 | 0.03 |

### 6.3.5 Classification of Sentences

In the following section we analyse the classification of sentences by taking the best individual and applying it to the training corpus. The purpose is to measure how the effective the weights learnt are at classifying definitions and non-definitions. Table 6.13 shows how the corpus was classified, giving the score range the sentences obtained, and the number and percentages of the positive and negative sentences according to their classification.

These results show that there is at least some certainty in classifying non-definitions correctly with only 3% (655 from 21222 sentences) being classified as definitions. The problem remains that even this 3% remains a high figure when compared to the number of definitional sentences (111 sentences). However, 94% of non-definitions captured as definitions revolve mainly around the 9 — 0 score range. From this figure we confirm how difficult the task of definition extraction is. The problem remains with those sentences which share linguistic characteristics and yet are not necessarily definitions. To further analyse the results, we look at a qualitative assessment of the classification of sentences to understand how

Table 6.13: Percentage of sentences

| Score Range | Positive Sentences | Negative Sentences | Percentage of Positives | Percentage of Negatives |
|---|---|---|---|---|
| $50 - 40$ | 4 | 0 | 100% | 0% |
| $39 - 30$ | 35 | 16 | 69% | 31% |
| $29 - 20$ | 2 | 4 | 33% | 67% |
| $19 - 10$ | 15 | 19 | 44% | 56% |
| $9 - 0$ | 21 | 626 | 3% | 97% |
| $-1 - -10$ | 25 | 547 | 4% | 96% |
| $-11 - -20$ | 4 | 77 | 5% | 95% |
| $-21 - -30$ | 2 | 2792 | 0% | 100% |
| $-31 - -40$ | 2 | 10893 | 0% | 100% |
| $-41 - -50$ | 0 | 284 | 0% | 100% |
| $-51 - -60$ | 1 | 4121 | 0% | 100% |
| $-61 - -70$ | 0 | 1267 | 0% | 100% |
| $-71 - -80$ | 0 | 27 | 0% | 100% |
| $-81 - -90$ | 0 | 87 | 0% | 100% |
| $-91 - -100$ | 0 | 362 | 0% | 100% |

Table 6.14: Classification of definitions as non-definitions

| Problem Summary | Sample Sentence | # |
|---|---|---|
| Good Sentences | A version of UNIX that is popular and available for no cost today, even for less powerful personal computers, is Linux. | 11 |
| Wrongly annotated as a definition | Beads are of several forms (circle, square, and triangle) and colors (red, green, blue, yellow, black, and white). | 5 |
| Grammatical Problem | ePortfolio (...) is [a] collection of electronic documents.... | 1 |
| Questionable Definitions | Notebook computers are complete computers with full functionality and with all major devices of the desktop computer. | 1 |
| Conversion Problem | [A trackball is a ball about the size of] an egg embedded into a panel , which you can rotate. | 11 |

further improvements in classification could be carried out. Due to time and resource restrictions, it is not possible to analyse the whole training corpus. We focus on the following questions:

1. Why are the definitional sentences with negative scores ranging from $-1$ to $-60$ achieving such a low score and are thus not being classified as definitions?

2. Why are the top 39 non-definitional sentences achieving such a high score ranging from 10 to 39 and are thus being classified as definitions?

#### 6.3.5.1 Definitions Classified as Non-definitions

In order to understand the results obtained at this stage, we first look as those sentences which were manually tagged as definitions, but have not been classified so by the GA. We analyse why these sentences are so classified by grouping them into groups as follows:

- Good definitional sentences, which should have been captured.

- Sentences which have been wrongly annotated as definitions, and thus have been correctly identified as non-definitions by the GA.
- Sentences with grammatical problems, originating from the text used.
- Questionable definitional sentences which have the structure of a definition, and seem to be describing something, but in reality such sentences would not be considered a complete or usable definition.
- Sentences whose structure has been changed during the conversion process; e.g. joining of two sentences as one, or splitting a sentence into two.

Table 6.14 presents the problems outlined above, a sample sentence from our training data and the number of sentences which fall under each problem category. This exercise has helped us to identify not only what the problems are, but why are they happening and if they are avoidable. In the case where the sentences are good definitions, these are not being captured because the features used are not rich enough and do not cater for their sentence structure. In the example given in the table, we do not find the sequence “is a”, but rather “is Linux”. Similarly, in some other sentences under this category, the “is a” phrase was interrupted by another word, e.g. “is in fact a”, which was also not present in our feature set. It is clear that the features used were not flexible enough to cater for these differences, omitting 11 sentences from being classified as definitions. In the case of sentences wrongly annotated as definitions the problem lies in human error and is difficult to eradicate entirely. At times it is difficult to judge with certainty if a sentence is a definition or not, giving rise to questionable definitions. The definition of a notebook computer is a circular one, where it is defined by the term computer itself. The sentence on its own does not define a notebook computer, but its structure still gives the impression that it is a definition. Finally, some problems arose from the conversion process where a sentence was either split in two, or two sentences joined as one. In either case it is difficult to classify such definitions and thus it is acceptable to have such an error that is not dependent on the definition extraction process itself.

Table 6.15: Classification of non-definitions as definitions

| Problem Summary | Sample Sentence | # |
|---|---|---|
| Should be marked as definitions | An operating system for which the source code is freely available is Linux. | 21 |
| Questionable sentences | The most important input device for computers currently is the keyboard, which serves mostly for text input, and, to a large extent, imitates the keyboards of typewriters. | 4 |
| Non-definitions | Objects are given names; an object is the value of its name. | 14 |

#### 6.3.5.2 Non-definitions Classified as Definitions

In the case of the classification of non-definitions as definitions, table 6.15 shows why 39 non-definitional sentences receiving a positive score between 39 and 10 have been classified as definitions. Half of these sentences turn out to be definitions which were missed out by the human annotators. This implies that our experiment results would actually improve if the data were to be re-annotated with these corrections. Since it is not the scope of this thesis to provide a good quality corpus, we decided to leave the annotations as is, thus also factoring in human error in our results.

## 6.4 Conclusion

In this chapter we described the setup of the GA experiment and the results for the different settings. We show that the best resulting settings for the GA are using the `CountShifted` technique with the SUS, SUS-Bit and Elite selection methods, and the `DistanceShifted` technique with SUS, SUS-Bit, Elite, Rank and Rank-Bit selection methods. From a precision of 17% achieved using manually crafted rules in the is-a category, we manage to achieve a maximum of 64% precision. When evaluating the classification of sentences from a qualitative perspective, we discovered errors present in the manual annotation of the definitions, with some definitional sentences which had not been annotated as definitions, and other questionable or non-definitional sentences having been annotated as definitions. Taking these factors into consideration, it would be possible to further improve the results achieved. We also show that by adjusting the kappa value used by the fitness function in the f-measure metric, it is possible to influence the learning of weights which could favour either precision or recall. We discussed how the best individuals learnt are related to the set of features used and showed how they can be interpreted to show the relevance of each of the features.

One of the expected outcomes of this experiment was that the set of features chosen did not cater for all definitional sentences in our corpus, thus classifying some definitions as non-definitions. The purpose of the GP experiment aims at tackling this problem by identifying possible features automatically, described in the following chapter.

# Chapter 7

# Genetic Programming for Definition Extraction

In this chapter we describe the GP experiments carried out and the results and rules learnt in the different definitional categories. We describe the way the individual is encoded and implemented, the fitness function used, the selection method, and other configurational aspects in the experiment setup. The central point of this experiment is to learn rules which could extract definitions. Thus the results present both f-measure values obtained with the rules learnt, and the rules themselves. We explore different settings for the GP by increasing the search space through the introduction of more features that could be part of the rules.

The experiments focus on the three definitional categories described previously, the is-a category, the verb category and the punctuation category. We also ran a generic experiment to capture all types of definitions without categorical restrictions. In all categories we have succeeded to learn rules that do capture some of the definitions. In general, when analysing the rules learnt by the GP, we find a similar subset to the manually crafted rules. This observation is also reflected by the similar f-measure values obtained. In the punctuation category we successfully manage to obtain better results than those obtained by the manually crafted rules.

## 7.1 Experiment Description

The GP experiment aims at having a program which creates rules in a random manner and tests if they are good candidate rules to extract definitions or not. Since the principles of the GP are the same as that of a GA, the best candidates are kept and allowed to evolve further in an experiment's lifetime. The rules are restricted to cover the language specified in section 5.1.2.1, yet different experiment runs will cover different sets of linguistic objects. In the experiments described in this chapter, linguistic objects are limited to either part-of-speech items (e.g. VBZ — verb, 3rd person singular, present), classes of part-of-speech (e.g. all verbs), and specific words or their base form (e.g. called, call). A rule learnt by the GP can also be a conjunction of different sub-rules, with each sub-rule being a different sequence or occurrence of linguistic objects.

The main objective of the GP experiment is to produce an automated process which can identify rules by trying out as many variations as computationally possible or feasible. When a linguist expert creates rules, he already has a predefined knowledge of certain grammar restrictions and thus avoids testing ungrammatical rules. The GP has no such knowledge, and thus will probably test a majority of ungrammatical rules for which no sentence (whether definitional or non-definitional) would match. However, as the evolution process proceeds, the rules that do capture sentences are passed on to future generations, modified (through genetic operations such as crossover and mutation) to evolve into possibly better performing rules. These are tested against the set of definitional and non-definitional sentences in order for the GP to be able to score their performance. Of course, a linguist would change rules based on his knowledge, and does not attempt any conceivable configuration knowing a priori that the result would be an ungrammatical rule. The GP experiment might attempt every conceivable structure or form of a rule, which might result in a worse performance.

The set of linguistic objects specified for a particular experiment influences the size of the search space. Whilst this was not an issue in our GA experiment since we worked with a fixed set of features, in the GP experiment this issue comes to the forefront of our experiments since it directly affects the results. A small set of linguistic objects reflects a small search space by limiting the possible options for the rules which can be composed by the GP. The further the set of linguistic objects is increased, the larger the search space becomes since the GP is presented with more possible options or paths that could be traversed. It is not possible to test each and every path as this would be an infinite exercise. Thus the GP is trying out possible rules that are created either randomly (through the initial population or by mutation) or genetically (crossover). This means that the rules and the elements within the rules are largely dependent on an amount of randomness. If, in the initial population, a particular linguistic object is left out (randomly), the search space of rules containing that element is not explored until it is introduced by mutation. Similarly, it can happen that if a certain linguistic object is indeed present in the beginning, but the structure of the rule is ungrammatical or matches only non-definitions, such a rule would perform poorly and probably be discarded within the following few generations. Thus the linguistic object present in that rule is also lost through the evolution process. For these reasons we try out different sets of linguistic objects according to the category of definitions being tackled.

A difference worth noting between the GP and the GA is in the fitness function. In the GA experiment we had tested different functions by shifting the classifying line which scores definitions or non-definitions and aiming at getting a level of confidence in the classification of the sentences. In the case of the GP we are primarily interested in the resulting rules that are able to capture definitions. It would make our problem of rule-finding much harder had we had to include a similar principle of level of confidence, since it would have made the fitness function stricter than that of a general classifier. Thus, in this experiment we maintain the classical view of classifying sentences as definitions that score above zero when calculating f-measure.

### 7.1.1 Encoding of the Individual

An individual represents a candidate rule which could be used to classify definitions. In section 5.1.2.1 we described the structure of the individual, specifying that it is composed of a conjunction (only at top level) of features, and that features translate to different sequences and occurrences of either linguistic objects,

```
typedef struct Feature {
  enum featureType ftag;
  union FeatureList {
    struct sFeature *simpleF;
    struct cFeature {
      struct sFeature *sf1;
      struct Feature *sf2;
    } compoundF;
  } featureList;
} FEATURE;
```

Figure 7.1: Node structure implementation of a feature in C

```
typedef struct SimpleFeature {
  enum TAG tag;
  union currFeature {
    enum lObj object;
    struct followby {
      struct sFeature *sfptr1;
      struct sFeature *sfptr2;
    } fb;
    struct or1 {
      struct sFeature *sfptr1;
      struct sFeature *sfptr2;
    } or;
    struct zeromany {
      struct sFeature *sfptr1;
    } zm;
    struct zeroorone {
      struct sFeature *sfptr1;
    } zo;
  } currf;
} SIMPLEFEATURE;
```

Figure 7.2: Node structure implementation of a simple-feature in C

the empty string or match any element. We also specified the BNF determining the language over which rules could be created. The BNF has been implemented in C as a tree structure by implementing two types of nodes. Figure 7.1 shows the implementation of the node at top level where conjunction can occur, giving the feature the possibility to be a structure of type feature or a conjunction of simple-feature together with feature. Figure 7.2 shows the implementation of the node simple-feature which can have different occurrences and sequences of linguistic objects, empty string or any element.

The tree structure is used by the GP to perform the necessary operations during a generation on a population. Apart from the manipulation of an individual, we need to interpret the individual in terms of its success as a possible candidate solution. Thus, an individual (tree) is translated into a regular expression which can be applied directly to string sentences in our corpus. This was done by using a standard GNU C library for regular expressions[1]. When a regular expression, which is another interpretation of the individual, is applied to a sentence, the function will either return a match or a fail. Those sentences which match the rule are considered to be classified as definitions, whilst those that fail are classified as non-definitions. In the next section we will discuss the fitness function which uses the interpretation of an individual in order to evaluate the population.

In order to restrict slightly the way a tree is randomly generated, we introduced weights or probabilities to each possible node, so as to give a higher preference to certain types of nodes that others. The initial population for each experiment is generated randomly, influenced by these weights. The same technique is used when generating of sub-trees required during the mutation process. At a top level of the node generation, the GP was given a 50% chance of going either way for a simple-feature or a conjunction of a feature and a simple-feature. At simple-feature level, we set weights at different levels according to what was deemed to have a higher priority. The highest probability was placed on the linguistic object since they are the main focus of our rule learning. An issue in the generation of trees from a computational and operational perspective is their

[1] The GNU C library and its POSIX.2 functions used are described in more detail here: `http://www.gnu.org/software/libtool/manual/libc/Regular-Expressions.html` last accessed 15 December 2008.

size. In order to introduce an element of control, we set an arbitrary depth limit of ten during the creation of a tree. However, this depth limit in not respected during the genetic operations, meaning that the depth of a tree can grow through evolution. The following are the weights used for the different possible nodes at simple-feature level:

**Linguistic Object** 30%, with all possible linguistic elements having equal probability;

**Followed By** 14%;

**Either Or** 14%;

**Zero or Many** 14%;

**Zero or One** 14%;

**Empty String** 7%; and

**Any Element** 7%.

## 7.1.2 General GP Configuration

As described in chapter 5, the GP uses f-measure as its fitness function. The values to calculate the fitness function will be based on how sentences match against the individuals (rules) and their classifications will contribute to the following values:

- True Positives - manually marked definitional sentences classified as definitions by the system (correct classification);
- False Negatives - manually marked definitional sentences classified as non-definitions by the system (incorrect classification);
- False Positives - manually marked non-definitional sentences classified as definitions by the system (incorrect classification);
- True Negatives - manually marked non-definitional sentences classified as non-definitions by the system (correct classification),

which are then used to calculate precision, recall and f-measure. The latter is the metric used to score individuals, and used by the selection function to select those individuals for crossover. We use the Elite selection technique implemented for the GP (keeping the top individuals), with the remaining individuals being selected for mating through the SUS selection technique with sigma scaling. This means that a predetermined number of the best individuals are copied into the next generation without any modifications. The remaining places in the new population are filled in by the children of the current generation, with selection for mating being done through the SUS proportionate-fitness function. Thus top individuals not only remain intact, but also have the opportunity to mate with other individuals in the population. In our experiments we found that keeping the top 10% of the population gave the best results, and increasing the percentage did not seem to improve results any further.

### 7.1.3 Linguistic Objects and Structure of Rules

The rules learnt by the GP are sequences of what we refer to as linguistic objects. Several of the related work reviewed in chapter 2, such as Malaisé *et al.* (2004); Muresan & Klavans (2002); Park *et al.* (2002), and work carried out in the LT4eL project described in chapter 4 focus primarily on part-of-speech when manually crafting rules. Furthermore, in experiment described in Westerhout & Monachesi (2008), the authors state that using deep parsers for the purpose of definition extraction did not improve results. Based on these observations, we focus on part-of-speech annotations and specify our rules according to the Penn Treebank tagset (Marcus *et al.*, 1993) used to annotate the corpus[1] or occurrence of particular words and their base form.

In describing the set of linguistic objects being applied to a particular experiment(s), we choose to either use specific part-of-speech tags such as NN (noun, common, singular or mass); NNP (noun, proper, singular); NNPS (noun, proper, plural); NNS (noun, common, plural). Otherwise we generalise these tags into

[1]The Penn Treebank tagset and an explanation of the tags used is available here: `http://www.comp.leeds.ac.uk/amalgam/tagsets/upenn.html` last accessed 15 December 2008.

one class and refer to them as nouns. The principle is applied to other categories in a similar manner.

### 7.1.4 Definitional Categories

In section 4.3.1 we described the way definitions were categorised in the LT4eL project and subsequently how our definitional corpus is divided. Out of the six categories, we believe that the first three categories are ideal for machine learning. These are the is-a category, verb category and punctuation category. The layout category depends highly on HTML elements such as table layout which were not maintained during the conversion process to XML. Even if so, it would be difficult to include such layout information into the rule learning process. The anaphora category (where 'this' is used in the definitional sentence to refer to the defining term) requires anaphora resolution and the final category, other, is a catch-all category with no particular characteristics. Thus our GP experiments focus on the first three categories. A final experiment attempts at learning rules without taking the definitional categories into consideration and thus includes all six categories.

## 7.2 Results and Evaluation

The first set of experiments focused on the is-a category, where most work is generally carried out. Initially, we test the GP with a rather 'small' search space, limiting the number of linguistic objects to five. At this stage we are also interested in experimenting with elements such as population size and how many generations are necessary for the GP to converge. Once the initial observations are made, we then increased the set of linguistic objects to nine and succeeded to improve the results, albeit slightly. We also carry out similar experiments for the verb category and the punctuation category. For the verb category we try using two different sets of linguistic objects; one set emphasises on the part-of-speech, whilst the other included certain words found usually in cue phrases such as 'call', 'define' or 'known'. The last experiment attempts to capture all definitions in the six categories in order to evaluate the effectiveness of categorisation and the ability of a GP to possibly learn rules over such a large search space.

The best individual of each of the experiments is presented in appendix A in full. In this chapter we present important snippets of these rules to discuss interesting aspects of what the GP has learnt.

### 7.2.1 GP Results for the Is-a Category

In our first set of experiments, we aim to determine how the GP should be set up to achieve the best results. We ran a few experiments with a relatively small search space (of five linguistic objects) with the idea of trying out different variables for population size, number of generation runs and the number of top individuals the Elite selection method should copy to the new generation. The set of linguistic objects chosen are:

- the word *is*
- noun
- adjective
- adverb
- modal

Notably, certain important parts-of-speech for this category have been purposely left out to determine if the use of *any* defined in our language will be included in the rules learnt by the GP to result in a rule such as *noun . is . any . noun*, with *any* replacing the determiner.

Table 7.1 shows the settings of five different runs together with the f-measure achieved by the best individual. The first three experiments are exactly the same in settings, with the aim of giving an overall indication of performance. In the case of Exp2 the run did not converge (indicated with the letters DNC) within 100 generations, and since it was fairing better than Exp1 and Exp3, we ran the experiment with the same seed, allowing it to run for up to 500 generations. By 270 generations it converged with no further improvement in the population

Table 7.1: Results for GP experiment one

| **Configuration** | **Exp1** | **Exp2** | **Exp3** | **Exp4** | **Exp5** | **Exp2b** |
|---|---|---|---|---|---|---|
| Population Size | 200 | 200 | 200 | 500 | 1000 | 200 |
| Elite $n$ | 10 | 10 | 10 | 50 | 50 | 10 |
| Generations | 100 | 100 | 100 | 500 | 500 | 500 |
| Converged | 70 | DNC | 100 | 120 | 430 | 270 |
| F-measure | 0.21 | 0.24 | 0.18 | 0.24 | 0.26 | 0.25 |

after that. The improvement overall is minimal, and on average, 100 generations seemed enough to achieve a respectable performance by the GP. Exp4 and Exp5 focus on trying out the same pool of linguistic objects but with a larger population size. We also allowed the GP to run for up to 500 generations. With a population of 500 individuals, Exp4 converged by the 120th generation and reached an f-measure of 24%. Exp5 had a population of 1,000 individuals and converged in generation 430, reaching an f-measure of 26%.

If we solely rely on f-measure as a metric of performance, Exp5 would be the best performing individual. However, if one had to look at the actual rules learnt, we are inclined to say that Exp5 learnt the longest and most illegible rule out of all the experiments run. As shown in appendix A the rule description for the best individual in Exp5 runs over almost 11 pages, whereas most rules learnt are simply a few lines in length. Even in the case of Exp4 the rule runs over only one page. A general technical observation over the experiments run is that the longer a GP took to converge, the more complex (and not necessarily sophisticated) the resulting rules were.

Overall, all the experiments learnt the sequence *noun is any noun*. This is a positive aspect since it is generally considered as the most obvious rule from a human's perspective in the is-a category. Other rules learnt were either a subset of this sequence, such as *is any noun*, or other rules such as *adjective any any noun*, and *noun any noun*. More complex rules included $\text{NN} \cdot \textit{any}? \cdot \text{IS} \cdot (\textit{any} + (\text{NN} \cdot \text{IS} \cdot \text{NN})) \cdot \text{NN}$, where the + indicating an *or* results in the rule having two paths:

- NN · *any*? · IS · *any* · NN
- NN · *any*? · IS · NN · IS · NN · NN

In such a rule, if one of the paths never matches, but the other matched some definitions, the rule as a whole will always perform well even if part of it might never be executed. The GP views rules globally and has no knowledge on the validity of sub-rules. This means that the GP is bound to learn duplicate or 'useless' information, and throughout the process we do not have any means of simplifying or reducing the complexity of the rules. It would also be unwise to throwaway rules which do not match any sentences in our corpus since there is always a possibility, no matter how slight, that such a rule might match an unseen sentence.

In the second set of experiments in the is-a category, we experiment with increasing the search space by adding further linguistic objects to the above set as follows:

- determiner
- foreign word or symbol
- preposition
- which

First we ran two experiments, ExpF1 and ExpF2, in order to analyse the affect the additional linguistic objects have on the GP. With a population size of 1,000 and allowed to run over 500 generations, both experiments converged rather early within 30 generations. The Elite selection techniques kept 1% of the individuals as is in the first experiment, and 5% in the second. This did not seem to hinder the results so we decided to keep 2% of the population for the remaining runs. We also decided to retain the population size at 500 since in the previous experiments the larger populations risked converging later and resulting in complex rules.

Table 7.2: Results for GP experiment with a larger search space

| **Experiment** | **F-measure** | **Precision** | **Recall** | **Converged** | **Population Size** | **Elite** |
|---|---|---|---|---|---|---|
| ExpF1 | 0.25 | 0.20 | 0.33 | 30 | 1000 | 10 |
| ExpF2 | 0.25 | 0.20 | 0.33 | 30 | 1000 | 50 |
| ExpM1 | 0.25 | 0.20 | 0.33 | 50 | 500 | 10 |
| ExpM2 | 0.25 | 0.20 | 0.33 | 50 | 500 | 10 |
| ExpM3 | 0.25 | 0.20 | 0.33 | 40 | 500 | 10 |
| ExpM4 | 0.28 | 0.22 | 0.39 | 90 | 500 | 10 |
| ExpM5 | 0.21 | 0.15 | 0.31 | 90 | 500 | 10 |
| ExpM6 | 0.25 | 0.20 | 0.33 | 30 | 500 | 10 |
| ExpM7 | 0.26 | 0.22 | 0.33 | 80 | 500 | 10 |
| ExpM8 | 0.26 | 0.22 | 0.32 | 100 | 500 | 10 |
| ExpM9 | 0.26 | 0.20 | 0.36 | 100 | 500 | 10 |
| ExpM10 | 0.26 | 0.22 | 0.33 | 60 | 500 | 10 |

The experiment ExpF1 learnt the rule *noun is determiner noun*, while experiment ExpF2 learnt *noun is determiner* and *is determiner determiner*[1]. The remaining experiments invariably learnt *noun is determiner noun*, with some of the runs learning other interesting features as follows:

**ExpM3** learnt the rule:

- $\mathrm{NN} \cdot ((((\mathrm{DET} \cdot \mathrm{IS}^{*})^{*} \cdot ((((\mathrm{IS} \cdot \mathrm{IS}^{*})^{*} \cdot (\mathrm{IS}^{2} \cdot \mathrm{DET})^{*} \cdot \mathrm{IS}^{*} \cdot \mathrm{DET})^{*} \cdot \mathrm{DET} \cdot \mathrm{IS})^{*} \cdot (\mathrm{IS}^{2} \cdot \mathrm{DET})^{*})^{*} \cdot \mathrm{IS}^{*})^{*} \cdot \mathrm{DET})^{*} \cdot \mathrm{MD})^{*} \cdot \mathrm{IS} \cdot \mathrm{DET} \cdot \mathrm{NN}$

This rule has included *modal* (MD) as an optional with one of the possible paths being that of *noun modal is determiner noun.*

**ExpM4** achieved the highest F-measure (0.28) with the rule:

- $\mathrm{NN}\cdot((\mathit{any}^{*}\cdot((\mathrm{NN}^{4}\cdot\mathrm{NN}^{*}\cdot\mathrm{IS}\cdot\mathrm{DET}\cdot\mathrm{NN}\cdot((\mathit{any}^{*}\cdot(((\mathit{any}\cdot(\mathrm{NN}\cdot(\mathit{any}\cdot(\mathit{any}\cdot\mathrm{NN}\cdot\mathit{any}\cdot(\mathit{any}^{2}\cdot\mathrm{NN}\cdot\mathit{any}^{2})^{*}\cdot(\mathit{any}\cdot(\mathrm{NN}\cdot(\mathit{any}\cdot\mathrm{NN}^{2}\cdot(\mathrm{NN}^{2}\cdot\mathit{any})^{*}\cdot((\mathit{any}\cdot\mathrm{NN}\cdot(((\mathit{any}\cdot(\mathrm{NN}\cdot(\mathit{any}\cdot(\mathit{any}\cdot\mathrm{NN}\cdot\mathit{any}\cdot(\mathit{any}^{2}\cdot\mathrm{NN}\cdot\mathit{any}^{2})^{*}\cdot(\mathit{any}\cdot(\mathrm{NN}\cdot(\mathit{any}\cdot\mathrm{NN}^{2}\cdot(\mathit{any}^{*}\cdot(((\mathit{any}\cdot(\mathrm{NN}\cdot(\mathit{any}\cdot(\mathit{any}\cdot\mathrm{NN}\cdot\mathit{any}\cdot(\mathit{any}^{2}\cdot\mathrm{NN}\cdot\mathit{any}^{2})^{*}\cdot(\mathit{any}\cdot(\mathrm{NN}\cdot(\mathit{any}\cdot\mathrm{NN}^{2}\cdot(\mathit{any}\cdot\mathrm{NN}\cdot\mathit{any})^{*}\cdot\mathrm{IS}\cdot\mathrm{DET}\cdot\mathrm{IS}\cdot\mathrm{DET})^{*}\cdot\mathit{any}\cdot(\mathit{any}\cdot\mathrm{NN}\cdot(\mathit{any}\cdot\mathrm{NN}^{2}\cdot(\mathit{any}\cdot\mathrm{NN}\cdot\mathit{any})^{*}\cdot\mathrm{IS}\cdot\mathrm{DET}\cdot\mathrm{IS}\cdot\mathrm{DET})^{*}\cdot\mathit{any})^{*}\cdot\mathit{any})^{*}\cdot\mathit{any})^{*})^{*}\cdot\mathit{any})^{*}\cdot\mathit{any}\cdot(\mathit{any}\cdot\mathrm{NN}\cdot\mathit{any})^{*}\cdot\mathit{any})^{*}\cdot\mathit{any})^{*}\cdot\mathrm{NN}\cdot\mathit{any})^{*}\cdot\mathrm{NN}^{2})^{*}\cdot\mathit{any})^{*}\cdot\mathrm{IS}\cdot\mathrm{DET}\cdot\mathrm{IS}\cdot\mathrm{DET})^{*}\cdot\mathit{any}\cdot(\mathit{any}\cdot\mathrm{NN}\cdot(\mathit{any}\cdot\mathrm{NN}^{2}\cdot(\mathit{any}\cdot\mathrm{NN}\cdot\mathit{any})^{*}\cdot\mathrm{IS}\cdot\mathrm{DET}\cdot\mathrm{IS}\cdot\mathrm{DET})^{*}\cdot\mathit{any})^{*}\cdot\mathit{any})^{*}\cdot\mathit{any})^{*})^{*}\cdot\mathit{any})^{*}\cdot\mathit{any}\cdot(\mathit{any}\cdot\mathrm{NN}\cdot\mathit{any})^{*}\cdot\mathit{any})^{*}\cdot\mathit{any})^{*}\cdot\mathrm{NN}\cdot\mathit{any})^{*}\cdot\mathrm{NN}^{2})^{*}\cdot\mathit{any})^{*}\cdot\mathrm{NN}\cdot\mathit{any})^{*}\cdot\mathrm{DET}\cdot\mathrm{IS}\cdot\mathrm{DET})^{*}\cdot\mathit{any}\cdot(\mathit{any}\cdot\mathrm{NN}\cdot(\mathit{any}\cdot\mathrm{NN}^{2}\cdot(\mathit{any}\cdot\mathrm{NN}\cdot\mathit{any})^{*}\cdot\mathrm{IS}\cdot\mathrm{DET}\cdot\mathrm{IS}\cdot\mathrm{DET})^{*}\cdot\mathit{any})^{*}\cdot\mathit{any})^{*}\cdot\mathit{any})^{*})^{*}\cdot\mathit{any})^{*}\cdot\mathit{any}\cdot(\mathit{any}\cdot\mathrm{NN}\cdot\mathit{any})^{*}\cdot\mathit{any})^{*}\cdot\mathit{any})^{*}\cdot\mathrm{NN}\cdot\mathit{any})^{*}\cdot\mathrm{NN}^{2})^{*}\cdot\mathit{any})^{*}\cdot\mathrm{NN}\cdot\mathit{any})^{*}\cdot\mathrm{IS}\cdot\mathrm{DET}\cdot\mathrm{NN}\cdot\mathrm{DET}\cdot\mathrm{NN}\cdot\mathit{any})^{*}\cdot\mathrm{NN}^{2})^{*}\cdot\mathit{any})^{*}\cdot\mathrm{NN}\cdot\mathit{any})^{*}\cdot\mathrm{IS}\cdot\mathrm{DET}\cdot\mathrm{NN}$

One could easily discard from what is in the first bracket after NN to the last closing bracket star — )∗, since the star means zero or more, we can take the brackets to be a zero occurrence and we are left with the already learnt rule *noun is determiner noun.* However, we can safely assume that since this rule achieved a 3-point higher f-measure than the other experiments (which

[1]Only one sentence of the definitional sentences matches the last rule.

learnt the well-known sequence *noun is determiner noun*), it probably has done so by matching something within the brackets that helped to identify definitions slightly better.

**ExpM9** learnt the rule:

- $\text{NN} \cdot \text{IS} \cdot \text{DET} \cdot ((\mathit{any} \cdot ((\text{IS} \cdot \text{DET} \cdot \text{IS} \cdot \text{DET}) + (\text{IS} \cdot \text{NN})? + (\mathit{any} \cdot \text{IS}?)? + (\text{NN} \cdot \text{IS} \cdot \text{DET} \cdot \text{NN}) + (\text{NN} \cdot (\text{DET}? + (\mathit{any} \cdot \text{IS} \cdot ((\text{WHCH}? \cdot \text{MD}) + (\text{NN} \cdot \text{IS} \cdot \text{DET} \cdot \text{NN}))? \cdot \text{NN}))? \cdot \text{NN})) \cdot \text{DET}? \cdot \text{NN})? + \text{NN})? \cdot \text{NN}$

where again what is in brackets could be ignored, leaving us with the sequence *noun is determiner noun*. However, this rule also obtains a slightly higher f-measure (0.26) than other rules that simply learnt this sequence, so it is possible that inside the brackets lies a configuration or sequence of linguistic objects which capture more definitions.

Overall, through our observations of the experiments in the is-a category, we note that the most obvious rule, even to our machine learning technique, is the sequence *noun is determiner noun*. With just that rule as the basis of definition extraction it is possible to achieve an f-measure of 25%. The average recall stood around 33% and precision around 20%. In experiment ExpM4, where we had an f-measure of 28%, recall stood at 40% and precision at 22%. Although the comparison might not be well-levelled, manually crafted rules used within the LT4eL project in the is-a category obtained an f-measure of 26%, recall of 58% and precision of 17%. The manually crafted rules contained more detail than the sequence *noun is determiner noun* by trying to identify the general structure of the sentence apart from this phrase. However, based on these results, we are led to think that the manually crafted rules, having been specified over particular sentences, might have a negative impact over the process of definition extraction as a whole. This is based on the fact that the manual rules are capturing a higher number of definitions (higher recall), but also capturing a higher quantity of non-definitions (lower precision).

### 7.2.2 GP Results for the Verb Category

The verb category places together definitions that are based on certain cue phrases such as “is defined as”, “is known as”, “is referred to” and so on. Again, our approach is to split the GP experiment into two different sets. In the first set we focus on different parts-of-speech, with the hope of pinning down the types of verbs used in such sentences rather than using a generic verb category. In the second set of experiments we include the use of particular words, such as “called”, “known” and others.

The first set of experiments are split in two sections. In the first one, the GP is set up to include all different part-of-speech tags for the verb category shown below, whilst the second only contains the generic verb category to catch all below as one tag. The aim of this distinction is to see if there is any particular part-of-speech tags within the verb category that are more important to definitions than others. Apart from the usual set of linguistic objects (noun, adverb, adjective, modal, determiner, foreign word, preposition, to), we also have specific categories of verbs as follows:

- VBD — verb, past tense
- VBG — verb, gerund/present participle
- VBN — verb, past participle
- VBP — verb, non-3rd person singular present
- VBZ — verb, 3rd person singular present
- VB — verb

These categories are specified according to the Penn Treebank (Marcus *et al.* (1993)) part-of-speech tagset used when the linguistic objects were annotated. Table 7.3 shows the results for the two experiments, with VExp1 and VExp2 having all verb parts-of-speech, and VExp3, VExp4 having only the generic verb category specified as a catch-all tag. When one looks at the rules learnt by the best individuals in these runs, only one rule included the tag ‘VBP’ learnt by VExp2, with the rest of the rules leaving out the verb component entirely. It thus

Table 7.3: Results for GP in the verb category — first set

| Experiment | F-measure | Precision | Recall | Converged |
|---|---|---|---|---|
| VExp1 | 0.08 | 0.07 | 0.09 | 100 |
| VExp2 | 0.11 | 0.10 | 0.12 | 100 |
| VExp3 | 0.11 | 0.16 | 0.09 | 100 |
| VExp4 | 0.12 | 0.15 | 0.09 | 100 |

seems to be by coincidence that there is a slight increase in f-measure by using the general verb linguistic object rather than all parts-of-speech. Since the rules did not learn anything specific related to 'verbs' within a definitional sentences, it explains the poor results achieved in this category, and that in retrospect, one would expect a large amount of non-definitional sentences to contain these features as well, and thus it would be difficult for the GP to identify what type of verbs are distinguishing in definitions when such characteristic are present in an equally strong manner in non-definitions.

In the second experiment we augmented the linguistic objects by including the following set of words:

- mean
- define
- relate
- call
- consist
- know

This set of linguistic objects introduces the concept of the typical *verb* words found in such definitional sentences. The set above is primarily based on the words used in the manually crafted rules, since the latter was based on human

Table 7.4: Results for GP in the verb category — second set

| Experiment | F-measure | Precision | Recall | Converged |
|---|---|---|---|---|
| VExp5 | 0.19 | 0.13 | 0.33 | 40 |
| VExp6 | 0.20 | 0.33 | 0.14 | 100 |
| VExp7 | 0.21 | 0.19 | 0.23 | 100 |

observation. It is by no means complete, and could certainly be improved in further experiments.

The results presented in table 7.4 show that by introducing such words, it is now possible for the GP to actually learn rules which distinguish better definitional sentences from those learnt in the first set of experiments in this category.

In this set of experiments, the rules learnt included the words 'call', 'mean' and 'define'. Other words seem as though they were not properly explored. For instance, in Exp7, the word 'relate' was included in the initial population, but seems to have been discarded shortly afterwards. As we mentioned previously, search space exploration can be affected by how the initial population is drafted. If the rule containing 'relate' did not perform well because it was in an ungrammatical rule or was not present in the manually annotated definitions as an example, the token is easily lost through evolution. Notwithstanding, a notable improvement was registered by adding cue words that are generally used in this category.

### 7.2.3 GP Results for the Punctuation Category

The final category to which we applied the GP is the punctuation category, where the interesting aspect of sentences in this class of definitions is that they contain certain punctuation marks which are normally found only in such definitions. For instance the presence of a term followed by a colon would usually indicate that what follows is the definition of that term. In this experiment we include punctuation marks to our set of linguistic objects to try and learn rules that would specify where those marks fall in a sentence in order to distinguish a definition from a non-definition. For the purpose of this experiment we ran the GP with the

Table 7.5: Results for GP in the punctuation category

| **Experiment** | **F-measure** | **Precision** | **Recall** | **Converged** |
|---|---|---|---|---|
| PExp1 | 0.27 | 0.23 | 0.33 | 90 |
| PExp2 | 0.30 | 0.25 | 0.36 | 200 |

normal set of linguistic objects (noun, verb, adjective, adverb, modal, determiner, foreign word, preposition, which and to), and included the punctuation marks:

- colon
- semi-colon
- comma

We could have also included open and close brackets, however in our set of examples we did not have such sentences and thus the GP would not have learnt rules for such cases. In table 7.5 we present the f-measure, precision and recall for the experiments ran, and immediately one notices that in this category the GP achieves the best results. This could be due to two possible reasons: (a) the number of identifying features to be learnt are rather small and so it is could have resulted that the GP had an easier task to learn, and (b) it could also indicate that the sentences in this category do not vary much in their structure, and thus made the learning task easier.

Both experiments learnt rather simple and straightforward rules, each learning a conjunction of small rules. Interestingly enough, none of the rules contain the comma element, and seem to focus only on the use of the colon. For instance in the first experiment, PExp1, the best individual learnt that a sentence must contain:

- a preposition,
- a noun followed by a colon,
- a noun followed by zero or one colon followed by a noun,

- a noun followed by a colon followed by a determiner,
- a determiner followed by a noun,
- zero or one determiner,

in order to be considered as a definition. Unlike the other experiments, the rules learnt by the GP are very easy to understand and could be even translated manually to *lxtransduce* rules. The simplicity of the rules could indeed reflect that the learning task itself was a simple one and that sentences were similar to each other. However, the fact that the comma was not used in to rules could indicate that the search space was not explored enough, since after all we are only capturing one third of the definitions.

### 7.2.4 Learning Rules for All Definitions

A final experiment aims at running the algorithm on the whole set of definitions without the categories. This would mean a larger search space in terms of what needs to be learnt, and would probably result in less rules being learnt. However, it is of interest to try such an experiment to see how effective a GP could be without a categorised training set. The idea is to see what type of rules in such a generic setting the GP could learn. It might also be the case that the GP is able to identify rules for other categories in which we did not train the GP previously, especially in the case of the generic 'other' category.

Again we take two approaches to the experiments. The first set (AExp1, AExp2) includes a mixture of all generic linguistic objects used above, including the words under the verb category ('mean', 'known', 'define', 'relate', 'call', 'consist') and the punctuation marks (comma and colon), using a total of 19 linguistic objects. The second set (AExp3, AExp4) augmented the first set to include all parts-of-speech specified in the the Penn Treebank tagset, leaving the noun and verb categories in a general format, reaching a total of 29 linguistic objects. Table 7.6 shows the f-measure, precision and recall for these four experiments, all of which obtained very similar results.

The rules learnt by these experiments centred around the sequence "is a", and "is called a", with no other rules from different categories learnt. This reinforces

Table 7.6: Results for GP without definition categorisation

| Experiment | F-measure | Precision | Recall | Converged |
|---|---|---|---|---|
| AExp1 | 0.20 | 0.18 | 0.21 | 150 |
| AExp2 | 0.22 | 0.18 | 0.27 | 140 |
| AExp3 | 0.20 | 0.18 | 0.22 | 60 |
| AExp2 | 0.20 | 0.19 | 0.22 | 140 |

our statement that by increasing the search space it becomes more difficult for the GP to explore different rules which cover a wider area of the search space. The positive outcome from this experiment is that the two sets obtained very similar results, which implies that if we had to apply the GP to a different language, it is possible to start off by learning the most common rules in that language using the part-of-speech tagset as the set of linguistic objects.

## 7.3 Conclusion

In this chapter we describe the setup of the GP experiment, the way the individual is encoded and how it is evaluated during the experiment, and the different sets of linguistic objects used. We presented the results based on the f-measure metric obtained by the rules derived from our GP experiments for the different categories. The best results registered were in the punctuation category, where the GP obtained an f-measure of 30%, an improvement over manually crafted rules drafted in the LT4eL project. In the is-a category we obtain an average of around 25%, with the best run reaching 28%. In this category we observe that overall, the easiest rule learnt was *noun is a noun* and that rules reaching a higher f-measure included some additional information, such as the presence of a modal within the sentence. The verb category proved the most difficult category to learn due to the large search space for this particular problem. We observe that although certain cue words would be present in the initial randomly generated rules, these would be lost within a few evolutions since they are contained in rules which fair badly. It is also probable that certain words are not highly represented in our corpus, making it harder for the GP to learn rules based on a few sentences. In a final experiment we tried to learn rules over all categories, and as expected, the search space was too large and in fact only two rules were explored by the GP.

Overall the results are promising, with the GP achieving similar or better results than the manually crafted grammar in the LT4eL project. The GP manages to learn common rules automatically. At times, the rules learnt by the GP are long and complex to decipher, but in the best cases, the rules were rather simple and easy to understand once duplication is removed. The experiment also provides a case study for the use of GPs in Natural Language Processing, a combination which is hardly ever explored.

# Chapter 8

# Evolutionary Algorithms for Definition Extraction

The two experiments carried out have achieved positive results in isolation, as stand-alone algorithms. We set out to solve two problems in an independent manner. We applied a GP to learn rules which could be used to identify definitions, and a GA to learn weights to a fixed set of rules or features. In this chapter we discuss how it is possible to combine the two experiments and achieve a complete definition extraction tool which covers rule discovery and ranks the sentences classified as definitions by a certain level of confidence. Through the use of evolutionary algorithms it is possible to automate completely the task of definition extraction, with the added advantage of having also a ranking mechanism for the resulting classification of sentences. Through this application, we are able to increase the f-measure average of 25% in the is-a category by the GP (described in the previous chapter) up to 68%. This shows the validity of such algorithms in the problem of definition extraction.

## 8.1 Combining the Experiments for Definition Extraction

Our experiments have so far been isolated, with the purpose of solving two separate problems within definition extraction task. As specified in chapter 1, definition extraction poses two challenges to the task itself, (i) the discovery of rules which could be used to extract definitions, and (ii) the ranking of sentences proposed as definitions so that users are presented with definitions in a likelihood order. We have tackled both issues separately, the first using a GP to learn rules which could capture definitions, and the second using a GA to give weights which indicate a level of importance to a fixed set of rules. However, we can also form a 'pipeline' architecture and use the whole system as a complete definition extraction tool which would include rule-learning for a particular category, and a way of learning how to rank classified sentences.

As we described in the previous chapter, the GP is able to learn the most common rules that are able to extract a portion of the definitions annotated in a corpus. Using these rules, we would be able to have a definition extraction tool that simply classifies sentences as definitions or non-definitions. The GP has no knowledge of which sentences are more likely to be definitions, nor does it have any level of confidence attached to the rules. As we saw, each GP experiment over the same category would produce different rules, and although they would share several similarities between them, each GP can produce an individual which focuses on a particular aspect of the problem. For instance, in the verb category, one GP run focused on the cue word 'call', whilst another included rules with the word 'mean'. This means that running the GP several times could provide a larger coverage of the search space as each run could provide different rules (individuals). It is thus ideal to combine the best individuals from different GP runs in the definition extraction task. Each individual can be seen as one feature which captures a subset of the definitions present in a corpus.

It is now possible to apply the GA to this newly-formed set of features, and learn their respective weights. As outlined in chapter 6, these weights will indicate which rules are more likely to classify correctly definitions, and as a result, rank the candidate definitional sentences themselves. By providing this mechanism, we are able to present definitions to the user according to a level of confidence and make the end-purpose of the definition extraction task more usable.

For the purpose of this experiment, we use the best individuals from the different GP runs under the is-a category, described in section 7.2.1. In particular, we refer to the experiment results presented in table 7.2, where we presented

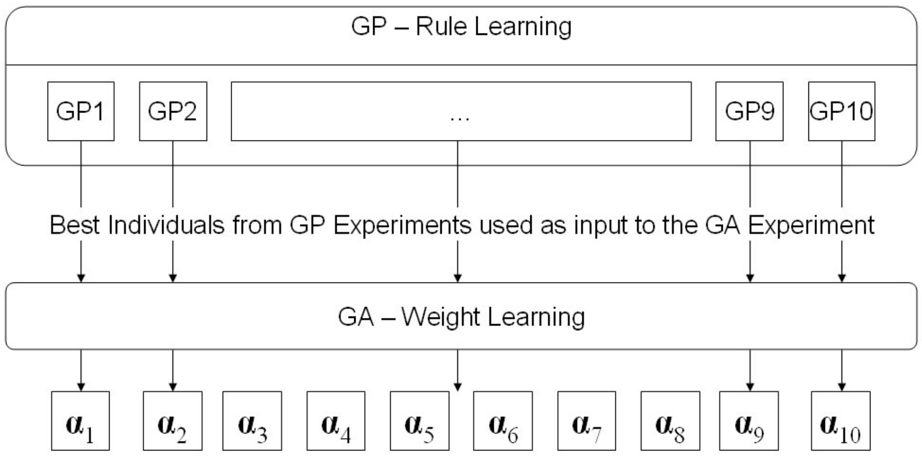

Figure 8.1: Combining the two experiments

the results for experiments entitled ExpM1 to ExpM10. The GP for these ten experiments had exactly the same settings for each run, with the best individual's f-measure ranging from 21% to 28%. In our analysis of the rules learnt by the different individuals, we noted that in certain cases the best individual resulted in the simple rule *noun is a noun*, whilst in other cases more complex rules were identified, with the inclusion of linguistic objects such as modal or other different linguistic sequences.

A visual representation of this experiment is presented in figure 8.1, where the output of the various GP runs (in our case, ten runs), becomes the input to the GA as a fixed set of features. The resulting output of the GA is a set of weights, with each weight being associated to the respective rule. For this experiment we used the best GA configuration identified in chapter 6, using `CountShifted` as our fitness function, and Elite with SUS selection technique. The GA was allowed to run for a maximum of 400 generations, with a population size of 100 individuals.

## 8.2 Results and Evaluation

The average f-measure by these set of rules learnt by the GP was at 25%, with various similarities shared within the rules themselves. Using this set of rules in the GA, we manage to achieve an f-measure of 68% by applying the weights learnt to the respective rules in the classification process. The following describes in better detail the figures behind our f-measure metric:

**Precision** 100% — The sentences classified as definitions have been classified correctly.

**Recall** 51% — Out of all the definitions manually marked, we manage to classify 51% of them.

This means that together with the rules learnt and their weights, we successfully manage to classify just over half of the definitions, without classifying any incorrect definitions. If we compare this result to the evaluation of the classification of sentences in the GA, specifically in section 6.3.5, it is possible to gain a better understanding of why this result was achieved. In that section we explained that through the evaluation of the classified sentences, we discovered some anomalies in the manual tagging of the corpus. There were cases where definitional sentences were left out from the manual marking, and other cases where non-definitions were inadvertently marked as definitions. In the analysis of this classification, we found that some of the non-definitional sentences classified as definitions were similar in structure to definitional sentences. In fact, it was observed that the sentences achieved similar scores, and in reality should have been annotated as definitions.

The identification of this problem is of extreme importance when we analyse what the GP learnt, and how it might have derived its individuals. We know that the corpus used in this experiment has incorrect classifications of some of the sentences. Thus we have a situation where sentences sharing similar definitional structures have not been marked in a consistent manner. As an example, the following two sentences are taken from our corpus, where the first sentence $S_1$, was not manually marked as a definition, whilst the second one was.

$S_1$ — **not marked as a definition:** Pro/DESKTOP is a commercial computer-aided design (CAD) software package that enables users to design using 3D solid models, and then go on to produce engineering drawings and photo-realistic renderings.

$S_2$ — **marked as a definition:** A daemon is a program like a print spooler, a mail listener or a WWW server that lurks in the background, waiting for things to do.

In its learning process, the GP learnt rules that covered both sentences, meaning that its f-measure was also being affected by capturing *negative* examples. The GA surprisingly managed to learn weights in such a way as to distinguish completely between the positive and negative sentences, and managed to classify $S_1$ as a non-definition, according to the training set. In fact the GA did not manage to achieve a higher recall only because the collective GP rules did not cover any of the remaining definitions.

## 8.3 Conclusion

The result of this experiment shows the success evolutionary algorithms can have in the classification of definitional sentences based on linguistic knowledge. The experiment shows that by applying a GA to a set of rules learnt by the GP, it is possible to obtain a perfect distinction between definitions and non-definitions. We have also showed that it is possible to achieve a complete automatic process from rule-learning to definition ranking without requiring specialist knowledge. In our initial requirements we set out the objective to classify definitions without including too many non-definitions. This experiment has superseded that objective by classifying only definitions for all those sentences that GP learnt rules for, even where the GP rules also covered non-definitions.

# Chapter 9

# Conclusions and Future Work

In this chapter we review the work carried out and discuss the results achieved and the level of success of our experiment. We look at each experiment separately, and discuss what was achieved and what further possibilities are present for the respective experiments. Finally, we conclude by giving a direction for further experiments and work for definition extraction in general, not limited to the techniques used in this thesis.

## 9.1 Genetic Algorithms in Definition Extraction

The GA experiment focuses on learning weights to a set of features which could identify definitions. The weights would then act as a ranking mechanism in the classification of sentences, providing a level of certainty as to whether a sentence is actually a definition or a non-definition. Through the experiments carried out, we found that zero is not an ideal classifying value, and that it should be optimised to the particular set of features for which we are learning weights. Through this process we successfully managed to obtain a precision of 62% in the is-a category by using a set of simple features such as *has keyword*, *contains "is a"*, and so on. We also saw that it is possible to influence the learning of weights to favour either precision or recall by changing the kappa value in the f-measure metric. The kappa value reflects the relative importance of precision and recall which is used by the fitness function to incline towards one of these values so as to emphasis the context and purpose of our learning algorithm.

An important outcome of the evaluation process was the observation of incorrect manual annotation of definitions. This brings forward the idea of using the GA as a way of assisting in the manual annotation of definitions in a corpus. By identifying simple key features present in some of the definitional sentences, one could use the GA to present similar sentences as candidate definitions which might be marked as definitions. The GA could also be used to correct irregularities found in the corpus, simply by carrying out an evaluation which looks at the top percentages of the incorrectly classified sentences. Through such a process it would then be possible to rectify any errors in the annotation. It is certainly the case that when annotating definitions manually, the annotator is seeing sentences in context, whereby some sentences might appear as definitions. But when these are seen outside of their context, as sole sentences, they might be vague or unclear and thus lack the necessary properties to be considered as complete definitions. The use of the classification by the GA allows an annotator to see these sentences in such a manner, and thus be able to perform a better judgement on the quality of the sentence as a definition. Through the ranking mechanism, one would also see the sentence next to other likely candidates, and can compare the qualitative value of a sentence compared to other similarly ranked sentences.

Our GA experiments were limited to the is-a category. Further experimentation would be required to see the effectiveness of weights learnt by a GA on other definitional categories. In the case of the verb category, a set of features could easily include a set of cue phrases used by both the LT4eL project and other work reviewed such as Liu *et al.* (2003); Malaisé *et al.* (2004); Muresan & Klavans (2002); Storrer & Wellinghoff (2006). The GA could learn weights to the respective phrases and identify which ones are more likely to be found in definitions. Similarly, in the punctuation category we could apply similar features to those identified by Przepiórkowski *et al.* (2007) such as `NounPhrase: NounPhrase`, or simply using different punctuation mark sequences (e.g. comma noun-phrase comma). We would expect similar results in the verb and punctuation categories if we apply the GA to learn weights to the rules learnt by the GP. During this process one could also consider re-examining the other categories to analyze their classification. For instance, there were sentences in the anaphora category that were classified as definitions under the is-a category. Thus, it is possible that definitions share features over the different categories.

Another interesting experiment would be to put all the categories together and their respective learnt features to try and learn weights for a single definition extraction tool. After all, the definition extractor will be used not on sole categories but rather as a full tool covering all types of categories.

## 9.2 Genetic Programming in Definition Extraction

The GP experiment was designed to learn rules which could identify definitions. From the different experiments carried out we observe that the size of the search space influences the ability of the GP to learn generic rules. We also saw how the coverage of the search space by the rules learnt depends on the initial random generation of the population and on mutation, since linguistic objects can only be introduced at these times, and risk being lost through the evolution process at the early stages of the run if they are present in ungrammatical rules. If a particular linguistic object is discarded, the search space cannot be fully explored and thus the resulting rules from that GP will not contain any rules representing that solution. In fact we observed that in general, the GP learnt rules for the most common occurrences present in definitions. The lack of exploration of the search space was noticed most in the verb category experiments, with most rules emphasising on the presence of the word ‘call’. Other words were rarely considered in the rules learnt, and thus achieving the lowest results in our experiments. The punctuation category performed very well, achieving a considerable improvement in f-measure over the manually crafted rules in the LT4eL grammars. This improvement is probably due to the restriction over the search space, where we are only considering sentences which have certain punctuation mark occurrences.

At times, the rules learnt by the GP were long and complex, containing several duplicate parts (e.g. at the top level, we would find a conjunction of two identical rules). The growth of the GP trees was controlled only during the random creation of individuals in the initial population and of subtrees during mutation. However, during the crossover function we did not consider the depth of the tree, and allowed crossover to be carried out irrespective of the depth of the node on which crossover was to be done. In the worst case scenario, the depth of the tree would have doubled at each generation. Certain conditions could be included in future to limit the depth of a tree. One operation which might reduce a tree's depth is to remove duplicate rules from the trees, say, after one complete generation. Such rules add only computational requirements to the GP and contain no useful information. Sometimes the rules learnt are also a subset of each other. Of course, one would have to weigh the computational requirements of removing such extra information during each generation of the GP.

An interesting experiment to be carried out in future would be to use a GP to learn rules in an iterative manner. The GP would start with a full training set but after the first experiment, the sentences for which it learnt rules are removed from the training corpus, and a second experiment is carried out. At each stage, we would remove sentences for which we learnt the rules, leaving the remaining sentences as the training corpus. In this way we would be reducing the search space, and forcing the GP to learn new rules (having removed the more common sentences). It might be the case that the GP does not learn certain rules as they would classify to many non-definitions to simply capture few definitions. However, by carrying out such an experiment we might be able to learn rules which cover the search space better, and at the same time identify those definitions for which it is difficult to define rules which provide acceptable results.

The GP was a successful experiment, managing to learn similar rules to the manually crafted rules by the human expert in the LT4eL project. Categorisation of definitions did help the GP to identify more specific rules, rather than attempting rule-learning over all definitions.

## 9.3 Future Work in Definition Extraction

The final experiment of using both the GP and the GA together as one tool for definition extraction gave surprising results, managing to identify only definitions, achieving a 100% precision, albeit having identified rules to capture only half of the definitional set of sentences. This result is certainly encouraging when considering that the process is fully automated and also provides a ranking mechanism to the classification of sentences. In the work reviewed, Fahmi & Bouma (2006) manage to obtain a maximum of 92% accuracy when extracting definitions from the Dutch Wikipedia medical articles using maximum entropy. In this work, they use information we do not have access to, such as sentence position, since a definition in such an article is typically found in the beginning. Recall is not calculated since the definitions are not annotated. Blair-Goldensohn *et al.* (2004) manage to obtain 96% precision (again recall is not given), where the extracted definitions are used for a question answering system which summarised the information to provide a long format answer.

Our experiments would need to be evaluated further, initially by adjusting the incorrect annotations discovered during the evaluation process. Such an adjustment would have a positive outcome on our results, enabling the GP to learn better classifying rules by providing stronger reinforcement of certain type of sentences. We would also need to experiment with other corpora in different domains. For instance, medical texts contain several terms which a part-of-speech tagger might not recognise and would tag as foreign word (FW). Thus the rules learnt for our eLearning corpus might not necessarily apply for a medical corpus. We intend to use Henry Gray's Anatomy of the Human Body (Gray, 1918) — a popular textbook which is still used in today's medical undergraduate courses.

Another interesting experiment that could be carried out in future would be to compare the weights learnt by the GA to other machine learning applications such as Hidden Markov Models, Support Vector Machines, and other classification algorithms found in the Weka toolset (Witten & Frank, 2005). Another possibility would be to compare the results of a simple probabilistic n-gram model to what is learnt by the GP, and to see whether applying the model to the GA to learn their weights would enhance its performance. This might not be necessarily the case, since the probabilistic model is already learning the more relevant features.

Our machine learning approach is completely language independent, and thus further experiments would attempt at learning rules over corpora in different languages. Our first experiments will be carried out in languages present in the LT4eL project for which resources are readily available, and definitions are annotated. The only knowledge required to set the linguistic objects for the GP would be a slight understanding of the part-of-speech tagger (e.g. understanding which tags represent different types of nouns). By using the GP in an iterative process as suggested above, it might be possible to identify the important tags which are present in definitions.

One aspect missing in this thesis is an evaluation of the definition extraction tool over an unseen corpus. For this purpose we plan to evaluate our techniques on more technical texts, specifically in the domain of eLearning through the MIT Open Courseware[1] and Open University's Learning Space[2]. Such an evaluation might show that the rules learnt by the GP are not generic enough to cover unseen definitions, a result which is common in such machine learning techniques. It would be ideal to have some form of feedback loop from an expert to the learning algorithm to integrate new knowledge gained over unseen copora.

## 9.4 Conclusions

Overall, the experiments carried out have resulted in very interesting results for definition extraction. The final experiment of using both the GA and GP as a single tool for definition extraction brought out a surprising and positive results that the GA was able to learn weights which achieved a 100% precision, a feat that was not achieved in any of the reviewed work. Of course, such an application would need to be tested on an unseen corpus to evaluate the quality and applicability of the results achieved in this work. Further work needs to be carried out to integrate the two algorithms into one seamless application. Based on the results achieved it is worth exploring the possibility of including this tool not only as part of an eLearning system, but even in other domains such as that of question answering. Further experiments should consolidate the results achieved and further encourage experimentation in natural language processing using evolutionary algorithms.

[1]`http://ocw.mit.edu/OcwWeb/web/home/home/index.htm`
[2]`http://openlearn.open.ac.uk/`

# Appendix A

# Rules Learnt by The Genetic Program

This appendix presents the rules learnt by the GP during the experiments carried out. These rules have been simplified from the original rules by removing obvious redundant instances such as duplicates to make them more legible and understandable.

## A.1 Is-a Category

### A.1.1 GP Experiments with a Small Set of Features

1. Best individual for Exp1; Satisfy all the following:

   - $\mathrm{NN} \cdot \mathrm{IS} \cdot \mathit{any}$
   - $\mathrm{IS} \cdot \mathit{any} \cdot \mathrm{NN} \cdot \mathit{any}^2$
   - $\mathit{any}^4 \cdot \mathrm{NN}$
   - $\mathrm{IS} \cdot \mathit{any} \cdot \mathrm{NN} \cdot \mathit{any}^2$
   - $\mathrm{NN} \cdot \mathrm{IS} \cdot \mathit{any} \cdot \mathrm{NN}$

2. Best individual for Exp2b; Satisfy all the following:

   - $\mathrm{NN} \cdot \mathit{any?} \cdot \mathrm{IS} \cdot (\mathit{any} + (\mathrm{NN} \cdot \mathrm{IS} \cdot \mathrm{NN})) \cdot \mathrm{NN}$
   - $\mathrm{NN} \cdot \mathit{any} \cdot (\mathit{any} + (\mathrm{NN} \cdot \mathit{any?} \cdot \mathit{any}^2)) \cdot \mathrm{NN}$

- $\mathrm{NN} \cdot \mathrm{IS}? \cdot \mathit{any} \cdot (\mathit{any} + (\mathrm{NN} \cdot \mathrm{IS} \cdot \mathit{any} \cdot \mathrm{IS} \cdot (\mathit{any}? + (\mathit{any}? \cdot \mathrm{IS})) \cdot \mathrm{NN} \cdot (\mathit{any} + \mathit{any}? + (\mathit{any}? \cdot \mathrm{IS})) \cdot \mathit{any}))$
- $\mathrm{IS} \cdot \mathit{any}? \cdot (\mathit{any} + (\mathrm{NN} \cdot \mathit{any}? \cdot \mathit{any} \cdot (\mathit{any} + (\mathrm{NN} \cdot \mathit{any} \cdot (\mathit{any} + \mathit{any}? + (\mathit{any}? \cdot \mathrm{IS})) \cdot \mathit{any})))) \cdot (\mathit{any} + (\mathrm{IS} \cdot \mathit{any} \cdot \mathrm{IS} \cdot (\mathit{any}? + (\mathit{any}? \cdot \mathrm{IS})))) \cdot \mathrm{NN}$

3. Best individual for Exp3; Satisfy all the following:

- $(\mathrm{IS} \cdot \mathit{any})?$
- $\mathit{any}^3 \cdot \mathrm{AJ} \cdot \mathit{any}^2 \cdot \mathrm{NN} \cdot \mathit{any}$
- $\mathit{any}^2 \cdot \mathrm{NN}$
- $\mathit{any}^4 \cdot \mathrm{AJ} \cdot \mathit{any}^2 \cdot \mathrm{NN} \cdot \mathit{any}$
- $\mathrm{NN} \cdot \mathit{any}^3 \cdot \mathrm{NN}$
- $\mathit{any}^4 \cdot \mathrm{AJ} \cdot \mathit{any}^3$
- $\mathrm{NN} \cdot \mathit{any} \cdot \mathrm{NN}$
- $\mathit{any}^5 \cdot \mathrm{NN}$
- $\mathrm{AJ} \cdot \mathit{any}^2 \cdot \mathrm{NN}$
- $\mathrm{IS} \cdot \mathit{any} \cdot \mathrm{NN}$

4. Best individual for Exp4; Satisfy all the following:

- $\mathrm{NN} \cdot (((((\mathit{any} \cdot (\mathrm{AJ} + ((((\mathrm{AJ} + \mathrm{NN}) \cdot (\mathit{any} \cdot \mathrm{MD})?) + (\mathrm{NN} \cdot (\mathrm{NN} \cdot \mathit{any})? \cdot \mathrm{NN} \cdot \mathit{any}^2 \cdot \mathrm{MD}) + \mathit{any})? \cdot \mathrm{NN} \cdot \mathit{any}) + (\mathrm{AJ} \cdot (\mathit{any} \cdot \mathrm{MD})?)) \cdot \mathit{any}^3)? \cdot \mathrm{NN} \cdot \mathit{any}) + \mathit{any}) \cdot \mathrm{NN} \cdot \mathit{any})? \cdot (\mathrm{AJ}? + ((\mathit{any} \cdot \mathrm{MD})? \cdot \mathrm{AJ} \cdot ((\mathit{any} \cdot (\mathrm{AJ} + \mathrm{MD})) + (\mathrm{NN} \cdot \mathit{any} \cdot \mathrm{MD}))) + \mathrm{AJ} + ((\mathit{any} \cdot (\mathrm{AJ} + (((\mathrm{AJ}? \cdot (((\mathit{any} \cdot \mathrm{MD})? \cdot (\mathit{any} + (\mathrm{NN} \cdot \mathit{any}))) + \mathit{any})) + (\mathrm{NN}^2 \cdot \mathit{any} \cdot \mathrm{MD}) + \mathit{any})? \cdot \mathit{any}^3) + (\mathrm{NN} \cdot \mathrm{AJ} \cdot \mathrm{MD}?)) \cdot \mathit{any}^3)? \cdot \mathrm{NN} \cdot \mathit{any})))? \cdot \mathrm{IS} \cdot ((\mathrm{NN} \cdot ((\mathrm{AJ} + (\mathit{any}? \cdot \mathrm{NN} \cdot \mathit{any}) + ((\mathit{any} \cdot \mathrm{MD})? \cdot \mathrm{AJ} \cdot ((\mathrm{AJ} + \mathrm{IS})^2 + (\mathit{any}^2? \cdot \mathrm{NN} \cdot \mathit{any}))) + ((\mathit{any} \cdot ((\mathit{any} \cdot (\mathrm{AJ} + \mathrm{IS})) + (\mathit{any}^2? \cdot \mathrm{NN} \cdot \mathit{any})))? \cdot \mathrm{NN} \cdot \mathit{any}))? \cdot (((\mathrm{IS} + (\mathrm{NN} \cdot \mathit{any}))? \cdot (\mathrm{AJ} + (\mathrm{NN} \cdot ((\mathit{any}? \cdot \mathrm{NN} \cdot \mathit{any}^2)? \cdot (\mathrm{AJ}? + ((\mathrm{AJ} + (\mathrm{NN} \cdot ((\mathit{any}? \cdot \mathrm{NN} \cdot \mathit{any}^2)? \cdot (\mathrm{AJ}? + ((\mathit{any} \cdot \mathrm{MD})? \cdot \mathrm{AJ} \cdot \mathrm{NN}) + \mathrm{AJ} + ((\mathit{any} \cdot (\mathrm{AJ} + (((\mathrm{AJ}? \cdot (\mathit{any} \cdot \mathrm{MD})?) + (\mathrm{NN} \cdot (\mathrm{NN} \cdot \mathit{any})? \cdot \mathrm{NN} \cdot \mathit{any}^2 \cdot \mathrm{MD}) + \mathit{any})? \cdot \mathrm{NN} \cdot \mathit{any}) + (\mathrm{NN} \cdot \mathrm{AJ} \cdot (\mathit{any} \cdot \mathrm{MD})?)) \cdot \mathit{any}^3)? \cdot \mathrm{NN} \cdot \mathit{any})))? \cdot \mathrm{IS} \cdot ((\mathrm{NN} \cdot ((\mathit{any} + ((\mathit{any} \cdot \mathrm{MD})? \cdot \mathrm{AJ} \cdot ((\mathrm{AJ} + \mathrm{IS})^2 + (\mathit{any}^2? \cdot \mathrm{NN} \cdot \mathit{any}))) + \mathrm{AJ} + ((\mathit{any} \cdot$

$((\mathit{any}\cdot((\mathit{any}?\cdot\mathrm{NN}\cdot\mathit{any})?+\mathrm{AJ}+\mathrm{NN}))+((\mathit{any}?\cdot\mathrm{NN}\cdot\mathit{any})?\cdot\mathrm{NN}\cdot\mathit{any})))?\cdot$
$\mathrm{NN}\cdot\mathit{any}))?\cdot((\mathrm{IS}+(\mathit{any}?\cdot\mathrm{NN}\cdot\mathit{any}))?+(\mathrm{NN}\cdot\mathit{any})))?)+\mathit{any})\cdot\mathrm{NN}\cdot\mathit{any}^2)+$
$((\mathit{any}\cdot\mathrm{MD})?\cdot\mathrm{AJ}\cdot(\mathrm{AJ}+\mathrm{IS})^2?))\cdot\mathrm{AJ}\cdot\mathrm{NN})+\mathrm{AJ}+((\mathit{any}\cdot(\mathrm{AJ}+(((\mathrm{AJ}?\cdot(\mathit{any}\cdot$
$\mathrm{MD})?)+(\mathit{any}\cdot((\mathit{any}?\cdot\mathrm{NN}\cdot\mathit{any})?+\mathrm{AJ}+\mathrm{NN}))+\mathit{any})?\cdot\mathrm{NN}\cdot\mathit{any})+(\mathrm{NN}\cdot\mathrm{AJ}\cdot$
$(\mathit{any}\cdot\mathrm{MD})?))\cdot\mathit{any}^3)?\cdot\mathrm{NN}\cdot\mathit{any})))?\cdot\mathrm{IS}\cdot((\mathrm{NN}\cdot((\mathrm{AJ}+(((\mathrm{AJ}?\cdot((\mathit{any}\cdot\mathrm{MD})?\cdot$
$\mathit{any}?\cdot\mathrm{AJ}\cdot\mathit{any}?)?)+(\mathrm{NN}\cdot\mathit{any}\cdot((\mathrm{AJ}+\mathrm{IS})^2+(\mathit{any}^2?\cdot\mathrm{NN}\cdot\mathit{any})))+\mathit{any})?\cdot$
$\mathrm{NN}\cdot\mathit{any})+((\mathit{any}\cdot\mathrm{MD})?\cdot\mathrm{AJ}\cdot((\mathrm{NN}\cdot((\mathrm{AJ}+(((\mathrm{AJ}?\cdot((\mathit{any}\cdot\mathrm{AJ}\cdot((\mathrm{AJ}+\mathrm{IS})^2+$
$(\mathit{any}^2?\cdot\mathrm{NN}\cdot\mathit{any})))?\cdot\mathit{any}?\cdot\mathrm{AJ}\cdot\mathit{any}?)?)+(\mathrm{NN}\cdot\mathit{any}\cdot((\mathrm{AJ}+\mathrm{IS})^2+(\mathit{any}^2?\cdot$
$\mathrm{NN}\cdot\mathit{any})))+\mathit{any})?\cdot\mathrm{NN}\cdot\mathit{any})+((\mathit{any}\cdot\mathrm{MD})?\cdot\mathrm{AJ}\cdot((\mathrm{AJ}+\mathrm{IS})^2+(\mathit{any}^2?\cdot$
$\mathrm{NN}\cdot\mathit{any})))+((\mathit{any}\cdot((\mathit{any}\cdot((\mathit{any}?\cdot\mathrm{NN}\cdot\mathit{any})?+\mathrm{AJ}+\mathrm{NN}))+((\mathit{any}?\cdot\mathrm{NN}\cdot$
$\mathit{any})?\cdot\mathrm{NN}\cdot\mathit{any})))?\cdot\mathrm{NN}\cdot\mathit{any}))?\cdot(((\mathrm{IS}+(\mathit{any}?\cdot\mathrm{NN}\cdot\mathit{any}))?\cdot(\mathrm{AJ}?+(((\mathit{any}\cdot$
$\mathrm{MD})+\mathit{any})\cdot\mathrm{AJ}\cdot(\mathrm{AJ}+\mathrm{IS})^2?)+\mathrm{AJ}+((\mathit{any}\cdot((\mathrm{AJ}?\cdot\mathit{any}^2)+(\mathrm{NN}\cdot\mathit{any}\cdot\mathrm{MD})+$
$\mathit{any})\cdot\mathrm{NN}\cdot\mathit{any}^2)?\cdot\mathrm{NN}\cdot\mathit{any})))+(\mathrm{NN}\cdot\mathit{any})))?)+\mathit{any}+(\mathit{any}^2?\cdot\mathrm{NN}\cdot\mathit{any})))+$
$((\mathit{any}\cdot((\mathit{any}\cdot(((\mathit{any}\cdot\mathrm{NN}\cdot(((\mathrm{AJ}+(((\mathrm{AJ}?\cdot((\mathit{any}\cdot\mathrm{MD})?\cdot\mathit{any}?\cdot\mathrm{AJ}\cdot\mathit{any}?)?)+$
$\mathrm{MD}+(((\mathrm{AJ}?\cdot((\mathit{any}\cdot\mathrm{MD})?\cdot\mathrm{AJ}\cdot((\mathrm{AJ}+\mathrm{IS})^2+(\mathit{any}^2?\cdot\mathrm{NN}\cdot\mathit{any})))?)+(\mathrm{NN}\cdot$
$\mathit{any}\cdot\mathrm{MD})+\mathit{any})?\cdot\mathrm{NN}\cdot(\mathrm{AJ}+\mathrm{IS}))+\mathit{any})?\cdot\mathrm{NN}\cdot\mathit{any})+((\mathit{any}\cdot\mathrm{MD})?\cdot\mathrm{AJ}\cdot$
$((\mathrm{AJ}+\mathrm{IS})^2+(\mathit{any}^2?\cdot\mathrm{NN}\cdot\mathit{any})))+((\mathit{any}\cdot(\mathit{any}^2+((\mathit{any}?\cdot\mathit{any}^2?\cdot\mathit{any})?\cdot$
$\mathrm{NN}\cdot\mathit{any})))?\cdot\mathrm{NN}\cdot\mathit{any})+\mathrm{NN})\cdot(((\mathrm{IS}+(\mathit{any}?\cdot\mathrm{NN}\cdot\mathit{any}))?\cdot(\mathrm{AJ}?+(((\mathit{any}\cdot$
$\mathrm{MD})+\mathit{any})\cdot\mathrm{AJ}\cdot(\mathrm{AJ}+\mathrm{IS})^2?)+\mathrm{AJ}+((\mathit{any}\cdot((\mathrm{AJ}?\cdot\mathit{any}^2)+(\mathrm{NN}\cdot\mathit{any}\cdot\mathrm{MD})+$
$\mathit{any})\cdot\mathrm{NN}\cdot\mathit{any}^2)?\cdot(\mathit{any}\cdot\mathrm{MD})?\cdot\mathit{any})))+((\mathrm{NN}\cdot\mathit{any})?\cdot\mathrm{NN}\cdot\mathit{any})))+\mathrm{AJ}+$
$(\mathrm{NN}\cdot((\mathit{any}?\cdot\mathrm{NN}\cdot\mathit{any})?\cdot(\mathrm{AJ}?+((\mathit{any}\cdot\mathrm{MD})?\cdot\mathrm{AJ}\cdot\mathrm{NN})+\mathrm{AJ}+((\mathit{any}\cdot(\mathrm{AJ}+$
$(((\mathrm{AJ}?\cdot(\mathit{any}\cdot\mathrm{MD})?)+(\mathrm{NN}\cdot(\mathrm{NN}\cdot\mathit{any})?\cdot\mathrm{NN}\cdot\mathit{any}^2\cdot\mathrm{MD})+(\mathit{any}\cdot\mathrm{IS}))?\cdot$
$\mathrm{NN}\cdot\mathit{any}))?\cdot\mathit{any}^3)?\cdot\mathrm{NN}\cdot\mathit{any})))?\cdot\mathrm{IS}\cdot((\mathrm{NN}\cdot((\mathrm{AJ}+(((\mathrm{AJ}?\cdot(((\mathit{any}\cdot\mathrm{MD})+$
$\mathit{any})\cdot\mathit{any}?\cdot\mathrm{AJ}\cdot\mathit{any}?)?)+\mathrm{MD}+(((\mathrm{AJ}?\cdot((\mathit{any}\cdot\mathrm{MD})?\cdot\mathrm{AJ}\cdot((\mathrm{AJ}+\mathrm{IS})^2+$
$(\mathit{any}^2?\cdot\mathrm{NN}\cdot\mathit{any})))?)+(\mathrm{NN}\cdot\mathit{any}\cdot\mathrm{MD})+\mathit{any})?\cdot\mathrm{NN}\cdot(\mathrm{AJ}+\mathrm{IS}))+\mathit{any})?\cdot$
$\mathrm{NN}\cdot\mathit{any})+((\mathit{any}\cdot\mathrm{MD})?\cdot\mathrm{AJ}\cdot((\mathrm{AJ}+\mathrm{IS})^2+(\mathit{any}^2?\cdot\mathrm{NN}\cdot\mathit{any})))+((\mathit{any}\cdot$
$((\mathit{any}\cdot((\mathit{any}?\cdot\mathrm{NN}\cdot\mathit{any})?+\mathrm{AJ}+\mathrm{NN}))+((\mathit{any}?\cdot\mathrm{NN}\cdot\mathit{any})?\cdot\mathrm{NN}\cdot\mathit{any})))?\cdot$
$\mathrm{NN}\cdot\mathit{any}))?\cdot(((\mathrm{IS}+(\mathit{any}?\cdot\mathrm{NN}\cdot\mathit{any}))?\cdot(\mathrm{AJ}?+(((\mathrm{NN}\cdot\mathrm{AJ}\cdot(\mathit{any}\cdot\mathrm{MD})?)+$
$\mathrm{AJ})\cdot\mathrm{AJ}\cdot(\mathrm{AJ}+\mathrm{IS})^2?)+\mathrm{AJ}+((\mathit{any}\cdot((\mathrm{AJ}?\cdot\mathit{any}^2)+(\mathrm{NN}\cdot\mathit{any}\cdot\mathrm{MD})+\mathit{any})\cdot$
$\mathrm{NN}\cdot\mathit{any}^2)?\cdot\mathrm{NN}\cdot\mathit{any})))+((\mathrm{NN}\cdot\mathit{any})?\cdot\mathrm{NN}\cdot\mathit{any})))?)+\mathit{any})\cdot\mathrm{NN}\cdot\mathit{any}^2)+$
$((\mathit{any}\cdot\mathrm{MD})?\cdot\mathrm{AJ}\cdot(\mathrm{AJ}+\mathrm{IS})^2?)+((\mathit{any}\cdot(\mathrm{MD}+((\mathit{any}+(\mathrm{AJ}?\cdot((\mathit{any}\cdot\mathrm{MD})?\cdot$
$\mathrm{AJ}\cdot((\mathrm{AJ}+\mathrm{IS})^2+(\mathit{any}^2?\cdot\mathrm{NN}\cdot\mathit{any})))?)+(\mathrm{NN}\cdot\mathit{any}\cdot\mathrm{MD}))\cdot\mathrm{NN}\cdot\mathit{any})+$
$(\mathrm{NN}\cdot\mathrm{AJ}\cdot(\mathit{any}\cdot\mathrm{MD})?)+\mathrm{AJ})\cdot\mathrm{NN}\cdot\mathit{any}^2)?\cdot\mathrm{NN}\cdot\mathit{any})))?\cdot\mathrm{NN}\cdot\mathit{any})?+\mathrm{AJ}+$

$\mathrm{NN}))+((\mathit{any}?\cdot\mathrm{NN}\cdot\mathit{any})?\cdot\mathrm{NN}\cdot\mathit{any})))?\cdot\mathrm{NN}\cdot\mathit{any})+\mathrm{NN})\cdot(((\mathrm{IS}+(\mathit{any}?\cdot\mathrm{NN}\cdot$
$\mathit{any}))?\cdot(\mathrm{AJ}?+(((\mathit{any}\cdot\mathrm{MD})+\mathit{any})\cdot\mathrm{AJ}\cdot(\mathrm{AJ}+\mathrm{IS})^2?)+\mathrm{AJ}+((\mathit{any}\cdot((\mathrm{AJ}?\cdot$
$\mathit{any}^2)+(\mathrm{NN}\cdot\mathit{any}\cdot\mathrm{MD})+\mathit{any})\cdot\mathrm{NN}\cdot\mathit{any}^2)?\cdot\mathrm{NN}\cdot\mathit{any})))+(\mathrm{NN}\cdot\mathit{any})))?)+$
$\mathit{any})\cdot\mathrm{NN}\cdot\mathit{any}^2)+((\mathit{any}\cdot\mathrm{MD})?\cdot\mathrm{AJ}\cdot(\mathrm{AJ}+\mathrm{IS})^2?)+((\mathit{any}\cdot(\mathrm{MD}+((\mathit{any}+$
$(\mathrm{AJ}?\cdot((\mathit{any}\cdot\mathrm{MD})?\cdot\mathrm{AJ}\cdot((\mathrm{AJ}+\mathrm{IS})^2+(\mathrm{NN}^2\cdot\mathit{any})))?)+(\mathrm{NN}\cdot\mathit{any}\cdot\mathrm{MD}))\cdot$
$\mathrm{NN}\cdot\mathit{any})+\mathit{any})\cdot\mathrm{NN}\cdot\mathit{any}^2)?\cdot\mathrm{NN}\cdot\mathit{any})))+(\mathrm{NN}\cdot\mathit{any})))?)+\mathit{any})\cdot\mathrm{NN}\cdot\mathit{any}^2$

5. Best individual for Exp5; Satisfy all the following:

   - $((\mathit{any}+(\mathit{any}^3\cdot\mathrm{NN}))\cdot((\mathit{any}\cdot\mathrm{NN})+(((\mathit{any}\cdot((\mathit{any}\cdot\mathrm{NN})+\mathit{any}))+\mathrm{NN}+$
     $\mathrm{IS})\cdot(((\mathit{any}^2\cdot\mathrm{NN})?\cdot((\mathit{any}\cdot\mathrm{NN})+(((\mathit{any}\cdot((\mathit{any}\cdot\mathrm{NN})+\mathit{any}))+\mathrm{NN}+$
     $(\mathrm{IS}\cdot\mathit{any}\cdot((\mathit{any}\cdot((((\mathit{any}\cdot(\mathit{any}^3+(\mathrm{IS}\cdot\mathrm{NN}\cdot\mathit{any}^2)))+\mathrm{NN}+\mathit{any}+(\mathrm{IS}\cdot$
     $\mathit{any}))\cdot(\mathit{any}^2+\mathrm{NN}))+(\mathrm{IS}\cdot((((\mathit{any}\cdot(\mathit{any}^4+\mathrm{NN}))+(\mathrm{IS}\cdot\mathit{any}))\cdot((\mathit{any}\cdot$
     $\mathrm{NN}\cdot((\mathit{any}\cdot\mathrm{NN})+\mathit{any}))+\mathrm{NN}))+\mathrm{NN})\cdot\mathrm{IS}\cdot\mathit{any}))\cdot\mathrm{IS}\cdot\mathit{any}^2\cdot\mathrm{IS}\cdot\mathit{any})+$
     $\mathrm{NN}+\mathit{any}^3)\cdot((\mathit{any}\cdot((((\mathit{any}\cdot(\mathit{any}^2+(\mathrm{IS}\cdot\mathit{any}^3)))+\mathit{any}^2)\cdot\mathrm{NN})+\mathrm{NN}))+$
     $\mathrm{NN})\cdot\mathit{any}))\cdot\mathit{any})))+\mathit{any}+\mathrm{NN}))))+\mathrm{NN}$
   - $\mathit{any}^2+\mathrm{IS}$
   - $\mathit{any}\cdot\mathrm{IS}\cdot\mathit{any}\cdot\mathrm{NN}$
   - $(\mathrm{NN}\cdot\mathrm{IS}\cdot\mathit{any}\cdot\mathrm{NN})+(\mathrm{IS}\cdot\mathit{any}\cdot((\mathrm{NN}\cdot(((\mathit{any}^3\cdot\mathrm{NN})?\cdot((\mathit{any}\cdot\mathrm{NN})+(((\mathit{any}\cdot$
     $(\mathit{any}^2+\mathit{any}))+\mathit{any}+\mathrm{IS})\cdot\mathit{any}))\cdot\mathit{any})+(\mathrm{IS}\cdot\mathit{any}^2\cdot\mathrm{IS}\cdot\mathit{any})))+\mathrm{NN}+\mathit{any})\cdot$
     $\mathit{any}\cdot(\mathit{any}\cdot(\mathit{any}^3+\mathrm{NN})\cdot\mathit{any}\cdot(\mathit{any}^2+(\mathit{any}\cdot(\mathit{any}^3+\mathit{any})\cdot\mathit{any}\cdot((\mathit{any}\cdot((\mathrm{IS}\cdot$
     $\mathit{any}\cdot((\mathit{any}\cdot((((\mathit{any}\cdot(\mathit{any}^3+\mathrm{NN})\cdot\mathit{any}\cdot((\mathit{any}\cdot((\mathrm{IS}\cdot\mathit{any}\cdot((\mathit{any}\cdot((((\mathrm{IS}\cdot$
     $\mathit{any}^2)+\mathit{any})\cdot\mathit{any})+\mathrm{NN}))+\mathrm{NN}+\mathit{any})\cdot((\mathit{any}\cdot((((\mathit{any}\cdot(\mathit{any}^2+\mathit{any}^4))+$
     $\mathit{any}^2)\cdot\mathrm{NN})+\mathrm{NN}))+\mathrm{NN})\cdot\mathit{any})+\mathrm{NN}))+\mathrm{IS})\cdot\mathit{any}^2\cdot(\mathit{any}\cdot(\mathit{any}^3+\mathrm{NN})\cdot$
     $\mathit{any}\cdot((\mathit{any}\cdot((\mathrm{IS}\cdot\mathit{any}\cdot((\mathit{any}\cdot((((\mathrm{IS}\cdot\mathit{any}^3)+(\mathrm{IS}\cdot\mathit{any}^2\cdot\mathrm{IS}\cdot\mathit{any}))\cdot\mathit{any})+$
     $(\mathrm{IS}\cdot\mathit{any}^3)))+\mathrm{NN}+\mathit{any})\cdot((\mathit{any}\cdot((((\mathit{any}\cdot(\mathit{any}^2+(\mathit{any}^3\cdot(\mathrm{IS}+(\mathit{any}\cdot((\mathrm{IS}\cdot$
     $\mathit{any}^2)+(\mathrm{IS}\cdot\mathit{any}^3)))+\mathrm{NN}+\mathit{any}))))+\mathit{any}^2)\cdot(((\mathit{any}+(\mathrm{IS}\cdot\mathit{any}))\cdot(\mathit{any}^2+$
     $\mathrm{NN}))+(\mathrm{IS}\cdot\mathit{any}\cdot\mathrm{IS}\cdot\mathit{any}))\cdot\mathrm{IS}\cdot\mathit{any}^2\cdot\mathrm{IS}\cdot\mathit{any}\cdot\mathrm{IS}\cdot\mathit{any})+\mathit{any}))+\mathrm{NN})\cdot$
     $\mathit{any})+\mathrm{NN}))+\mathrm{IS}))?\cdot\mathit{any}^2)+\mathit{any})\cdot\mathit{any})+(\mathrm{IS}\cdot\mathit{any}^3)))+\mathrm{NN}+\mathit{any})\cdot((\mathit{any}\cdot$
     $(((\mathit{any}^3+(\mathit{any}\cdot\mathrm{NN}))\cdot\mathrm{NN})+\mathrm{NN}))+\mathrm{NN})\cdot\mathit{any})+\mathrm{NN}))+\mathrm{IS}))))?\cdot\mathrm{NN}\cdot\mathit{any}^2)$
   - $\mathrm{IS}\cdot\mathit{any}\cdot((\mathit{any}\cdot((((\mathrm{IS}\cdot\mathit{any}^3)+\mathit{any})\cdot\mathit{any})+(\mathrm{IS}\cdot\mathit{any}^4)))+\mathrm{NN}+(\mathit{any}^2\cdot$
     $((\mathit{any}\cdot\mathrm{NN})+\mathit{any})))\cdot((\mathit{any}\cdot((\mathit{any}\cdot\mathrm{NN})+\mathrm{NN}))+\mathrm{NN})$

- $(\mathrm{NN} \cdot \mathrm{IS} \cdot \mathit{any} \cdot \mathrm{NN}) + (\mathit{any}^3 \cdot (\mathit{any}^2 + (\mathit{any} \cdot (\mathit{any}^3 + \mathrm{NN}) \cdot \mathit{any} \cdot ((\mathit{any} \cdot ((\mathit{any} \cdot ((\mathit{any} \cdot ((((\mathit{any} \cdot (\mathit{any}^2 + (\mathrm{NN} \cdot \mathit{any}^2))) + \mathit{any}^2) \cdot \mathrm{NN}) + \mathrm{NN})) + \mathrm{NN}) \cdot \mathit{any}) + \mathrm{NN})) + \mathrm{IS}))) \cdot \mathrm{NN} \cdot \mathit{any}^2)$
- $(\mathrm{NN} \cdot \mathrm{IS} \cdot \mathit{any} \cdot \mathrm{NN}) + (((\mathit{any} \cdot (\mathit{any}^3 + \mathrm{NN})) + (\mathrm{IS} \cdot \mathit{any})) \cdot ((\mathit{any} \cdot \mathrm{NN} \cdot \mathit{any}) + \mathrm{NN}) \cdot \mathrm{NN} \cdot (\mathit{any}^2 + (\mathrm{NN} \cdot \mathit{any}^2)))$
- $((\mathrm{NN} \cdot \mathit{any}^2 \cdot \mathrm{IS})? \cdot (\mathit{any} + (((\mathit{any} \cdot ((\mathit{any} \cdot \mathrm{NN}) + \mathit{any})) + \mathrm{NN} + \mathrm{IS}) \cdot \mathit{any}))) + \mathit{any}$
- $\mathrm{IS} \cdot \mathit{any} \cdot ((\mathrm{NN} \cdot ((\mathit{any} \cdot (\mathit{any} + (\mathrm{IS} \cdot \mathit{any}))) + ((((\mathit{any} + (\mathrm{IS} \cdot \mathit{any})) \cdot (\mathit{any}^2 + \mathrm{NN})) + (\mathrm{IS} \cdot \mathit{any})) \cdot \mathrm{IS} \cdot \mathit{any}^2 \cdot \mathrm{IS} \cdot \mathit{any} \cdot \mathrm{IS} \cdot \mathit{any}))) + \mathrm{NN} + \mathit{any} + (\mathit{any} \cdot (\mathit{any}^4 + \mathrm{NN}))) \cdot ((\mathit{any} \cdot ((\mathit{any} \cdot \mathrm{NN}) + \mathit{any})) + \mathrm{NN}) \cdot ((\mathrm{NN} \cdot (\mathit{any}^2 + (\mathrm{NN} \cdot ((\mathrm{NN} \cdot \mathrm{IS} \cdot \mathit{any} \cdot ((\mathit{any} \cdot \mathrm{NN} \cdot \mathit{any}) + ((\mathit{any}^3 \cdot \mathrm{NN})? \cdot ((\mathit{any} \cdot \mathrm{NN}) + (((\mathit{any} \cdot ((\mathit{any} \cdot \mathrm{NN}) + \mathit{any})) + \mathit{any} + \mathrm{IS}) \cdot \mathit{any})) \cdot \mathit{any}) + (\mathrm{IS} \cdot \mathit{any}^2 \cdot \mathrm{IS} \cdot \mathit{any}))) + \mathit{any}^2 + (\mathrm{IS} \cdot ((\mathrm{NN} \cdot ((\mathit{any} \cdot (\mathit{any} + (\mathrm{IS} \cdot \mathit{any}))) + ((((\mathit{any} + (\mathrm{IS} \cdot \mathit{any})) \cdot (\mathit{any}^2 + \mathrm{NN})) + (\mathrm{IS} \cdot \mathit{any})) \cdot \mathrm{IS} \cdot \mathit{any}^2 \cdot \mathrm{IS} \cdot ((\mathrm{IS} \cdot \mathit{any}) + \mathit{any}) \cdot \mathrm{IS} \cdot \mathit{any}))) + \mathrm{NN} + \mathit{any}) \cdot \mathit{any}^2)))))) + \mathrm{NN} + \mathit{any})$
- $(((((\mathit{any} + (\mathrm{NN} \cdot (\mathit{any}^3 + (\mathit{any}^3 \cdot \mathrm{NN} \cdot \mathit{any} \cdot \mathrm{IS} \cdot \mathit{any})))) \cdot (\mathit{any}^3 + \mathrm{NN})) + (\mathrm{IS} \cdot \mathit{any})) \cdot ((\mathit{any} \cdot \mathrm{NN} \cdot \mathit{any}) + \mathrm{NN})) + \mathrm{NN}) \cdot \mathrm{IS} \cdot \mathit{any} \cdot \mathrm{NN}$
- $(\mathrm{NN} \cdot \mathrm{IS} \cdot \mathit{any} \cdot \mathrm{NN}) + (\mathrm{IS} \cdot \mathit{any} \cdot ((\mathrm{NN} \cdot (((\mathit{any}^3 \cdot \mathrm{NN})? \cdot ((\mathit{any} \cdot \mathrm{NN}) + (((\mathit{any} \cdot ((\mathit{any} \cdot \mathrm{NN}) + \mathit{any})) + \mathit{any} + \mathrm{IS}) \cdot \mathit{any})) \cdot \mathit{any}) + (\mathrm{IS} \cdot \mathit{any}^2 \cdot \mathrm{IS} \cdot \mathit{any}))) + \mathrm{NN} + \mathit{any}) \cdot \mathit{any} \cdot (\mathit{any} \cdot (\mathit{any}^3 + \mathrm{NN}) \cdot \mathit{any} \cdot ((\mathit{any} \cdot ((\mathrm{IS} \cdot \mathit{any} \cdot ((\mathit{any} \cdot ((((\mathrm{IS} \cdot \mathit{any}^3) + \mathit{any}) \cdot \mathit{any}) + (\mathrm{IS} \cdot \mathit{any}^3))) + \mathrm{NN} + \mathit{any}) \cdot ((\mathit{any} \cdot ((((\mathit{any} \cdot (\mathit{any}^2 + (\mathit{any} \cdot (\mathit{any}^2 + (\mathit{any} \cdot ((\mathrm{IS} \cdot \mathit{any}^2) + (\mathrm{IS} \cdot \mathit{any}^3))) + \mathrm{NN} + \mathit{any})))) + \mathit{any}^2) \cdot (((\mathit{any} + (\mathrm{IS} \cdot \mathit{any})) \cdot (\mathit{any}^2 + \mathrm{NN})) + (\mathrm{IS} \cdot ((\mathit{any} \cdot \mathrm{NN}) + \mathrm{NN}) \cdot \mathrm{IS} \cdot \mathit{any})) \cdot \mathrm{IS} \cdot \mathit{any}^2 \cdot \mathrm{IS} \cdot \mathit{any}^2) + \mathrm{NN})) + \mathrm{NN}) \cdot \mathit{any}) + \mathrm{NN})) + \mathrm{IS}))? \cdot \mathrm{NN} \cdot \mathit{any}^2)$
- $(\mathrm{NN} \cdot \mathrm{IS} \cdot \mathit{any} \cdot \mathrm{NN}) + (\mathrm{NN} \cdot \mathit{any}^3 \cdot (\mathit{any}^2 + (\mathit{any} \cdot (\mathit{any}^2 + \mathrm{NN}) \cdot \mathit{any} \cdot ((\mathit{any} \cdot ((\mathrm{IS} \cdot \mathit{any} \cdot ((\mathit{any} \cdot (\mathit{any} + (\mathrm{IS} \cdot \mathit{any}^3))) + \mathrm{NN} + \mathit{any}) \cdot ((\mathit{any} \cdot ((((\mathit{any} \cdot \mathrm{NN}) + \mathit{any}^2) \cdot \mathrm{NN}) + \mathrm{NN})) + \mathrm{NN}) \cdot \mathit{any}) + \mathrm{NN})) + \mathrm{IS}))) \cdot \mathrm{NN} \cdot \mathit{any}^2)$
- $((((\mathit{any} \cdot (\mathit{any}^2 + \mathrm{NN})) + (\mathrm{IS} \cdot \mathit{any})) \cdot (\mathit{any}^3 + \mathrm{NN})) + \mathrm{NN}) \cdot \mathrm{IS} \cdot \mathit{any} \cdot \mathrm{NN}$
- $(\mathrm{NN} \cdot \mathrm{IS} \cdot \mathit{any} \cdot \mathrm{NN}) + (\mathrm{IS} \cdot \mathit{any} \cdot ((\mathrm{NN} \cdot (((\mathit{any} \cdot \mathrm{IS} \cdot \mathit{any}^3 \cdot \mathrm{NN})? \cdot ((\mathit{any} \cdot \mathrm{NN}) + \mathit{any}^2) \cdot \mathit{any}) + (\mathrm{IS} \cdot \mathit{any}^2 \cdot \mathrm{IS} \cdot \mathit{any}))) + \mathrm{NN} + \mathit{any}) \cdot \mathit{any} \cdot (((\mathit{any}^2 + \mathrm{NN} + \mathit{any}) \cdot \mathit{any}) + (\mathit{any}^3 \cdot ((\mathit{any} \cdot ((\mathrm{IS} \cdot \mathit{any} \cdot ((\mathit{any} \cdot ((((\mathrm{IS} \cdot \mathit{any}^3) + \mathit{any}) \cdot \mathit{any}) + (\mathrm{IS} \cdot \mathit{any}^4))) + \mathrm{NN} + (\mathit{any}^2 \cdot (\mathit{any} + (\mathrm{IS} \cdot \mathit{any})))) \cdot ((\mathit{any} \cdot ((((\mathit{any} \cdot (\mathit{any}^2 + (\mathrm{IS} \cdot \mathit{any}^3))) + \mathit{any}^2) \cdot \mathrm{NN}) + \mathrm{NN})) + \mathrm{NN}) \cdot \mathit{any}) + \mathrm{NN})) + \mathrm{IS}))) \cdot \mathrm{NN} \cdot (\mathit{any}^2 +$

$(\mathrm{IS}\cdot \mathit{any}\cdot((\mathrm{NN}\cdot(((\mathit{any}^2\cdot(((\mathit{any}+(\mathit{any}^3\cdot\mathrm{NN}))\cdot((\mathit{any}\cdot\mathrm{NN})+(((\mathit{any}\cdot((\mathit{any}\cdot$
$\mathrm{NN})+\mathit{any}))+\mathrm{NN}+\mathrm{IS})\cdot(((\mathit{any}^3\cdot\mathrm{NN})?\cdot((\mathit{any}\cdot\mathrm{NN})+(((\mathit{any}\cdot((\mathit{any}\cdot$
$\mathrm{NN})+\mathit{any}))+\mathrm{NN}+\mathrm{IS})\cdot\mathit{any})))+\mathit{any}+\mathrm{NN}))))+\mathrm{NN})\cdot\mathrm{NN})?\cdot((\mathit{any}^2\cdot$
$\mathrm{NN})+(((\mathit{any}\cdot((\mathit{any}\cdot\mathrm{NN})+\mathit{any}))+\mathit{any}+\mathrm{IS})\cdot\mathit{any}))\cdot\mathit{any})+(\mathrm{IS}\cdot\mathit{any}^2\cdot\mathrm{IS}\cdot$
$\mathit{any})))+\mathrm{NN}+\mathit{any})\cdot\mathit{any}\cdot(\mathrm{IS}\cdot((\mathit{any}\cdot((\mathrm{NN}\cdot((\mathit{any}\cdot(\mathit{any}+(\mathrm{IS}\cdot\mathit{any})))+(\mathrm{NN}\cdot$
$(\mathit{any}^2+(\mathrm{IS}\cdot\mathit{any}\cdot((\mathrm{NN}\cdot(((\mathit{any}^3\cdot\mathrm{NN})?\cdot((\mathrm{IS}\cdot\mathit{any}^2\cdot((\mathit{any}\cdot((\mathit{any}\cdot\mathrm{NN})+$
$\mathrm{NN})\cdot\mathit{any})+\mathrm{IS}+\mathit{any})\cdot\mathit{any})+\mathit{any}^2+(((\mathit{any}\cdot((\mathit{any}\cdot\mathrm{NN})+\mathit{any}))+\mathit{any}+$
$\mathrm{IS})\cdot\mathit{any}))\cdot\mathit{any})+(\mathrm{IS}\cdot\mathit{any}^2\cdot\mathrm{IS}\cdot\mathit{any})))+\mathrm{NN}+\mathit{any})\cdot\mathit{any}\cdot(\mathit{any}\cdot(\mathit{any}^3+$
$\mathrm{NN})\cdot\mathit{any}\cdot((\mathit{any}\cdot((\mathrm{IS}\cdot\mathit{any}\cdot(((\mathit{any}+(\mathit{any}^3\cdot\mathrm{NN}))\cdot((\mathit{any}\cdot\mathrm{NN})+(((\mathit{any}\cdot$
$((\mathit{any}\cdot\mathrm{NN})+\mathit{any}))+\mathrm{NN}+\mathrm{IS})\cdot(((\mathit{any}^3\cdot\mathrm{NN})?\cdot((\mathit{any}\cdot\mathrm{NN})+(((\mathit{any}\cdot((\mathit{any}\cdot$
$\mathrm{NN})+\mathit{any}))+\mathrm{NN}+\mathrm{IS})\cdot\mathit{any})))+\mathit{any}+\mathrm{NN}))))+\mathrm{NN})\cdot((\mathit{any}\cdot((((\mathit{any}\cdot$
$((\mathrm{IS}\cdot\mathit{any}\cdot(\mathit{any}^3+\mathrm{NN}))+\mathit{any}))+\mathit{any}^2)\cdot((\mathit{any}\cdot((\mathit{any}\cdot\mathrm{IS}\cdot\mathit{any}^2)+\mathrm{NN}))+$
$(\mathrm{IS}\cdot((((\mathit{any}\cdot((\mathit{any}^2\cdot((\mathit{any}^3\cdot((\mathit{any}\cdot((((\mathrm{IS}\cdot\mathit{any}^3)+(\mathit{any}\cdot((((\mathit{any}\cdot(\mathit{any}^2+$
$\mathrm{NN}))+\mathrm{NN})\cdot\mathrm{NN})+\mathrm{NN})))\cdot\mathit{any})+(\mathrm{IS}\cdot\mathit{any}^3)))+\mathrm{NN}+\mathit{any})\cdot((\mathit{any}\cdot((((\mathit{any}\cdot$
$(\mathit{any}^2+\mathrm{NN}))+\mathrm{NN})\cdot\mathrm{NN})+\mathrm{NN}))+\mathrm{NN})\cdot\mathit{any})+\mathit{any}^2))+\mathrm{NN}))+(\mathrm{IS}\cdot$
$\mathit{any}))\cdot((\mathit{any}\cdot\mathrm{NN}\cdot((\mathit{any}\cdot\mathrm{NN})+\mathit{any}))+\mathrm{NN}))+\mathrm{NN})\cdot\mathrm{IS}\cdot\mathit{any}))\cdot\mathrm{IS}\cdot\mathit{any}^2\cdot\mathrm{IS}\cdot$
$\mathit{any}\cdot\mathrm{IS}\cdot\mathit{any})+\mathrm{NN}))+\mathrm{NN})\cdot\mathit{any})+\mathrm{NN}))+\mathrm{IS}))?\cdot\mathrm{NN}\cdot\mathit{any}^4))\cdot\mathrm{IS}\cdot\mathit{any}^2\cdot\mathrm{IS}\cdot$
$\mathit{any}\cdot\mathrm{IS}\cdot\mathit{any})))+\mathrm{NN}+\mathit{any})\cdot\mathit{any})+\mathrm{NN})\cdot\mathit{any}\cdot((\mathit{any}\cdot((\mathit{any}\cdot((\mathit{any}\cdot((((\mathit{any}\cdot$
$((\mathrm{IS}\cdot\mathit{any}\cdot\mathrm{IS}\cdot\mathit{any})+(\mathit{any}^3\cdot\mathrm{NN}\cdot\mathit{any})))+\mathit{any}^2)\cdot((((\mathit{any}\cdot(\mathit{any}^3+(\mathrm{IS}\cdot\mathrm{NN}\cdot$
$\mathit{any}^2)))+\mathrm{NN}+\mathit{any}+(\mathrm{IS}\cdot\mathit{any}))\cdot(\mathit{any}^2+\mathrm{NN}))+(\mathrm{IS}\cdot((((\mathit{any}\cdot(\mathit{any}^4+$
$\mathrm{NN}))+(\mathrm{IS}\cdot\mathit{any}))\cdot((\mathit{any}\cdot\mathrm{NN}\cdot((\mathit{any}\cdot\mathrm{NN})+\mathit{any}))+\mathrm{NN}))+\mathrm{NN})\cdot\mathrm{IS}\cdot\mathit{any}))\cdot$
$\mathrm{IS}\cdot\mathit{any}^2\cdot\mathrm{IS}\cdot\mathit{any}\cdot\mathrm{IS}\cdot\mathit{any})+\mathrm{NN}))+\mathrm{NN})\cdot\mathit{any})+\mathrm{NN}))+\mathrm{IS}))?\cdot\mathrm{NN}\cdot\mathit{any}^4)))$

- $(((((\mathit{any}+(\mathit{any}\cdot(((\mathit{any}+(\mathit{any}^3\cdot\mathrm{NN}))\cdot((\mathit{any}\cdot\mathrm{NN})+(((\mathit{any}\cdot((\mathit{any}\cdot$
  $\mathrm{NN})+\mathit{any}))+\mathrm{NN}+\mathrm{IS})\cdot(((\mathit{any}^3\cdot\mathrm{NN})?\cdot((\mathit{any}\cdot\mathrm{NN})+(((\mathit{any}\cdot((\mathit{any}\cdot$
  $\mathrm{NN})+\mathit{any}))+\mathrm{NN}+\mathrm{IS})\cdot\mathit{any})))+\mathit{any}+\mathrm{NN}))))+\mathrm{NN}))+\mathrm{NN})\cdot(\mathit{any}^3+$
  $\mathrm{NN}))+(\mathrm{IS}^2\cdot\mathit{any}))\cdot((\mathit{any}\cdot\mathrm{NN}\cdot\mathit{any})+\mathrm{NN}))+\mathrm{NN})\cdot((\mathit{any}\cdot((\mathit{any}\cdot$
  $\mathrm{NN})+\mathit{any}))+\mathit{any}+\mathrm{IS})\cdot\mathit{any}^2\cdot\mathrm{NN}$

- $(\mathrm{NN}\cdot\mathrm{IS}\cdot\mathit{any}\cdot\mathrm{NN})+(\mathrm{IS}\cdot\mathit{any}\cdot((\mathrm{NN}\cdot(((\mathit{any}^3\cdot\mathrm{NN})?\cdot((\mathit{any}\cdot\mathrm{NN})+$
  $(((\mathit{any}\cdot((\mathit{any}\cdot\mathrm{NN})+\mathit{any}))+\mathit{any}+\mathrm{IS})\cdot\mathit{any}))\cdot\mathit{any})+(\mathrm{IS}\cdot((\mathrm{IS}\cdot\mathit{any}^2)+$
  $\mathit{any})\cdot\mathrm{NN}\cdot\mathrm{IS}\cdot\mathit{any})))+\mathrm{NN}+\mathit{any})\cdot\mathit{any}\cdot(\mathit{any}\cdot(\mathit{any}^3+\mathrm{NN})\cdot\mathit{any}\cdot((\mathit{any}\cdot$
  $((\mathrm{NN}\cdot\mathit{any})+\mathrm{NN}))+\mathrm{IS}))?\cdot\mathrm{NN}\cdot\mathit{any}^2)$

- $((((any \cdot ((((any \cdot (any^2 + \mathrm{NN})) + \mathrm{NN}) \cdot any) + \mathrm{NN})) + (\mathrm{IS} \cdot any)) \cdot ((any \cdot \mathrm{NN} \cdot any) + \mathrm{NN})) + \mathrm{NN}) \cdot \mathrm{IS} \cdot any \cdot \mathrm{NN}$

- $(\mathrm{NN} \cdot \mathrm{IS} \cdot any \cdot \mathrm{NN}) + (\mathrm{NN} \cdot any^3 \cdot (any^2 + (any \cdot (any^3 + \mathrm{NN}) \cdot any^2 \cdot ((any \cdot ((\mathrm{IS} \cdot any \cdot ((any \cdot (any + (\mathrm{IS} \cdot \mathrm{NN} \cdot any^2))) + \mathrm{NN} + any) \cdot ((any \cdot ((((any \cdot \mathrm{NN}) + any^2) \cdot \mathrm{NN}) + \mathrm{NN})) + \mathrm{NN}) \cdot any) + \mathrm{NN})) + \mathrm{IS}))) \cdot \mathrm{NN} \cdot any^2)$

- $(\mathrm{NN} \cdot \mathrm{IS} \cdot any \cdot \mathrm{NN}) + (\mathrm{IS} \cdot any \cdot ((\mathrm{NN} \cdot (((any \cdot \mathrm{IS} \cdot any^3 \cdot \mathrm{NN})? \cdot ((any \cdot \mathrm{NN}) + (((any \cdot ((any \cdot \mathrm{NN}) + any)) + any + \mathrm{IS}) \cdot any)) \cdot any) + (\mathrm{IS} \cdot any^2 \cdot \mathrm{IS} \cdot any))) + \mathrm{NN} + any) \cdot any \cdot (any^2 + (any^3 \cdot ((any \cdot ((\mathrm{IS} \cdot any \cdot ((any \cdot ((((\mathrm{IS} \cdot any^3) + any) \cdot any) + (\mathrm{IS} \cdot any^4))) + \mathrm{NN} + any^3) \cdot (any^2 + \mathrm{NN}) \cdot any) + \mathrm{NN})) + \mathrm{IS}))) \cdot \mathrm{NN} \cdot (any^2 + (\mathrm{IS} \cdot any \cdot ((\mathrm{NN} \cdot (((any^3 \cdot \mathrm{NN})? \cdot ((any^2 \cdot \mathrm{NN}) + (((any \cdot ((any \cdot \mathrm{NN}) + any)) + any + \mathrm{IS}) \cdot any)) \cdot any) + (\mathrm{IS} \cdot any))) + \mathrm{NN} + any) \cdot any \cdot (\mathrm{IS} \cdot ((any \cdot ((\mathrm{NN} \cdot ((any \cdot (any + (\mathrm{IS} \cdot any))) + ((((any + (\mathrm{IS} \cdot any)) \cdot (any^2 + \mathrm{NN})) + (\mathrm{IS} \cdot any)) \cdot \mathrm{IS} \cdot any^2 \cdot \mathrm{IS} \cdot any \cdot \mathrm{IS} \cdot any))) + \mathrm{NN} + any) \cdot any) + \mathrm{NN}) \cdot any \cdot ((any \cdot ((\mathrm{IS} \cdot any \cdot ((any \cdot ((((\mathrm{IS} \cdot any^3) + any) \cdot any) + (\mathrm{IS} \cdot any^2 \cdot ((\mathrm{IS} \cdot any \cdot ((\mathrm{IS} \cdot any \cdot ((((\mathrm{IS} \cdot any^3) + any) \cdot any) + (\mathrm{IS} \cdot any^3))) + \mathrm{NN} + (any^2 \cdot (any^2 + ((\mathrm{NN} + \mathrm{IS}) \cdot any))) + \mathrm{IS}) \cdot ((any \cdot ((((any \cdot (any^2 + \mathrm{NN})) + \mathrm{NN}) \cdot \mathrm{NN}) + \mathrm{NN})) + \mathrm{NN}) \cdot any) + \mathrm{IS} + any)))) + \mathrm{NN} + any) \cdot ((any \cdot ((((any \cdot ((\mathrm{IS} \cdot any^2 \cdot ((any^2 \cdot (any^2 + ((\mathrm{NN} + \mathrm{IS}) \cdot any) + any)) + \mathrm{IS}) \cdot any) + (\mathrm{IS} \cdot any^3))) + any^2) \cdot ((((any \cdot (any^3 + (\mathrm{IS} \cdot \mathrm{NN} \cdot any^2))) + \mathrm{NN} + any + (\mathrm{IS} \cdot any)) \cdot (any^2 + \mathrm{NN})) + (\mathrm{IS} \cdot ((((any \cdot (any^4 + \mathrm{NN})) + (\mathrm{IS} \cdot any)) \cdot ((any \cdot \mathrm{NN} \cdot ((any \cdot \mathrm{NN}) + any)) + \mathrm{NN})) + \mathrm{NN}) \cdot \mathrm{IS} \cdot any)) \cdot \mathrm{IS} \cdot any^2 \cdot \mathrm{IS} \cdot \mathrm{IS}? \cdot \mathrm{IS} \cdot any) + \mathrm{NN})) + \mathrm{NN}) \cdot any) + \mathrm{NN})) + \mathrm{IS}))? \cdot \mathrm{NN} \cdot any^4)))$

- $((((any \cdot (any^4 + \mathrm{NN})) + any^2) \cdot ((any \cdot \mathrm{NN} \cdot ((any \cdot \mathrm{NN}) + any)) + \mathrm{NN})) + \mathrm{NN}) \cdot \mathrm{IS} \cdot any \cdot \mathrm{NN}$

- $(\mathrm{NN} \cdot \mathrm{IS} \cdot any \cdot ((any \cdot \mathrm{NN} \cdot any) + ((any^3 \cdot \mathrm{NN})? \cdot any^2) + (any \cdot \mathrm{NN} \cdot any^2 \cdot \mathrm{IS} \cdot any))) + (\mathrm{IS} \cdot any \cdot \mathrm{NN} \cdot (any^2 + (\mathrm{IS} \cdot any^2 \cdot \mathrm{IS} \cdot any)) \cdot ((any \cdot \mathrm{NN}) + \mathrm{NN}) \cdot any)$

- $(\mathrm{NN} \cdot \mathrm{IS} \cdot any \cdot \mathrm{NN}) + (\mathrm{IS} \cdot any \cdot ((\mathrm{NN} \cdot (((any^2 \cdot \mathrm{NN})? \cdot ((any \cdot \mathrm{NN}) + (((any \cdot ((any \cdot \mathrm{NN}) + any)) + any + \mathrm{IS}) \cdot any)) \cdot any) + (\mathrm{IS} \cdot any^2 \cdot \mathrm{IS} \cdot any))) + \mathrm{NN} + any^2 + (any \cdot ((any \cdot \mathrm{NN}) + any)) + any + \mathrm{IS}) \cdot any \cdot (any^2 + (\mathrm{IS} \cdot ((any \cdot ((\mathrm{IS} \cdot any \cdot ((any \cdot ((((\mathrm{IS} \cdot any^3) + any) \cdot any) + (\mathrm{IS} \cdot any^4))) + \mathrm{NN} + any^3) \cdot ((any \cdot ((((any \cdot (any^2 + (\mathrm{IS} \cdot any^2 \cdot (\mathrm{IS} + \mathrm{NN})))) + any^2) \cdot \mathrm{NN}) + \mathrm{NN})) + \mathrm{NN}) \cdot any) + \mathrm{NN})) + \mathrm{IS}))) \cdot \mathrm{NN} \cdot (any^2 + (\mathrm{IS} \cdot any \cdot ((\mathrm{NN} \cdot (((any^3 \cdot \mathrm{NN})? \cdot ((any \cdot \mathrm{NN}) + (((any \cdot ((\mathrm{IS} \cdot any) + any)) + any + \mathrm{IS}) \cdot any)) \cdot any) +$

$(\mathrm{IS} \cdot any^2 \cdot \mathrm{IS} \cdot any))) + \mathrm{NN} + any) \cdot any \cdot (any \cdot (any^3 + \mathrm{NN}) \cdot any \cdot ((any \cdot ((\mathrm{IS} \cdot any \cdot ((any \cdot ((((\mathrm{IS} \cdot any^3) + any) \cdot any) + (\mathrm{IS} \cdot any^2 \cdot ((\mathrm{IS} \cdot any^2 \cdot ((any \cdot ((((any \cdot (any^2 + \mathrm{NN})) + \mathrm{NN}) \cdot \mathrm{NN}) + \mathrm{NN})) + \mathrm{NN}) \cdot any) + \mathrm{IS} + any)))) + \mathrm{NN} + any) \cdot ((any \cdot ((((any \cdot ((((any \cdot ((any \cdot \mathrm{NN}) + any)) + any + \mathrm{IS}) \cdot any \cdot (any^3 + \mathrm{NN})) + (\mathrm{IS} \cdot any^3))) + any^2) \cdot (((any + (\mathrm{IS} \cdot any)) \cdot (any^2 + \mathrm{NN})) + (\mathrm{IS} \cdot ((((any \cdot ((any^2 \cdot (any^2 + (any \cdot \mathrm{IS} \cdot any))) + \mathrm{NN})) + (\mathrm{IS} \cdot any)) \cdot ((any \cdot \mathrm{NN} \cdot ((any \cdot \mathrm{NN}) + any)) + \mathrm{NN})) + \mathrm{NN}) \cdot \mathrm{IS} \cdot any)) \cdot \mathrm{IS} \cdot any^2 \cdot \mathrm{IS} \cdot any \cdot \mathrm{IS} \cdot any) + \mathrm{NN})) + \mathrm{NN}) \cdot any) + \mathrm{NN})) + \mathrm{IS}))? \cdot \mathrm{NN} \cdot (\mathrm{NN} + any^2) \cdot any^2)))$

- $\mathrm{IS} \cdot any \cdot ((\mathrm{NN} \cdot ((any \cdot (any + (\mathrm{IS} \cdot any))) + ((((any + (\mathrm{IS} \cdot any)) \cdot (any^2 + \mathrm{NN})) + (\mathrm{IS} \cdot any)) \cdot \mathrm{IS} \cdot any \cdot ((((any \cdot (any^4 + \mathrm{NN})) + (\mathrm{IS} \cdot any)) \cdot ((any \cdot \mathrm{NN} \cdot ((any \cdot \mathrm{NN}) + any)) + \mathrm{NN})) + \mathrm{NN}) \cdot \mathrm{IS} \cdot any \cdot \mathrm{NN} \cdot \mathrm{IS} \cdot any \cdot \mathrm{IS} \cdot any))) + \mathrm{NN} + any) \cdot ((any \cdot ((any \cdot \mathrm{NN}) + any)) + \mathrm{NN}) \cdot ((\mathrm{NN} \cdot (any^2 + (\mathrm{NN} \cdot any))) + \mathrm{NN} + any)$
- $((((any \cdot ((any^2 \cdot (any^2 + (((\mathrm{NN} \cdot \mathrm{IS} \cdot any \cdot \mathrm{NN}) + (\mathrm{NN} \cdot any^3 \cdot (any^2 + (any \cdot (any^2 + \mathrm{NN}) \cdot any \cdot ((any \cdot ((\mathrm{IS} \cdot any \cdot ((any \cdot (any + (\mathrm{IS} \cdot any^3))) + \mathrm{NN} + any) \cdot ((any \cdot ((((any \cdot \mathrm{NN}) + any^2) \cdot \mathrm{NN}) + \mathrm{NN})) + \mathrm{NN}) \cdot any) + \mathrm{NN})) + \mathrm{IS}))) \cdot \mathrm{NN} \cdot any^2) + \mathrm{IS}) \cdot any))) + \mathrm{NN})) + (\mathrm{IS} \cdot any)) \cdot ((any \cdot \mathrm{NN} \cdot ((any \cdot \mathrm{NN}) + any)) + \mathrm{NN})) + \mathrm{NN}) \cdot \mathrm{IS} \cdot any \cdot \mathrm{NN}$
- $(\mathrm{NN} \cdot \mathrm{IS} \cdot any \cdot ((any \cdot \mathrm{NN} \cdot any) + ((any^3 \cdot \mathrm{NN})? \cdot ((any \cdot \mathrm{NN}) + (((any \cdot ((any \cdot \mathrm{NN}) + any)) + any + \mathrm{IS}) \cdot any)) \cdot any) + (\mathrm{IS} \cdot any^2 \cdot \mathrm{IS} \cdot any))) + (\mathrm{IS} \cdot any \cdot \mathrm{NN} \cdot (any^2 + (\mathrm{IS} \cdot any^2 \cdot \mathrm{IS} \cdot any)) \cdot ((any \cdot \mathrm{NN}) + \mathrm{NN}) \cdot any)$
- $(\mathrm{NN} \cdot \mathrm{IS} \cdot any \cdot \mathrm{NN}) + (\mathrm{IS} \cdot any \cdot ((\mathrm{NN} \cdot (any^3 + (any \cdot \mathrm{IS} \cdot any))) + \mathrm{NN} + (\mathrm{IS} \cdot any \cdot ((any \cdot ((((\mathrm{IS} \cdot any^2) + \mathrm{NN}) \cdot any) + (\mathrm{IS} \cdot any^3))) + \mathrm{NN} + any) \cdot ((any \cdot ((((any \cdot (any^2 + \mathrm{NN})) + \mathrm{NN}) \cdot \mathrm{NN}) + \mathrm{NN})) + \mathrm{NN}) \cdot any) + \mathrm{IS}) \cdot any \cdot (any^2 + (\mathrm{IS} \cdot any^2 \cdot ((\mathrm{IS} \cdot any^4) + \mathrm{IS}))) \cdot \mathrm{NN} \cdot ((((\mathrm{NN}? \cdot ((any \cdot \mathrm{NN}) + any^2)) + \mathrm{NN}) \cdot any) + (any \cdot \mathrm{NN})))$
- $\mathrm{IS} \cdot any \cdot ((\mathrm{NN} \cdot ((any \cdot (any + (\mathrm{IS} \cdot any))) + ((((any + (\mathrm{IS} \cdot any)) \cdot (any^2 + \mathrm{NN})) + (\mathrm{IS} \cdot any)) \cdot \mathrm{IS} \cdot any^2 \cdot \mathrm{IS} \cdot any \cdot \mathrm{IS} \cdot any))) + \mathrm{NN} + any + any^2) \cdot ((any \cdot ((any \cdot \mathrm{NN}) + any)) + \mathrm{NN}) \cdot ((\mathrm{NN} \cdot (any^2 + (\mathrm{NN} \cdot any))) + \mathrm{NN} + any)$
- $((((any \cdot (any^3 + \mathrm{NN})) + (\mathrm{IS}^2 \cdot any)) \cdot ((any \cdot \mathrm{NN} \cdot any) + \mathrm{NN})) + \mathrm{NN}) \cdot ((any \cdot ((any \cdot \mathrm{NN}) + any)) + (any \cdot (any^3 + \mathrm{NN}) \cdot any) + \mathrm{IS}) \cdot any^2 \cdot \mathrm{NN}$

- $(\mathrm{NN}\cdot\mathrm{IS}\cdot\mathit{any}\cdot\mathrm{NN})+(\mathrm{IS}\cdot\mathit{any}\cdot((\mathrm{NN}\cdot(((\mathit{any}^3\cdot\mathrm{NN})?\cdot((\mathit{any}\cdot\mathrm{NN})+(((\mathit{any}\cdot((\mathit{any}\cdot\mathrm{NN})+\mathit{any}))+\mathit{any}+\mathrm{IS})\cdot\mathit{any}))\cdot\mathit{any})+(\mathrm{IS}\cdot\mathit{any}\cdot\mathrm{NN}\cdot\mathit{any})))+\mathrm{NN}+\mathit{any})\cdot\mathit{any}\cdot(\mathit{any}\cdot(\mathit{any}^3+\mathrm{NN})\cdot\mathit{any}\cdot((\mathit{any}\cdot((\mathrm{IS}\cdot\mathit{any}\cdot((\mathit{any}\cdot(((\mathit{any}+(\mathit{any}^3\cdot\mathrm{NN}))\cdot((\mathit{any}\cdot\mathrm{NN})+(((\mathit{any}\cdot((\mathit{any}\cdot\mathrm{NN})+\mathit{any}))+\mathrm{NN}+\mathrm{IS})\cdot(((\mathit{any}^3\cdot\mathrm{NN})?\cdot((\mathit{any}\cdot\mathrm{NN})+\mathit{any}^2))+\mathit{any}+\mathrm{NN}))))+\mathrm{NN}))+\mathrm{NN}+\mathit{any})\cdot((\mathit{any}\cdot((((\mathit{any}\cdot(\mathit{any}^2+(\mathit{any}^3\cdot(\mathrm{IS}+(\mathit{any}\cdot(\mathit{any}+(\mathrm{IS}\cdot\mathit{any}^3)))+\mathrm{NN}+\mathit{any}))))+(\mathit{any}\cdot(\mathit{any}^2+(\mathit{any}^3\cdot(\mathrm{IS}+(\mathit{any}\cdot((\mathrm{IS}\cdot\mathit{any}^2)+(\mathrm{IS}\cdot\mathit{any}^3)))+\mathrm{NN}+\mathit{any})))\cdot\mathit{any}))\cdot(((\mathit{any}+(\mathrm{IS}\cdot\mathit{any}))\cdot(\mathit{any}^2+\mathrm{NN}))+(\mathrm{IS}\cdot((((\mathit{any}\cdot((\mathit{any}^2\cdot(\mathit{any}^2+((\mathrm{NN}+\mathrm{IS})\cdot\mathit{any})))+\mathrm{NN}))+(((\mathit{any}\cdot((\mathit{any}\cdot\mathrm{NN})+\mathit{any}))+\mathit{any}+\mathrm{IS})\cdot\mathit{any}))\cdot((\mathit{any}\cdot\mathrm{NN}\cdot((\mathit{any}\cdot\mathrm{NN})+\mathit{any}))+\mathrm{NN}))+\mathrm{NN})\cdot\mathrm{IS}\cdot\mathit{any}))\cdot\mathrm{IS}\cdot\mathit{any}^2\cdot\mathrm{IS}\cdot\mathit{any}\cdot\mathrm{IS}\cdot\mathit{any})+\mathrm{NN}))+\mathrm{NN})\cdot\mathit{any})+\mathrm{NN}))+\mathrm{IS}))?\cdot\mathrm{NN}\cdot\mathit{any}^2)$
- $((((\mathit{any}\cdot((\mathit{any}\cdot(\mathit{any}+(\mathit{any}\cdot\mathrm{NN}\cdot\mathit{any}))\cdot\mathit{any})+\mathrm{NN}))+(\mathrm{IS}\cdot\mathit{any}))\cdot((\mathit{any}\cdot\mathrm{NN}\cdot\mathit{any})+\mathrm{NN}))+\mathrm{NN})\cdot\mathrm{IS}\cdot\mathit{any}\cdot\mathrm{NN}$
- $(\mathrm{NN}\cdot\mathrm{IS}\cdot\mathit{any}\cdot\mathrm{NN})+(\mathrm{NN}\cdot\mathit{any}^3\cdot(\mathit{any}^2+(\mathit{any}\cdot(\mathit{any}^3+\mathrm{NN})\cdot\mathit{any}\cdot((\mathrm{NN}\cdot((\mathit{any}\cdot((\mathrm{IS}\cdot\mathit{any}\cdot((\mathit{any}\cdot((((\mathrm{IS}\cdot\mathit{any}^3)+\mathit{any})\cdot\mathit{any})+\mathit{any}^2))+\mathrm{NN}+(\mathrm{IS}\cdot((\mathrm{NN}\cdot((\mathit{any}\cdot(\mathit{any}+(\mathrm{IS}\cdot\mathit{any})))+((((\mathit{any}+(\mathrm{IS}\cdot\mathit{any}))\cdot(\mathit{any}^2+\mathrm{NN}))+(\mathrm{IS}\cdot\mathit{any}))\cdot\mathrm{IS}\cdot\mathit{any}^2\cdot\mathrm{IS}\cdot\mathit{any}\cdot\mathrm{IS}\cdot\mathit{any})))+\mathrm{NN}+\mathit{any})\cdot\mathit{any}^4))\cdot((\mathit{any}\cdot((((\mathit{any}\cdot(\mathit{any}^2+(\mathrm{IS}\cdot\mathit{any}^2\cdot(\mathrm{IS}+\mathrm{NN}))))+\mathit{any}^2)\cdot\mathrm{NN})+\mathrm{NN}))+\mathrm{NN})\cdot\mathit{any})+\mathrm{NN}))+\mathrm{NN}))+\mathrm{IS})))\cdot\mathrm{NN}\cdot\mathit{any}^2)$
- $(\mathrm{NN}\cdot\mathrm{IS}\cdot\mathit{any}\cdot\mathrm{NN})+(\mathrm{IS}\cdot\mathit{any}\cdot((\mathrm{NN}\cdot(((\mathit{any}^3\cdot\mathrm{NN})?\cdot((\mathit{any}\cdot\mathrm{NN})+(((\mathit{any}\cdot((\mathit{any}\cdot\mathrm{NN})+\mathit{any}))+\mathit{any}+\mathrm{IS})\cdot\mathit{any}))\cdot\mathit{any})+(\mathrm{IS}\cdot\mathit{any}^2\cdot\mathrm{IS}\cdot\mathit{any})))+\mathrm{NN}+\mathit{any})\cdot\mathit{any}\cdot(\mathit{any}+(\mathit{any}\cdot(\mathit{any}^3+\mathrm{NN})\cdot\mathit{any}\cdot((\mathit{any}\cdot(\mathit{any}^2+\mathrm{NN}))+\mathrm{IS})))\cdot\mathrm{NN}\cdot(\mathit{any}^2+(\mathrm{IS}\cdot\mathit{any}\cdot((\mathrm{NN}\cdot\mathit{any})+\mathrm{NN}+\mathit{any})\cdot\mathit{any}\cdot(\mathit{any}\cdot(\mathit{any}^3+\mathrm{NN})\cdot\mathit{any}\cdot((\mathit{any}\cdot((\mathrm{IS}\cdot\mathit{any}\cdot(((\mathit{any}+(\mathit{any}^3\cdot\mathrm{NN}))\cdot((\mathit{any}\cdot\mathrm{NN})+(((\mathit{any}\cdot((\mathit{any}\cdot\mathrm{NN})+\mathit{any}))+\mathrm{NN}+\mathrm{IS})\cdot(((\mathit{any}\cdot\mathrm{IS}\cdot\mathit{any}\cdot\mathrm{NN})?\cdot((\mathit{any}\cdot\mathrm{NN})+((\mathrm{NN}+\mathrm{IS})\cdot\mathit{any})))+\mathit{any}+\mathrm{NN}))))+\mathrm{NN})\cdot((\mathit{any}\cdot((((\mathit{any}\cdot((\mathrm{IS}\cdot\mathit{any}\cdot(\mathit{any}^3+\mathrm{NN}))+\mathit{any}))+\mathit{any}^2)\cdot(((\mathit{any}+(\mathrm{IS}\cdot\mathit{any}))\cdot((\mathit{any}\cdot\mathrm{IS}\cdot\mathit{any}^2)+\mathrm{NN}))+(\mathrm{IS}\cdot((((\mathit{any}\cdot((\mathit{any}^2\cdot((\mathit{any}\cdot\mathrm{IS}\cdot\mathit{any}\cdot((\mathit{any}\cdot((((\mathrm{IS}\cdot\mathit{any}^3)+(\mathit{any}\cdot((((\mathit{any}\cdot(\mathit{any}^2+\mathrm{NN}))+\mathrm{NN})\cdot\mathrm{NN})+\mathrm{NN})))\cdot\mathit{any})+(\mathrm{IS}\cdot\mathit{any}^3)))+\mathrm{NN}+\mathit{any})\cdot((\mathit{any}\cdot((((\mathit{any}\cdot(\mathit{any}^2+\mathrm{NN}))+\mathrm{NN})\cdot\mathrm{NN})+\mathrm{NN}))+\mathrm{NN})\cdot\mathit{any})+\mathit{any}^2))+\mathrm{NN}))+(\mathrm{IS}\cdot\mathit{any}))\cdot((\mathit{any}\cdot\mathrm{NN}\cdot((\mathit{any}\cdot\mathrm{NN})+\mathit{any}))+\mathrm{NN}))+\mathrm{NN})\cdot\mathrm{IS}\cdot\mathit{any}))\cdot\mathrm{IS}\cdot\mathit{any}^2\cdot\mathrm{IS}\cdot\mathit{any}\cdot\mathrm{IS}\cdot\mathit{any})+\mathrm{NN}))+\mathrm{NN})\cdot\mathit{any})+\mathrm{NN}))+\mathrm{IS}))?\cdot\mathrm{NN}\cdot\mathit{any}^4)))$

- $\mathrm{IS}\cdot \mathit{any}\cdot((\mathrm{NN}\cdot((\mathit{any}\cdot(\mathit{any}+(\mathrm{IS}\cdot \mathit{any})))+(\mathit{any}^2\cdot \mathrm{IS}\cdot \mathit{any}\cdot \mathrm{IS}\cdot \mathit{any})))+\mathit{any})\cdot ((\mathit{any}\cdot((\mathit{any}\cdot \mathrm{NN})+\mathit{any}))+\mathrm{NN})\cdot((\mathrm{NN}\cdot(\mathit{any}^2+(\mathrm{NN}\cdot \mathit{any})))+\mathrm{NN}+\mathit{any})$

- $(\mathrm{NN}\cdot \mathrm{IS}\cdot \mathit{any}\cdot \mathrm{NN})+(\mathrm{IS}\cdot \mathit{any}\cdot((\mathrm{NN}\cdot(((\mathit{any}^3\cdot \mathrm{NN})?\cdot((\mathit{any}\cdot \mathrm{NN})+((((\mathit{any}^3\cdot \mathrm{NN})?\cdot((\mathit{any}\cdot \mathrm{NN})+(((\mathit{any}\cdot((\mathit{any}\cdot \mathrm{NN})+\mathit{any}))+\mathit{any}+\mathrm{IS})\cdot \mathit{any}))\cdot \mathit{any})+ (\mathrm{IS}\cdot \mathit{any}^2\cdot \mathrm{IS}\cdot \mathit{any})+\mathrm{IS})\cdot \mathit{any}))\cdot \mathit{any})+(\mathrm{IS}\cdot \mathit{any}^2\cdot \mathrm{IS}\cdot \mathit{any})))+\mathrm{NN}+\mathit{any})\cdot \mathit{any}\cdot(\mathit{any}\cdot(\mathit{any}^3+\mathrm{NN})\cdot \mathit{any}\cdot(\mathit{any}^2+(\mathit{any}\cdot(\mathit{any}^3+\mathit{any})\cdot \mathit{any}\cdot((\mathit{any}\cdot ((\mathrm{IS}\cdot \mathit{any}\cdot \mathrm{IS}\cdot \mathit{any}\cdot((\mathrm{NN}\cdot(((\mathit{any}\cdot \mathrm{IS}\cdot \mathit{any}^3\cdot \mathrm{NN})?\cdot((\mathrm{NN}\cdot((\mathit{any}\cdot(\mathit{any}+(\mathrm{IS}\cdot \mathit{any})))+(\mathit{any}^2\cdot \mathrm{IS}\cdot \mathit{any}\cdot \mathrm{IS}\cdot \mathit{any})))+\mathit{any}+(((\mathit{any}\cdot((\mathit{any}\cdot \mathrm{NN})+\mathit{any}))+ \mathit{any}+\mathrm{IS})\cdot \mathit{any}))\cdot \mathit{any})+(\mathrm{IS}\cdot \mathit{any}^2\cdot \mathrm{IS}\cdot \mathit{any})))+\mathrm{NN}+\mathit{any})\cdot \mathit{any}\cdot(\mathit{any}^2+ (\mathit{any}\cdot(\mathit{any}+(\mathit{any}\cdot(\mathit{any}^2+\mathrm{NN}))+\mathrm{NN})\cdot \mathit{any}\cdot((\mathit{any}\cdot((\mathrm{IS}\cdot \mathit{any}\cdot((\mathit{any}\cdot((((\mathrm{IS}\cdot \mathit{any}^3)+\mathit{any})\cdot \mathit{any})+(\mathrm{IS}\cdot \mathit{any}^4)))+\mathrm{NN}+\mathit{any}^3)\cdot((\mathit{any}\cdot((((\mathit{any}\cdot(\mathit{any}^2+ (\mathrm{IS}\cdot \mathit{any}^3)))+\mathit{any}^2)\cdot \mathrm{NN})+\mathrm{NN}))+\mathrm{NN})\cdot \mathit{any})+\mathrm{NN}))+\mathrm{IS})))\cdot \mathrm{NN}\cdot(\mathit{any}^2+ (\mathrm{IS}\cdot \mathit{any}\cdot((\mathrm{NN}\cdot(((\mathit{any}^3\cdot \mathrm{NN})?\cdot((\mathit{any}^2\cdot \mathrm{NN})+(((\mathit{any}\cdot((\mathit{any}\cdot \mathrm{NN})+\mathit{any}))+ \mathit{any}+\mathrm{IS})\cdot \mathit{any}))\cdot \mathit{any})+(\mathrm{IS}\cdot \mathit{any}^2\cdot \mathrm{IS}\cdot \mathit{any})))+\mathrm{NN}+\mathit{any})\cdot \mathit{any}\cdot(\mathrm{IS}\cdot((\mathit{any}\cdot ((\mathrm{NN}\cdot((\mathit{any}\cdot(\mathit{any}+(\mathrm{IS}\cdot \mathit{any})))+(((((\mathit{any}\cdot((\mathit{any}\cdot \mathrm{NN})+\mathit{any}))+(\mathrm{IS}\cdot \mathit{any}))\cdot (\mathit{any}^2+\mathrm{NN}))+(\mathrm{IS}\cdot \mathit{any}))\cdot \mathrm{IS}\cdot \mathit{any}^2\cdot \mathrm{IS}\cdot \mathit{any}\cdot \mathrm{IS}\cdot \mathit{any})))+\mathrm{NN}+\mathit{any})\cdot \mathit{any})+ \mathrm{NN})\cdot \mathit{any}\cdot((\mathit{any}\cdot((\mathrm{IS}\cdot \mathit{any}\cdot((\mathrm{IS}\cdot((((\mathrm{IS}\cdot \mathit{any}^3)+\mathit{any})\cdot \mathit{any})+(\mathrm{IS}\cdot \mathit{any}^2\cdot((\mathrm{IS}\cdot \mathit{any}\cdot((\mathrm{IS}\cdot \mathit{any}\cdot((((\mathrm{IS}\cdot \mathit{any}^3)+\mathit{any})\cdot \mathit{any})+(\mathrm{IS}\cdot \mathit{any}^3)))+\mathrm{NN}+\mathit{any})\cdot((\mathit{any}\cdot ((((\mathit{any}\cdot((\mathit{any}\cdot(\mathrm{NN}\cdot \mathit{any}^3)?)+\mathrm{NN}))+\mathrm{NN})\cdot \mathrm{NN})+\mathrm{NN}))+\mathrm{NN})\cdot \mathit{any})+ \mathrm{IS}+\mathit{any}))))+\mathrm{NN}+\mathit{any})\cdot((\mathit{any}\cdot((((\mathit{any}\cdot((\mathrm{IS}\cdot \mathit{any}^2\cdot((\mathit{any}^2\cdot(\mathit{any}^2+((\mathrm{NN}+ \mathrm{IS})\cdot \mathit{any})+\mathit{any}))+\mathrm{IS})\cdot \mathit{any})+(\mathrm{IS}\cdot \mathit{any}^3)))+\mathit{any}^2)\cdot((((\mathit{any}\cdot(\mathit{any}^3+(\mathrm{IS}\cdot \mathrm{NN}\cdot \mathit{any}^2)))+\mathrm{NN}+\mathit{any}+(\mathrm{IS}\cdot \mathit{any}))\cdot(\mathit{any}^2+\mathrm{NN}))+(\mathrm{IS}\cdot((((\mathit{any}\cdot(\mathit{any}^4+ \mathrm{NN}))+(\mathrm{IS}\cdot \mathit{any}))\cdot((\mathit{any}\cdot \mathrm{NN}\cdot((\mathrm{IS}\cdot \mathit{any})+\mathit{any}))+\mathrm{NN}))+\mathrm{NN})\cdot \mathrm{IS}\cdot \mathit{any}))\cdot \mathrm{IS}\cdot \mathit{any}^2\cdot \mathrm{IS}\cdot \mathrm{IS}?\cdot \mathrm{IS}\cdot \mathit{any})+\mathrm{NN}))+\mathrm{NN})\cdot \mathit{any})+\mathrm{NN}))+\mathrm{IS}))?\cdot \mathrm{NN}\cdot \mathit{any}^4))\cdot ((\mathit{any}\cdot(((\mathit{any}^3+(\mathit{any}\cdot \mathrm{NN}))\cdot \mathrm{NN})+\mathrm{NN}))+\mathrm{NN})\cdot \mathit{any})+\mathrm{NN}))+\mathrm{IS})))))?\cdot \mathrm{NN})$

- $((\mathrm{NN}\cdot \mathit{any}^2\cdot \mathrm{NN})?\cdot((\mathit{any}\cdot \mathrm{NN})+(((\mathit{any}\cdot((\mathit{any}\cdot \mathrm{NN})+\mathit{any}))+\mathrm{NN}+ \mathrm{IS})\cdot \mathit{any})))+\mathit{any}$

- $\mathrm{IS}\cdot \mathit{any}\cdot((\mathrm{NN}\cdot((\mathit{any}\cdot(\mathit{any}+(\mathrm{IS}\cdot \mathit{any})))+((((\mathit{any}+(\mathrm{IS}\cdot \mathit{any}))\cdot(\mathit{any}^2+\mathrm{NN}))+ (\mathrm{IS}\cdot \mathit{any}))\cdot \mathrm{IS}\cdot \mathit{any}^2\cdot \mathrm{IS}\cdot \mathit{any}\cdot \mathrm{IS}\cdot \mathit{any})))+\mathrm{NN}+\mathit{any}+(((\mathit{any}\cdot(\mathit{any}^3+\mathrm{NN})\cdot \mathit{any}\cdot ((\mathit{any}\cdot((\mathrm{IS}\cdot \mathit{any}\cdot((\mathit{any}\cdot((((\mathrm{IS}\cdot \mathit{any}^2)+\mathit{any})\cdot \mathit{any})+\mathrm{NN}))+\mathrm{NN}+\mathit{any})\cdot((\mathit{any}\cdot ((((\mathit{any}\cdot(\mathit{any}^2+\mathit{any}^4))+\mathit{any}^2)\cdot \mathrm{NN})+\mathrm{NN}))+\mathrm{NN})\cdot \mathit{any})+\mathrm{NN}))+\mathrm{IS})\cdot \mathrm{IS}\cdot \mathit{any}\cdot((\mathrm{NN}\cdot(((\mathit{any}^3\cdot \mathrm{NN})?\cdot((\mathit{any}\cdot \mathrm{NN})+(((\mathit{any}\cdot((\mathit{any}\cdot \mathrm{NN})+\mathit{any}))+\mathit{any}+$

$\mathrm{IS})\cdot \mathit{any}))\cdot \mathit{any})+(\mathrm{IS}\cdot \mathit{any}^2\cdot \mathrm{IS}\cdot \mathit{any})))+\mathrm{NN}+\mathit{any})\cdot \mathit{any}\cdot(\mathit{any}\cdot((\mathrm{NN}\cdot \mathit{any}^2)+$
$\mathrm{NN})\cdot \mathit{any}\cdot((\mathit{any}\cdot((\mathrm{IS}\cdot \mathit{any}\cdot((\mathit{any}\cdot((\mathit{any}\cdot((((\mathrm{IS}\cdot \mathit{any}^3)+\mathit{any})\cdot(\mathit{any}+(\mathrm{IS}\cdot \mathit{any}\cdot$
$((\mathrm{NN}\cdot(\mathit{any}^2+((((\mathit{any}+(\mathrm{NN}\cdot \mathit{any}^2\cdot \mathrm{NN}\cdot \mathit{any}))\cdot(\mathit{any}^2+\mathrm{IS}+(((\mathit{any}\cdot((\mathrm{IS}\cdot \mathit{any}\cdot$
$((\mathit{any}^2\cdot(\mathit{any}^2+\mathrm{NN}))+\mathrm{NN}))+(\mathrm{NN}\cdot \mathit{any}^3)))+\mathit{any}^2)\cdot((\mathit{any}\cdot(\mathit{any}^2+\mathrm{NN}))+$
$(\mathrm{IS}\cdot(((((\mathit{any}+\mathit{any}^3)\cdot((\mathit{any}\cdot(\mathit{any}^3+\mathrm{NN})\cdot \mathit{any}\cdot((\mathit{any}\cdot((\mathrm{IS}\cdot \mathit{any}\cdot((\mathit{any}\cdot((((\mathrm{IS}\cdot$
$\mathit{any}\cdot \mathrm{IS}\cdot \mathit{any})+\mathit{any})\cdot \mathit{any})+(\mathrm{IS}\cdot \mathit{any}^3)))+\mathrm{NN}+\mathit{any})\cdot((\mathit{any}\cdot((((\mathit{any}\cdot(((\mathrm{NN}\cdot$
$\mathit{any}^2\cdot \mathrm{NN})?\cdot((\mathit{any}\cdot \mathrm{NN})+(((\mathit{any}\cdot((\mathit{any}\cdot \mathrm{NN})+\mathit{any}))+\mathrm{NN}+\mathrm{IS})\cdot \mathit{any})))+$
$(\mathit{any}^3\cdot(\mathrm{IS}+(\mathit{any}\cdot((\mathrm{IS}\cdot \mathit{any}^2)+(\mathrm{IS}\cdot \mathit{any}\cdot((\mathrm{NN}\cdot \mathrm{IS}\cdot \mathit{any}\cdot \mathrm{NN})+(\mathrm{IS}\cdot \mathit{any}\cdot((\mathrm{NN}\cdot$
$((\mathit{any}\cdot((\mathit{any}\cdot((\mathrm{IS}\cdot \mathit{any}\cdot \mathrm{NN}\cdot(\mathit{any}^2+(\mathrm{IS}\cdot \mathit{any}^2\cdot \mathrm{IS}\cdot((\mathit{any}\cdot \mathrm{NN})+(((\mathit{any}\cdot((\mathit{any}\cdot$
$\mathrm{NN})+\mathit{any}))+\mathrm{NN}+\mathrm{IS})\cdot \mathit{any})+\mathrm{IS}))))+\mathit{any}))+\mathit{any}^2))+(\mathit{any}^3\cdot \mathrm{NN}\cdot \mathit{any}\cdot$
$\mathrm{IS}\cdot \mathit{any})))+\mathrm{NN}+\mathit{any}^2+\mathrm{IS})\cdot \mathit{any}\cdot(\mathit{any}^2+(\mathrm{IS}\cdot \mathit{any}^2\cdot((\mathrm{IS}\cdot \mathit{any}^4)+\mathrm{IS})))\cdot \mathrm{NN}\cdot$
$((((\mathrm{NN}?\cdot((\mathit{any}\cdot \mathrm{NN})+\mathit{any}^2))+\mathrm{NN})\cdot \mathit{any})+(\mathit{any}\cdot \mathrm{NN}))))\cdot \mathit{any})))+\mathrm{NN}+$
$\mathit{any}))))+\mathit{any}^2)\cdot(((\mathit{any}+(\mathrm{IS}\cdot \mathit{any}))\cdot(\mathit{any}^2+\mathrm{NN}))+(\mathrm{IS}\cdot((\mathrm{IS}\cdot \mathit{any}^2\cdot((\mathit{any}\cdot$
$\mathrm{NN}^2)+\mathrm{NN}))+\mathrm{NN})\cdot \mathrm{IS}\cdot \mathit{any}))\cdot \mathrm{IS}\cdot \mathit{any}^2\cdot \mathrm{IS}\cdot \mathit{any}\cdot \mathrm{IS}\cdot \mathit{any})+\mathrm{NN}))+\mathrm{NN})\cdot$
$\mathit{any})+\mathrm{NN}))+\mathrm{IS})\cdot \mathit{any}\cdot(\mathit{any}^2+((\mathrm{IS}+(\mathit{any}\cdot((\mathrm{IS}\cdot \mathit{any}^2)+(\mathrm{IS}\cdot \mathit{any}^2\cdot(\mathrm{NN}+$
$(\mathrm{NN}\cdot(\mathit{any}^2+(\mathrm{NN}\cdot \mathit{any})))+\mathit{any}))))+\mathit{any}+\mathrm{NN})\cdot \mathit{any})))+\mathrm{NN}))+\mathrm{IS}^2)\cdot$
$((\mathit{any}\cdot \mathrm{NN}\cdot(\mathit{any}^2+\mathit{any}))+(\mathrm{IS}\cdot \mathit{any})))+\mathrm{NN})\cdot \mathrm{IS}\cdot \mathit{any}))\cdot \mathrm{IS}\cdot \mathit{any}^2\cdot \mathrm{IS}\cdot \mathit{any}\cdot \mathrm{IS}\cdot$
$\mathit{any})))+(\mathrm{IS}\cdot \mathit{any}))\cdot \mathit{any}^3\cdot \mathrm{NN}\cdot \mathit{any}^3\cdot \mathrm{IS}\cdot \mathit{any})))+\mathrm{NN}+\mathit{any})\cdot(\mathit{any}^2+\mathrm{NN}))))+$
$(\mathrm{IS}\cdot \mathit{any}^4)))+\mathrm{NN}+(\mathrm{IS}\cdot \mathit{any}^3)))+\mathrm{NN}+\mathit{any})\cdot((\mathit{any}\cdot((((\mathit{any}\cdot(\mathit{any}^2+$
$(\mathit{any}^3\cdot(\mathrm{IS}+(\mathit{any}\cdot((\mathrm{IS}\cdot \mathit{any}^2)+\mathit{any}^3))+\mathrm{NN}+\mathit{any}))))+\mathit{any}^2)\cdot(((\mathit{any}+$
$(\mathrm{IS}\cdot \mathit{any}))\cdot(\mathit{any}^2+\mathrm{NN}))+(\mathrm{IS}\cdot \mathit{any}\cdot \mathrm{IS}\cdot \mathit{any}))\cdot \mathrm{IS}\cdot \mathit{any}^2\cdot \mathrm{IS}\cdot((\mathrm{IS}\cdot \mathit{any})+$
$\mathit{any})\cdot \mathrm{IS}\cdot \mathit{any})+\mathit{any}))+\mathrm{NN})\cdot \mathit{any})+\mathrm{NN}))+\mathrm{IS}))?\cdot \mathit{any}^2)+\mathit{any})\cdot \mathit{any})+(\mathrm{IS}\cdot$
$\mathit{any}^3))\cdot((\mathit{any}\cdot((\mathit{any}\cdot \mathrm{NN})+\mathit{any}))+\mathrm{NN})\cdot((\mathrm{NN}\cdot(\mathit{any}^2+(\mathrm{NN}\cdot((\mathrm{NN}\cdot \mathrm{IS}\cdot$
$\mathit{any}\cdot((\mathit{any}\cdot \mathrm{NN}\cdot \mathit{any})+((\mathit{any}^3\cdot \mathrm{NN})?\cdot((\mathit{any}\cdot \mathrm{NN})+(((\mathit{any}\cdot((\mathit{any}\cdot \mathrm{NN})+$
$\mathit{any}))+\mathit{any}+\mathrm{IS})\cdot \mathit{any}))\cdot \mathit{any})+(\mathrm{IS}\cdot \mathit{any}^2\cdot \mathrm{IS}\cdot \mathit{any})))+\mathit{any}))))+\mathrm{NN}+\mathit{any})$

- $(((((\mathit{any}+(\mathrm{NN}\cdot(\mathit{any}^3+(\mathrm{NN}^2\cdot \mathit{any}^2))))\cdot(\mathit{any}^3+\mathrm{NN}))+(\mathrm{IS}\cdot \mathit{any}))\cdot$
  $((\mathit{any}\cdot \mathrm{NN}\cdot \mathit{any})+\mathrm{NN}))+\mathrm{NN})\cdot \mathrm{IS}\cdot \mathit{any}\cdot \mathrm{NN}$
- $(\mathrm{NN}\cdot \mathrm{IS}\cdot \mathit{any}\cdot \mathrm{NN})+(\mathrm{IS}\cdot \mathit{any}\cdot((\mathrm{NN}\cdot(((\mathit{any}^3\cdot \mathrm{NN})?\cdot((\mathit{any}\cdot \mathrm{NN})+$
  $(((\mathit{any}\cdot((\mathit{any}\cdot \mathrm{NN})+\mathit{any}))+\mathit{any}+\mathrm{IS})\cdot \mathit{any}))\cdot \mathit{any})+(\mathrm{IS}\cdot \mathit{any}^2\cdot \mathrm{IS}\cdot$
  $\mathit{any})))+\mathrm{NN}+\mathit{any})\cdot \mathit{any}\cdot(\mathit{any}\cdot(\mathit{any}^3+\mathrm{NN})\cdot \mathit{any}\cdot((\mathit{any}\cdot((\mathrm{IS}\cdot \mathit{any}\cdot$
  $((\mathit{any}\cdot((((\mathrm{IS}\cdot \mathit{any}^3)+\mathit{any})\cdot \mathit{any})+(\mathrm{IS}\cdot \mathit{any}^3)))+\mathrm{NN}+\mathit{any})\cdot((\mathit{any}\cdot$

$((((any \cdot (any^2 + (any^3 \cdot (any^2 + (any \cdot ((\mathrm{IS} \cdot any^2) + (\mathrm{IS} \cdot any^3))) + \mathrm{NN} + any)))) + any^2) \cdot any) + \mathrm{NN})) + \mathrm{NN}) \cdot any) + \mathrm{NN})) + \mathrm{IS}))? \cdot \mathrm{NN} \cdot any^2)$

- $((((any \cdot ((((any \cdot (any^2 + \mathrm{NN})) + \mathrm{NN}) \cdot any) + \mathrm{NN})) + (\mathrm{IS} \cdot any)) \cdot ((any \cdot ((any \cdot ((any \cdot \mathrm{NN}) + any)) + any + \mathrm{IS}) \cdot any) + \mathrm{NN})) + \mathrm{NN}) \cdot \mathrm{IS} \cdot any \cdot \mathrm{NN}$

- $(\mathrm{NN} \cdot \mathrm{IS} \cdot any \cdot \mathrm{NN}) + (\mathrm{NN} \cdot any^3 \cdot (any^2 + (any \cdot (any^2 + \mathrm{NN}) \cdot any \cdot ((any \cdot ((\mathrm{IS} \cdot any \cdot ((any \cdot (any + (\mathrm{IS} \cdot any^3))) + \mathrm{NN} + any) \cdot ((any \cdot ((((any \cdot \mathrm{NN}) + any^2) \cdot \mathrm{NN}) + \mathrm{NN})) + \mathrm{NN}) \cdot any^3) + \mathrm{NN})) + \mathrm{IS}))) \cdot \mathrm{NN} \cdot any^2)$

- $((((any \cdot (any^2 + \mathrm{NN})) + (\mathrm{IS} \cdot any)) \cdot ((any \cdot \mathrm{NN} \cdot any) + \mathrm{NN})) + \mathrm{NN}) \cdot \mathrm{IS} \cdot any \cdot \mathrm{NN}$

- $(\mathrm{NN} \cdot \mathrm{IS} \cdot any \cdot \mathrm{NN}) + (\mathrm{IS} \cdot any \cdot ((\mathrm{NN} \cdot (((any \cdot \mathrm{IS} \cdot any^3 \cdot \mathrm{NN})? \cdot ((any \cdot \mathrm{NN}) + (((any \cdot ((any \cdot \mathrm{NN}) + any)) + any + \mathrm{IS}) \cdot any)) \cdot any) + (\mathrm{IS} \cdot any^2 \cdot \mathrm{IS} \cdot any))) + \mathrm{NN} + any) \cdot any \cdot (any^2 + (any^3 \cdot ((any \cdot ((\mathrm{IS} \cdot any \cdot ((any \cdot ((((\mathrm{IS} \cdot any^3) + any) \cdot any) + (\mathrm{IS} \cdot any^4))) + \mathrm{NN} + any^3) \cdot ((any \cdot ((((any \cdot (any^2 + (\mathrm{IS} \cdot any^3))) + any^2) \cdot \mathrm{NN}) + \mathrm{NN})) + \mathrm{NN}) \cdot any) + \mathrm{NN})) + \mathrm{IS}))) \cdot \mathrm{NN} \cdot (any^2 + (\mathrm{IS} \cdot any \cdot ((\mathrm{NN} \cdot (((any^3 \cdot \mathrm{NN})? \cdot (any + (((any \cdot ((any \cdot \mathrm{NN}) + any)) + any + \mathrm{IS}) \cdot any)) \cdot any) + (\mathrm{IS} \cdot any^2 \cdot \mathrm{IS} \cdot any))) + \mathrm{NN} + any) \cdot any \cdot (\mathrm{IS} \cdot ((any \cdot ((\mathrm{NN} \cdot ((any \cdot (any + (\mathrm{IS} \cdot any))) + ((((any + (\mathrm{IS} \cdot any)) \cdot (any^2 + \mathrm{NN})) + (\mathrm{IS} \cdot any)) \cdot \mathrm{IS} \cdot any^2 \cdot \mathrm{IS} \cdot any \cdot \mathrm{IS} \cdot any))) + any + (\mathrm{IS} \cdot any \cdot (((any + (any^3 \cdot \mathrm{NN})) \cdot ((any \cdot \mathrm{NN}) + (((any \cdot ((any \cdot \mathrm{NN}) + any)) + \mathrm{NN} + \mathrm{IS}) \cdot (((any^3 \cdot \mathrm{NN})? \cdot ((any \cdot \mathrm{NN}) + (((any \cdot ((any \cdot \mathrm{NN}) + any)) + \mathrm{NN} + \mathrm{IS}) \cdot any))) + any + \mathrm{NN})))) + \mathrm{NN}))) \cdot any) + \mathrm{NN}) \cdot any \cdot ((any \cdot ((\mathrm{IS} \cdot any \cdot ((any \cdot ((((\mathrm{IS} \cdot any^3) + any) \cdot any) + (\mathrm{IS} \cdot any^2 \cdot ((\mathrm{IS} \cdot any \cdot ((\mathrm{IS} \cdot any \cdot ((((\mathrm{IS} \cdot any^3) + any) \cdot any) + (\mathrm{IS} \cdot any^3))) + \mathrm{NN} + any) \cdot ((any \cdot ((((any \cdot (any^2 + \mathrm{NN})) + \mathrm{NN}) \cdot \mathrm{NN}) + \mathrm{NN})) + \mathrm{NN}) \cdot any) + \mathrm{IS} + any)))) + \mathrm{NN} + any) \cdot ((any \cdot ((((any \cdot ((\mathrm{IS} \cdot any^2 \cdot ((any^2 \cdot (any^2 + ((\mathrm{NN} + \mathrm{IS}) \cdot any) + any)) + \mathrm{IS}) \cdot any) + (\mathrm{IS} \cdot any^3))) + any^2) \cdot ((((any \cdot (any^3 + (\mathrm{IS} \cdot \mathrm{NN} \cdot any^2))) + \mathrm{NN} + any + (\mathrm{IS} \cdot any)) \cdot (any^2 + \mathrm{NN})) + (\mathrm{IS} \cdot ((((any \cdot (any^4 + \mathrm{NN})) + (\mathrm{IS} \cdot any)) \cdot ((any \cdot \mathrm{NN} \cdot ((any \cdot \mathrm{NN}) + any)) + \mathrm{NN})) + \mathrm{NN}) \cdot \mathrm{IS} \cdot any)) \cdot \mathrm{IS} \cdot any^2 \cdot \mathrm{IS} \cdot \mathrm{IS}? \cdot \mathrm{IS} \cdot any) + \mathrm{NN})) + \mathrm{NN}) \cdot any) + \mathrm{NN})) + \mathrm{IS}))? \cdot \mathrm{NN} \cdot any^4)))$

- $any^2 \cdot ((((any \cdot (any^2 + \mathrm{NN})) + \mathrm{NN}) \cdot \mathrm{NN}) + any) \cdot ((any \cdot ((any \cdot \mathrm{NN}) + any)) + \mathrm{NN}) \cdot ((\mathrm{NN} \cdot (any^2 + (\mathrm{NN} \cdot any))) + \mathrm{NN} + any)$

- $(\mathrm{NN} \cdot \mathrm{IS} \cdot \mathit{any} \cdot ((\mathit{any} \cdot \mathrm{NN} \cdot \mathit{any}) + ((\mathit{any}^3 \cdot \mathrm{NN})? \cdot ((\mathit{any} \cdot \mathrm{NN}) + (((\mathit{any} \cdot ((\mathit{any} \cdot \mathrm{NN}) + \mathit{any})) + \mathit{any} + ((\mathit{any}^3 \cdot \mathrm{NN})? \cdot ((\mathit{any} \cdot \mathrm{NN}) + (((\mathit{any} \cdot ((\mathit{any} \cdot \mathrm{NN}) + \mathit{any})) + \mathit{any} + \mathrm{IS}) \cdot \mathit{any})) \cdot \mathit{any})) \cdot \mathit{any})) \cdot \mathit{any}) + (\mathit{any} \cdot \mathrm{NN} \cdot \mathit{any}^2 \cdot \mathrm{IS} \cdot \mathit{any}))) + (\mathrm{IS} \cdot \mathit{any} \cdot \mathrm{NN} \cdot (\mathit{any}^2 + (\mathrm{IS} \cdot \mathit{any}^2 \cdot \mathrm{IS} \cdot \mathit{any})) \cdot ((\mathit{any} \cdot \mathrm{NN}) + \mathrm{NN}) \cdot \mathit{any})$
- $(\mathrm{NN} \cdot \mathrm{IS} \cdot \mathit{any} \cdot \mathrm{NN}) + (\mathrm{IS} \cdot \mathit{any} \cdot ((\mathrm{NN} \cdot (((\mathit{any}^2 \cdot \mathrm{NN})? \cdot ((\mathrm{IS} \cdot \mathrm{NN}) + (((\mathit{any} \cdot ((\mathit{any} \cdot \mathrm{NN}) + \mathit{any})) + \mathit{any} + \mathrm{IS}) \cdot \mathit{any})) \cdot \mathit{any}) + (\mathrm{IS} \cdot \mathit{any}^2 \cdot \mathrm{IS} \cdot \mathit{any}))) + \mathrm{NN} + \mathit{any}^2 + (\mathit{any} \cdot ((\mathit{any} \cdot \mathrm{NN}) + \mathit{any})) + \mathit{any} + \mathrm{IS}) \cdot \mathit{any} \cdot (\mathit{any}^2 + (\mathrm{IS} \cdot ((\mathit{any} \cdot ((\mathrm{IS} \cdot \mathit{any} \cdot ((\mathit{any} \cdot ((((\mathrm{IS} \cdot \mathit{any}^3) + \mathit{any}) \cdot \mathit{any}) + (\mathrm{IS} \cdot \mathit{any}^4))) + \mathrm{NN} + \mathit{any}^3) \cdot ((\mathit{any} \cdot ((((\mathit{any} \cdot (\mathit{any}^2 + (\mathrm{IS} \cdot \mathit{any}^2 \cdot (\mathrm{IS} + \mathrm{NN})))) + \mathit{any}^2) \cdot \mathrm{NN}) + \mathrm{NN})) + \mathrm{NN}) \cdot \mathit{any}) + \mathrm{NN})) + \mathrm{IS}))) \cdot \mathrm{NN} \cdot (\mathit{any}^2 + (\mathrm{IS} \cdot \mathit{any} \cdot ((\mathrm{NN} \cdot (((\mathit{any}^3 \cdot \mathrm{NN})? \cdot ((\mathit{any} \cdot \mathrm{NN}) + (\mathrm{IS} \cdot \mathit{any}^2)) \cdot \mathit{any}) + (\mathrm{IS} \cdot \mathit{any}^2 \cdot \mathrm{IS} \cdot \mathit{any}))) + \mathrm{NN} + \mathit{any}) \cdot \mathit{any} \cdot (\mathit{any} \cdot (\mathit{any}^3 + \mathrm{NN}) \cdot \mathit{any} \cdot ((\mathit{any} \cdot ((\mathrm{IS} \cdot \mathit{any} \cdot ((\mathit{any} \cdot ((((\mathrm{IS} \cdot \mathit{any}^3) + \mathit{any}) \cdot \mathit{any}) + (\mathit{any} \cdot (\mathit{any}^3 + \mathit{any})))) + \mathrm{NN} + \mathit{any}) \cdot ((\mathit{any} \cdot ((((\mathit{any} \cdot ((((\mathit{any} \cdot (\mathit{any}^2 + \mathit{any})) + \mathit{any} + \mathrm{IS}) \cdot \mathit{any} \cdot (\mathit{any}^3 + \mathrm{NN})) + (\mathrm{IS} \cdot \mathit{any}^3))) + \mathit{any}^2) \cdot (((\mathit{any} + (\mathrm{IS} \cdot \mathit{any})) \cdot (\mathit{any}^2 + \mathrm{NN})) + (\mathrm{IS} \cdot (((\mathit{any}^2 + (\mathrm{IS} \cdot \mathit{any})) \cdot ((\mathit{any} \cdot \mathrm{NN} \cdot ((\mathit{any} \cdot \mathrm{NN}) + \mathit{any})) + \mathrm{NN})) + \mathrm{NN}) \cdot \mathrm{IS} \cdot \mathit{any})) \cdot \mathrm{IS} \cdot \mathit{any}^2 \cdot \mathrm{IS} \cdot \mathit{any} \cdot \mathrm{IS} \cdot \mathit{any}) + \mathrm{NN})) + \mathrm{NN}) \cdot \mathit{any}) + \mathrm{NN})) + \mathrm{IS}))? \cdot \mathrm{NN} \cdot (\mathrm{NN} + \mathit{any}^2) \cdot \mathit{any}^2)))$
- $(\mathrm{NN} \cdot \mathrm{IS} \cdot \mathit{any} \cdot ((\mathit{any} \cdot \mathrm{NN} \cdot \mathit{any}) + ((\mathit{any}^3 \cdot \mathrm{NN})? \cdot ((\mathit{any} \cdot \mathrm{NN}) + \mathit{any}^2) \cdot \mathit{any}) + (\mathrm{IS} \cdot \mathit{any}^2 \cdot \mathrm{IS} \cdot \mathit{any}))) + (\mathrm{IS} \cdot \mathit{any} \cdot \mathrm{NN} \cdot (\mathit{any}^2 + (\mathrm{IS} \cdot \mathit{any}^2 \cdot \mathrm{IS} \cdot \mathit{any})) \cdot ((\mathit{any} \cdot \mathrm{NN}) + \mathrm{NN}) \cdot \mathit{any})$
- $(\mathrm{NN} \cdot \mathrm{IS} \cdot \mathit{any} \cdot \mathrm{NN}) + (\mathrm{IS} \cdot \mathit{any} \cdot ((\mathrm{NN} \cdot (\mathit{any}^3 + (\mathit{any} \cdot \mathrm{IS} \cdot \mathit{any}))) + \mathrm{NN} + (\mathrm{IS} \cdot \mathit{any} \cdot ((\mathit{any} \cdot ((((\mathrm{IS} \cdot \mathit{any}^2) + \mathit{any}) \cdot \mathit{any}) + (\mathrm{IS} \cdot \mathit{any}^3))) + \mathrm{NN} + \mathit{any}) \cdot ((\mathit{any} \cdot ((((\mathit{any} \cdot (\mathit{any}^2 + \mathrm{NN})) + \mathrm{NN}) \cdot \mathrm{NN}) + \mathrm{NN})) + \mathrm{NN}) \cdot \mathit{any}) + \mathrm{IS}) \cdot \mathit{any} \cdot (\mathit{any}^2 + (\mathrm{IS} \cdot \mathit{any}^2 \cdot ((\mathrm{IS} \cdot \mathit{any}^4) + \mathrm{IS}))) \cdot \mathrm{NN} \cdot ((((\mathrm{NN}? \cdot ((\mathit{any} \cdot \mathrm{NN}) + \mathit{any}^2)) + \mathrm{NN}) \cdot \mathit{any}) + (\mathit{any} \cdot \mathrm{NN})))$
- $\mathrm{IS} \cdot \mathit{any} \cdot ((\mathrm{NN} \cdot ((\mathit{any} \cdot (\mathit{any} + (\mathrm{IS} \cdot \mathit{any}))) + ((((\mathit{any} + (\mathrm{IS} \cdot \mathit{any})) \cdot (\mathit{any}^2 + \mathrm{NN})) + (\mathrm{IS} \cdot \mathit{any})) \cdot \mathrm{IS} \cdot \mathit{any}^2 \cdot \mathrm{IS} \cdot \mathit{any} \cdot \mathrm{IS} \cdot \mathit{any}))) + \mathrm{NN} + \mathit{any}) \cdot ((\mathit{any} \cdot ((\mathit{any} \cdot \mathrm{NN}) + \mathit{any})) + \mathrm{NN}) \cdot ((\mathrm{NN} \cdot (\mathit{any}^2 + (\mathrm{NN} \cdot \mathit{any}))) + \mathrm{NN} + \mathit{any})$
- $(\mathrm{IS} \cdot \mathit{any}^3) + \mathit{any} + \mathrm{IS}$
- $((((\mathit{any} \cdot (\mathit{any}^3 + \mathrm{NN})) + (\mathrm{IS}^2 \cdot \mathit{any})) \cdot ((\mathit{any} \cdot \mathrm{NN} \cdot \mathit{any}) + \mathrm{NN})) + \mathrm{NN}) \cdot ((\mathit{any} \cdot ((\mathit{any} \cdot \mathrm{NN}) + \mathit{any})) + (\mathrm{IS} \cdot \mathit{any} \cdot ((\mathit{any} \cdot ((((\mathrm{IS} \cdot \mathit{any}^2) + \mathit{any}) \cdot \mathit{any}) + \mathrm{NN})) + \mathrm{NN} + \mathit{any}) \cdot ((\mathit{any} \cdot ((((\mathit{any} \cdot (\mathit{any}^2 + \mathit{any}^4)) + \mathit{any}^2) \cdot \mathrm{NN}) + \mathrm{NN})) + \mathrm{NN})) + \mathrm{IS}) \cdot \mathit{any}^2 \cdot \mathrm{NN}$

- $(\mathrm{NN}\cdot\mathrm{IS}\cdot \mathit{any}\cdot\mathrm{NN})+(\mathrm{IS}\cdot \mathit{any}\cdot((\mathrm{NN}\cdot(((\mathit{any}^3\cdot\mathrm{NN})?\cdot((\mathit{any}\cdot\mathrm{NN})+(((\mathit{any}\cdot((\mathit{any}\cdot\mathrm{NN})+\mathit{any}))+\mathit{any}+\mathrm{IS})\cdot \mathit{any}))\cdot \mathit{any})+(\mathrm{IS}\cdot \mathit{any}\cdot\mathrm{NN}\cdot\mathrm{IS}\cdot \mathit{any})))+\mathrm{NN}+\mathit{any})\cdot \mathit{any}\cdot(\mathit{any}\cdot(\mathit{any}^3+\mathrm{NN})\cdot \mathit{any}\cdot((\mathit{any}\cdot((\mathrm{IS}\cdot \mathit{any}\cdot(\mathit{any}^2+\mathrm{NN}+\mathit{any})\cdot((\mathit{any}\cdot((((\mathit{any}\cdot(\mathit{any}^2+(\mathit{any}^3\cdot(\mathrm{IS}+(\mathit{any}\cdot(\mathit{any}+(\mathrm{IS}\cdot \mathit{any}^3)))+\mathrm{NN}+\mathit{any}))))+(\mathit{any}\cdot(\mathit{any}^2+(\mathit{any}^3\cdot(\mathrm{IS}+(\mathit{any}\cdot((\mathrm{IS}\cdot \mathit{any}^2)+(\mathrm{IS}\cdot \mathit{any}^3)))+\mathrm{NN}+\mathit{any})))\cdot \mathit{any}))\cdot(((\mathit{any}+(\mathrm{IS}\cdot \mathit{any}))\cdot(\mathit{any}^2+\mathrm{NN}))+(\mathrm{IS}\cdot((((\mathit{any}\cdot((\mathit{any}^2\cdot(\mathit{any}^2+((\mathrm{NN}+\mathrm{IS})\cdot \mathit{any})))+\mathrm{NN}))+(((\mathit{any}\cdot((\mathit{any}\cdot\mathrm{NN})+\mathit{any}))+\mathit{any}+\mathrm{IS})\cdot \mathit{any}))\cdot((\mathit{any}\cdot\mathrm{NN}\cdot((\mathit{any}\cdot\mathrm{NN})+\mathit{any}))+\mathrm{NN}))+\mathrm{NN})\cdot\mathrm{IS}\cdot \mathit{any}))\cdot\mathrm{IS}\cdot \mathit{any}^2\cdot\mathrm{IS}\cdot \mathit{any}\cdot\mathrm{IS}\cdot \mathit{any})+\mathrm{NN}))+\mathrm{NN})\cdot \mathit{any})+\mathrm{NN}))+\mathrm{IS}))?\cdot\mathrm{NN}\cdot \mathit{any}^2)$
- $((((\mathit{any}\cdot((\mathit{any}\cdot(\mathit{any}+(\mathit{any}\cdot\mathrm{NN}\cdot \mathit{any}))\cdot(((\mathit{any}^3\cdot\mathrm{NN})?\cdot((\mathit{any}\cdot\mathrm{NN})+(((\mathit{any}\cdot((\mathit{any}\cdot\mathrm{NN})+\mathit{any}))+\mathit{any}+\mathrm{IS})\cdot \mathit{any}))\cdot \mathit{any})+(\mathrm{IS}\cdot \mathit{any}^2\cdot\mathrm{IS}\cdot \mathit{any})+\mathit{any}))+\mathrm{NN}))+(\mathrm{IS}\cdot \mathit{any}))\cdot((\mathit{any}\cdot\mathrm{NN}\cdot \mathit{any})+\mathrm{NN}))+\mathrm{NN})\cdot\mathrm{IS}\cdot \mathit{any}\cdot\mathrm{NN}$
- $(\mathrm{NN}\cdot\mathrm{IS}\cdot \mathit{any}\cdot\mathrm{NN})+(\mathrm{NN}\cdot \mathit{any}^3\cdot(\mathit{any}^2+(\mathit{any}\cdot(\mathit{any}^3+\mathrm{NN})\cdot \mathit{any}\cdot((\mathrm{NN}\cdot((\mathrm{IS}\cdot \mathit{any}\cdot((\mathit{any}\cdot((((\mathrm{IS}\cdot \mathit{any}^3)+\mathit{any})\cdot \mathit{any})+(\mathrm{IS}\cdot \mathit{any}^3)))+\mathrm{NN}+\mathit{any})\cdot((\mathit{any}\cdot((((\mathit{any}\cdot(\mathit{any}^2+\mathit{any}^4))+\mathit{any}^2)\cdot\mathrm{NN})+\mathrm{NN}))+\mathrm{NN})\cdot \mathit{any})+\mathrm{NN}))+\mathrm{IS})))\cdot\mathrm{NN}\cdot \mathit{any}^2)$
- $((((\mathit{any}\cdot(\mathit{any}^3+\mathrm{NN}))+(\mathrm{IS}\cdot \mathit{any}))\cdot((\mathit{any}\cdot\mathrm{NN}\cdot \mathit{any})+\mathrm{NN}))+\mathrm{NN})\cdot\mathrm{IS}\cdot \mathit{any}\cdot\mathrm{NN}$
- $(\mathrm{NN}\cdot\mathrm{IS}\cdot \mathit{any}\cdot\mathrm{NN})+(\mathrm{IS}\cdot \mathit{any}\cdot((\mathrm{NN}\cdot(((\mathit{any}^3\cdot\mathrm{NN})?\cdot((\mathit{any}\cdot\mathrm{NN})+(((\mathit{any}\cdot((\mathit{any}\cdot\mathrm{NN})+\mathit{any}))+\mathit{any}+\mathrm{IS})\cdot \mathit{any}))\cdot \mathit{any})+(\mathrm{IS}\cdot \mathit{any}^2\cdot\mathrm{IS}\cdot \mathit{any})))+\mathrm{NN}+\mathit{any})\cdot \mathit{any}\cdot(\mathit{any}^2+(\mathit{any}\cdot(\mathit{any}^3+\mathrm{NN})\cdot \mathit{any}\cdot((\mathit{any}\cdot(\mathit{any}^2+\mathrm{NN}))+\mathrm{IS})))\cdot\mathrm{NN}\cdot(\mathit{any}^2+(\mathrm{IS}\cdot \mathit{any}\cdot((\mathrm{NN}\cdot(((\mathit{any}^3\cdot\mathrm{NN})?\cdot(\mathrm{NN}+(((\mathit{any}\cdot((\mathit{any}\cdot\mathrm{NN})+\mathit{any}))+\mathit{any}+\mathrm{IS})\cdot \mathit{any}))\cdot \mathit{any})+(\mathrm{IS}\cdot \mathit{any}^2\cdot\mathrm{IS}\cdot \mathit{any})))+\mathrm{NN}+\mathit{any})\cdot \mathit{any}\cdot(\mathit{any}\cdot(\mathit{any}^3+\mathrm{NN})\cdot \mathit{any}\cdot((\mathit{any}\cdot((\mathrm{IS}\cdot \mathit{any}\cdot(((\mathit{any}+(\mathit{any}^3\cdot\mathrm{NN}))\cdot((\mathit{any}\cdot\mathrm{NN})+(((\mathit{any}\cdot((\mathit{any}\cdot\mathrm{NN})+\mathit{any}))+\mathrm{NN}+\mathrm{IS})\cdot(((\mathit{any}^3\cdot\mathrm{NN})?\cdot((\mathit{any}\cdot\mathrm{NN})+(((\mathit{any}\cdot((\mathit{any}\cdot\mathrm{NN})+\mathit{any}))+\mathrm{NN}+\mathrm{IS})\cdot \mathit{any})))+\mathit{any}+\mathrm{NN}))))+\mathrm{NN})\cdot((\mathit{any}\cdot((((\mathit{any}\cdot((\mathrm{IS}\cdot \mathit{any}\cdot(\mathit{any}^3+\mathrm{NN}))+\mathit{any}))+(\mathit{any}\cdot\mathrm{NN}))\cdot(((\mathit{any}+(\mathrm{IS}\cdot \mathit{any}))\cdot((\mathit{any}\cdot\mathrm{IS}\cdot \mathit{any}^2)+\mathrm{NN}))+(\mathrm{IS}\cdot((((\mathit{any}\cdot((\mathit{any}^2\cdot((\mathit{any}\cdot\mathrm{IS}\cdot \mathit{any}\cdot((\mathit{any}\cdot((((\mathrm{IS}\cdot \mathit{any}^3)+(\mathit{any}\cdot((((\mathit{any}\cdot(\mathit{any}^2+\mathit{any}))+\mathrm{NN})\cdot\mathrm{NN})+\mathrm{NN})))\cdot((((((\mathit{any}\cdot(\mathit{any}^4+\mathrm{NN}))+(\mathrm{IS}\cdot \mathit{any}))\cdot((\mathit{any}\cdot\mathrm{NN}\cdot((\mathit{any}\cdot\mathrm{NN})+\mathit{any}))+\mathrm{NN}))+\mathrm{NN})\cdot\mathrm{IS}\cdot \mathit{any})+\mathit{any}))+(\mathrm{IS}\cdot \mathit{any}^3)))+\mathrm{NN}+\mathit{any})\cdot((\mathit{any}\cdot((((\mathit{any}\cdot(\mathit{any}^2+\mathrm{NN}))+\mathrm{NN})\cdot\mathrm{NN})+\mathrm{NN}))+\mathrm{NN})\cdot \mathit{any})+\mathit{any}^2))+\mathrm{NN}))+(\mathrm{IS}\cdot \mathit{any}))\cdot((\mathit{any}\cdot\mathrm{NN}\cdot((\mathit{any}\cdot\mathrm{NN})+(\mathit{any}\cdot$

$((((\mathit{any} \cdot (\mathit{any}^2 + \mathrm{NN})) + \mathrm{NN}) \cdot \mathrm{NN}) + \mathrm{NN})) + \mathrm{NN})) + \mathrm{NN})) + \mathrm{NN}) \cdot \mathrm{IS})) \cdot \mathrm{IS} \cdot \mathit{any}^2 \cdot \mathrm{IS} \cdot \mathit{any} \cdot \mathrm{IS} \cdot \mathit{any}) + \mathrm{NN})) + \mathrm{NN}) \cdot \mathit{any}) + \mathrm{NN})) + \mathrm{IS}))? \cdot \mathrm{NN} \cdot \mathit{any}^4)))$

- $\mathrm{IS} \cdot \mathit{any} \cdot ((\mathrm{NN} \cdot ((\mathit{any} \cdot (\mathit{any} + (\mathrm{IS} \cdot \mathit{any}))) + (\mathit{any}^3 \cdot \mathrm{IS} \cdot \mathit{any}))) + \mathit{any}) \cdot ((\mathit{any} \cdot ((\mathit{any} \cdot \mathrm{NN}) + \mathit{any})) + \mathrm{NN}) \cdot ((\mathrm{NN} \cdot (\mathit{any}^2 + (\mathrm{NN} \cdot \mathit{any}))) + \mathrm{NN} + \mathit{any})$
- $((((\mathit{any} \cdot (\mathit{any}^4 + \mathrm{NN})) + (\mathrm{IS} \cdot \mathit{any})) \cdot ((\mathit{any} \cdot \mathrm{NN} \cdot ((\mathit{any} \cdot \mathrm{NN}) + \mathit{any})) + \mathrm{NN})) + \mathrm{NN}) \cdot \mathrm{IS} \cdot \mathit{any} \cdot \mathrm{NN}$
- $(\mathrm{NN} \cdot \mathrm{IS} \cdot \mathit{any} \cdot ((\mathit{any} \cdot \mathrm{NN} \cdot \mathit{any}) + ((\mathit{any}^3 \cdot \mathrm{NN})? \cdot ((\mathit{any} \cdot \mathrm{NN}) + (((\mathit{any} \cdot ((\mathit{any} \cdot \mathrm{NN}) + \mathit{any})) + \mathit{any} + \mathrm{IS}) \cdot \mathit{any})) \cdot \mathit{any}) + (\mathrm{IS} \cdot \mathit{any}^2 \cdot \mathrm{IS} \cdot \mathit{any}))) + (\mathrm{IS} \cdot \mathit{any} \cdot \mathrm{NN} \cdot (\mathit{any}^2 + (\mathrm{IS} \cdot \mathit{any}^2 \cdot ((\mathrm{NN} \cdot \mathit{any} \cdot ((\mathit{any} \cdot \mathrm{NN}) + \mathit{any})) + \mathrm{NN} + \mathit{any}) \cdot \mathit{any})) \cdot ((\mathit{any} \cdot \mathrm{NN}) + \mathrm{NN}) \cdot \mathit{any})$
- $(\mathrm{NN} \cdot \mathrm{IS} \cdot \mathit{any} \cdot \mathrm{NN}) + (\mathrm{IS} \cdot \mathit{any} \cdot ((\mathrm{NN} \cdot (\mathit{any}^3 + (\mathit{any}^3 \cdot \mathrm{NN} \cdot \mathit{any} \cdot \mathrm{IS} \cdot \mathit{any}))) + \mathrm{NN} + \mathit{any}^2 + \mathrm{IS}) \cdot \mathit{any} \cdot (\mathit{any}^2 + (\mathrm{IS} \cdot \mathit{any}^2 \cdot ((\mathrm{IS} \cdot \mathit{any}^4) + \mathrm{IS}))) \cdot \mathrm{NN} \cdot ((((\mathrm{NN}? \cdot ((\mathit{any} \cdot \mathrm{NN}) + \mathit{any}^4)) + \mathrm{NN}) \cdot \mathit{any}) + (\mathit{any} \cdot \mathrm{NN})))$
- $\mathrm{IS} \cdot \mathit{any} \cdot ((\mathrm{NN} \cdot (\mathit{any}^2 + ((((\mathit{any} + (\mathrm{NN} \cdot \mathit{any})) \cdot (\mathit{any}^2 + \mathit{any})) + (\mathrm{IS} \cdot \mathit{any})) \cdot \mathit{any}^2 \cdot \mathrm{IS} \cdot \mathit{any} \cdot ((\mathit{any} \cdot ((\mathit{any}^2 \cdot (\mathit{any}^2 + ((\mathrm{NN} + \mathrm{IS}) \cdot \mathit{any}))) + \mathrm{NN})) + (((\mathit{any} \cdot ((\mathit{any} \cdot \mathrm{NN}) + \mathit{any})) + \mathit{any} + \mathrm{IS}) \cdot \mathit{any}))))) + \mathrm{NN} + \mathit{any}) \cdot ((\mathit{any} \cdot ((\mathit{any} \cdot \mathrm{NN}) + \mathrm{NN})) + \mathrm{NN}) \cdot ((\mathrm{NN} \cdot \mathrm{IS} \cdot \mathit{any}) + \mathrm{NN} + \mathit{any})$
- $(\mathrm{IS} \cdot \mathit{any}^4) + \mathrm{IS}$

### A.1.2 GP Experiments with a Larger Set of Features

1. Best individual for ExpF1; Satisfy all the following:
   - $\mathrm{DET}^2$
   - $\mathrm{IS} \cdot \mathrm{DET} \cdot \mathrm{NN}$
   - $\mathrm{NN} \cdot \mathrm{IS} \cdot \mathrm{DET}$
2. Best individual for ExpF2; Satisfy all the following:
   - $\mathrm{NN} \cdot \mathrm{IS} \cdot \mathrm{DET}$
   - $\mathrm{NN} \cdot \mathrm{IS} \cdot \mathrm{DET}$
   - $\mathrm{IS} \cdot \mathrm{DET}^2$

### A.1.3 GP Experiments with Multiple Runs

1. Best individual for ExpM1; Satisfy all the following:

   - $\mathrm{NN} \cdot \mathrm{IS} \cdot \mathrm{DET} \cdot \mathrm{NN}$
   - $\mathrm{IS} \cdot \mathrm{DET}$

2. Best individual for ExpM2; Satisfy all the following:

   - $\mathrm{NN}^{*} \cdot \mathrm{IS}$
   - $\mathrm{IS}^{*} \cdot \mathrm{IS} \cdot \mathrm{DET}$
   - $\mathrm{NN} \cdot \mathrm{IS} \cdot \mathrm{DET} \cdot \mathrm{NN}$

3. Best individual for ExpM3; Satisfy all the following:

   - $\mathrm{NN} \cdot ((((\mathrm{DET} \cdot \mathrm{IS}^{*})^{*} \cdot ((((\mathrm{IS} \cdot \mathrm{IS}^{*})^{*} \cdot (\mathrm{IS}^{2} \cdot \mathrm{DET})^{*} \cdot \mathrm{IS}^{*} \cdot \mathrm{DET})^{*} \cdot \mathrm{DET} \cdot \mathrm{IS})^{*} \cdot (\mathrm{IS}^{2} \cdot \mathrm{DET})^{*})^{*} \cdot \mathrm{IS}^{*})^{*} \cdot \mathrm{DET})^{*} \cdot \mathrm{MD})^{*} \cdot \mathrm{IS} \cdot \mathrm{DET} \cdot \mathrm{NN}$

4. Best individual for ExpM4; Satisfy all the following:

   - $\mathrm{NN} \cdot ((\mathit{any}^{*} \cdot ((\mathrm{NN}^{4} \cdot \mathrm{NN}^{*} \cdot \mathrm{IS} \cdot \mathrm{DET} \cdot \mathrm{NN} \cdot ((\mathit{any}^{*} \cdot (((\mathit{any} \cdot (\mathrm{NN} \cdot (\mathit{any} \cdot (\mathit{any} \cdot \mathrm{NN} \cdot \mathit{any} \cdot (\mathit{any}^{2} \cdot \mathrm{NN} \cdot \mathit{any}^{2})^{*} \cdot (\mathit{any} \cdot (\mathrm{NN} \cdot (\mathit{any} \cdot \mathrm{NN}^{2} \cdot (\mathrm{NN}^{2} \cdot \mathit{any})^{*} \cdot ((\mathit{any} \cdot \mathrm{NN} \cdot (((\mathit{any} \cdot (\mathrm{NN} \cdot (\mathit{any} \cdot (\mathit{any} \cdot \mathrm{NN} \cdot \mathit{any} \cdot (\mathit{any}^{2} \cdot \mathrm{NN} \cdot \mathit{any}^{2})^{*} \cdot (\mathit{any} \cdot (\mathrm{NN} \cdot (\mathit{any} \cdot \mathrm{NN}^{2} \cdot (\mathit{any}^{*} \cdot (((\mathit{any} \cdot (\mathrm{NN} \cdot (\mathit{any} \cdot (\mathit{any} \cdot \mathrm{NN} \cdot \mathit{any} \cdot (\mathit{any}^{2} \cdot \mathrm{NN} \cdot \mathit{any}^{2})^{*} \cdot (\mathit{any} \cdot (\mathrm{NN} \cdot (\mathit{any} \cdot \mathrm{NN}^{2} \cdot (\mathit{any} \cdot \mathrm{NN} \cdot \mathit{any})^{*} \cdot \mathrm{IS} \cdot \mathrm{DET} \cdot \mathrm{IS} \cdot \mathrm{DET})^{*} \cdot \mathit{any} \cdot (\mathit{any} \cdot \mathrm{NN} \cdot (\mathit{any} \cdot \mathrm{NN}^{2} \cdot (\mathit{any} \cdot \mathrm{NN} \cdot \mathit{any})^{*} \cdot \mathrm{IS} \cdot \mathrm{DET} \cdot \mathrm{IS} \cdot \mathrm{DET})^{*} \cdot \mathit{any})^{*} \cdot \mathit{any})^{*} \cdot \mathit{any})^{*})^{*} \cdot \mathit{any})^{*} \cdot \mathit{any} \cdot (\mathit{any} \cdot \mathrm{NN} \cdot \mathit{any})^{*} \cdot \mathit{any})^{*} \cdot \mathit{any})^{*} \cdot \mathrm{NN} \cdot \mathit{any})^{*} \cdot \mathrm{NN}^{2})^{*} \cdot \mathit{any})^{*} \cdot \mathrm{IS} \cdot \mathrm{DET} \cdot \mathrm{IS} \cdot \mathrm{DET})^{*} \cdot \mathit{any} \cdot (\mathit{any} \cdot \mathrm{NN} \cdot (\mathit{any} \cdot \mathrm{NN}^{2} \cdot (\mathit{any} \cdot \mathrm{NN} \cdot \mathit{any})^{*} \cdot \mathrm{IS} \cdot \mathrm{DET} \cdot \mathrm{IS} \cdot \mathrm{DET})^{*} \cdot \mathit{any})^{*} \cdot \mathit{any})^{*} \cdot \mathit{any})^{*})^{*} \cdot \mathit{any})^{*} \cdot \mathit{any} \cdot (\mathit{any} \cdot \mathrm{NN} \cdot \mathit{any})^{*} \cdot \mathit{any})^{*} \cdot \mathit{any})^{*} \cdot \mathrm{NN} \cdot \mathit{any})^{*} \cdot \mathrm{NN}^{2})^{*} \cdot \mathit{any})^{*} \cdot \mathrm{NN} \cdot \mathit{any})^{*} \cdot \mathrm{DET} \cdot \mathrm{IS} \cdot \mathrm{DET})^{*} \cdot \mathit{any} \cdot (\mathit{any} \cdot \mathrm{NN} \cdot (\mathit{any} \cdot \mathrm{NN}^{2} \cdot (\mathit{any} \cdot \mathrm{NN} \cdot \mathit{any})^{*} \cdot \mathrm{IS} \cdot \mathrm{DET} \cdot \mathrm{IS} \cdot \mathrm{DET})^{*} \cdot \mathit{any})^{*} \cdot \mathit{any})^{*} \cdot \mathit{any})^{*})^{*} \cdot \mathit{any})^{*} \cdot \mathit{any} \cdot (\mathit{any} \cdot \mathrm{NN} \cdot \mathit{any})^{*} \cdot \mathit{any})^{*} \cdot \mathit{any})^{*} \cdot \mathrm{NN} \cdot \mathit{any})^{*} \cdot \mathrm{NN}^{2})^{*} \cdot \mathit{any})^{*} \cdot \mathrm{NN} \cdot \mathit{any})^{*} \cdot \mathrm{IS} \cdot \mathrm{DET} \cdot \mathrm{NN} \cdot \mathrm{DET} \cdot \mathrm{NN} \cdot \mathit{any})^{*} \cdot \mathrm{NN}^{2})^{*} \cdot \mathit{any})^{*} \cdot \mathrm{NN} \cdot \mathit{any})^{*} \cdot \mathrm{IS} \cdot \mathrm{DET} \cdot \mathrm{NN}$

5. Best individual for ExpM5; Satisfy all the following:

- IS · (DET + IS + ((((DET + IS) · *any*) + ((DET + (DET · (DET + (IS · DET$^2$)))) · DET)) · IS))
- ((DET · *any*) + MD) · *any* · (DET$^2$ + DET)
- IS · DET$^*$
- DET + IS
- (((DET + (IS · DET)) · *any*) + IS) · *any* · (DET + IS)
- DET
- DET + (IS · DET$^2$)

6. Best individual for ExpM6; Satisfy all the following:

   - NN·IS·((IS·(DET+((NN+(NN·IS·(DET+(NN·IS·DET·((IS·(NN+IS))+DET)))·DET))·IS$^2$·(DET+DET$^2$+(((IS·DET)+NN)·(DET+(IS·DET?)+((DET+IS)$^*$·IS·(IS+DET)))·IS·(DET+(NN·IS·((NN$^2$·IS·DET·(NN+(IS·(DET+NN))+IS)·DET·(DET+(NN·(DET+(NN·IS·(DET+(IS$^2$·NN)+IS)·IS·((IS·(NN+DET)·(DET+IS))+(NN·DET))·IS·NN)))))+NN)·NN·DET))))·NN·(DET+(NN·DET·(DET+(NN·IS·(DET+IS)·DET·IS·(DET+(NN·DET·NN·DET·NN))))·NN·(DET+(IS·DET)+(IS·(DET+(((IS·(DET+(DET·NN·IS·NN·IS$^3$·DET$^3$)+(NN·(DET+NN)))·NN)+((DET+NN)·DET)+DET)·(DET+IS+((DET+NN+(DET·IS·(DET+(NN·IS·(DET+(IS$^2$·NN$^3$·IS·NN))))·NN$^4$·IS$^2$·(DET+NN)·DET))·(NN+DET+(NN·IS·DET))·NN·((DET·NN)+NN+DET+((DET+(((IS·DET·(DET+IS)·IS$^2$·((IS·(DET+(((NN·(DET+IS)·NN)+DET)·NN)+(NN·IS$^2$·NN)))+DET+IS))+(((IS·NN)+IS)·(DET+IS)))·NN)+(NN·IS$^2$·NN))·IS·((IS·DET·NN)+IS)))))·NN)+(IS·(DET+NN+IS)))·NN)+(*any*·(DET+(DET·(DET+NN)·NN))))))))·((IS·(DET+(NN$^2$·DET·NN)+(DET·(DET+(IS$^2$·(DET+NN+(IS·(DET+(NN·IS·(NN+IS))))+(NN·DET))·(DET+NN+IS)·NN))·NN·(DET+((DET+NN)·DET·IS$^3$·(DET+(DET·IS$^2$·(DET+(DET·IS$^2$·DET·NN·IS·((IS·NN·((IS·((NN·DET)+IS))+NN)·(NN$^2$+IS)·NN$^2$)+DET+NN)·(DET+IS+(DET·NN))))·NN$^4$))·NN$^3$·(DET+NN)·(DET+(DET·NN))·(NN+DET$^2$)·(DET$^2$+(NN·IS$^2$))·NN·IS·(DET+(IS·DET))·DET))))·DET)+NN))+DET)·NN

7. Best individual for ExpM7; Satisfy all the following:
   - $\text{NN} \cdot \text{IS} \cdot \text{DET} \cdot \text{NN}$
   - $\textit{any}^2 \cdot \text{DET} \cdot \text{NN}$
   - $\text{IS} \cdot \text{DET} \cdot \text{NN}$
   - $\text{WHCH} + \text{DET}^*$
   - $\text{WHCH} + \text{IS}$
8. Best individual for ExpM8; Satisfy all the following:
   - $(\text{NN} + \text{DET}) \cdot \text{IS} \cdot ((\text{DET} \cdot \text{NN}) + \text{IS})$
   - $\text{DET}$
   - $\text{NN} \cdot \text{IS} \cdot (((\text{AJ} + \text{DET}) \cdot \text{NN}) + \text{DET})$
   - $\text{NN} \cdot \text{IS} \cdot \text{NN}^* \cdot ((\text{IS} \cdot (\text{IS} + \text{IS}^2 + \text{DET})) + \text{DET})$
   - $(\text{NN} \cdot \text{IS?} \cdot (\text{IS} \cdot (\text{IS} \cdot (((\text{IS} + \text{DET}) \cdot (\text{IS} + (\text{DET} \cdot \text{NN}) + (\text{NN} \cdot (\text{IS} + \text{DET})))) + \text{AJ})^*)^*)^*) + \text{AD}$
   - $\text{NN} \cdot \text{IS} \cdot (\text{AJ?} \cdot \text{DET})^* \cdot ((\text{IS} \cdot \text{NN}) + (\text{NN} \cdot ((\textit{any}^* \cdot \text{NN} \cdot \text{AJ}^* \cdot (\text{IS} \cdot (\text{IS} \cdot \textit{any})^*)^*) + \text{AJ} + \text{NN})))$
   - $(\text{NN} \cdot \text{IS?} \cdot (((\text{IS} \cdot \text{DET}^*)^* + \text{DET}) \cdot \text{IS})^* \cdot (\text{NN} \cdot \text{IS})^*) + (\text{NN} \cdot (\text{AJ} \cdot \textit{any}^*)^* \cdot ((\text{IS} \cdot (((\text{IS} \cdot \text{DET}^*)^* \cdot (\text{IS} \cdot \text{IS}^*)^*)^* + (\text{IS}^2)^* + (\text{NN} \cdot \text{IS}))^*) + \text{AD}) \cdot \text{DET})$
   - $(\text{IS} + (\text{IS} \cdot (\text{IS} + \text{IS}^*)) + \text{DET})^*$
   - $\text{IS} \cdot \text{DET?}$
9. Best individual for ExpM9; Satisfy all the following:
   - $\text{NN} \cdot \text{IS} \cdot \text{DET} \cdot ((\textit{any} \cdot ((\text{IS} \cdot \text{DET} \cdot \text{IS} \cdot \text{DET}) + (\text{IS} \cdot \text{NN})? + (\textit{any} \cdot \text{IS?})? + (\text{NN} \cdot \text{IS} \cdot \text{DET} \cdot \text{NN}) + (\text{NN} \cdot (\text{DET?} + (\textit{any} \cdot \text{IS} \cdot ((\text{WHCH?} \cdot \text{MD}) + (\text{NN} \cdot \text{IS} \cdot \text{DET} \cdot \text{NN}))? \cdot \text{NN}))? \cdot \text{NN})) \cdot \text{DET?} \cdot \text{NN})? + \text{NN})? \cdot \text{NN}$
10. Best individual for ExpM10; Satisfy all the following:
    - $\text{IS} \cdot \text{DET} \cdot \text{NN}$
    - $\text{NN} \cdot \text{IS}$
    - $\text{NN} \cdot \text{IS} \cdot \textit{any} \cdot \text{NN}$

## A.2 Verb Category

### A.2.1 Linguistic Objects with Verb Parts-of-speech

The following are the rules learnt by the best individuals from the GP having the following set of linguistic objects:

VBN, VBZ, VBD, VBG, VBP, VB, NOUN, ADJ, ADV, MOD, DET, FW, PREP, WHICH, TO

1. Best individual for VExp1; Satisfy all the following:
   - $(\mathrm{AD} + \mathrm{DET}) \cdot \mathrm{TO}$
   - $\mathrm{PREP} \cdot (\mathrm{PREP} + \mathrm{DET})$
   - $\mathrm{PREP}^* \cdot \mathrm{TO}$
   - DET
   - $\mathrm{AD}^* \cdot \mathit{any} \cdot \mathrm{AD}$
   - $\mathrm{TO}^* \cdot \mathit{any} \cdot \mathrm{AD}$
   - $\mathrm{PREP}^* \cdot \mathit{any}^2$
2. Best individual for VExp2; Satisfy all the following:
   - AD
   - AJ
   - $\mathit{any}^{20} \cdot \mathrm{TO}$
   - $(\mathrm{VBP} \cdot \mathit{any}^6)^* \cdot \mathit{any} \cdot \mathrm{TO}$
   - $\mathit{any}^8 \cdot \mathrm{TO}$
   - $\mathit{any}^{12} \cdot (\mathrm{VBP} \cdot \mathrm{AD})^* \cdot \mathit{any}^2 \cdot \mathit{any}^* \cdot \mathit{any}^{10} \cdot (\mathrm{VBP} \cdot \mathit{any}^6)^* \cdot \mathit{any}^{12} \cdot (\mathrm{VBP} \cdot \mathrm{AD})^* \cdot \mathit{any}^6$
   - PREP
   - WHCH
   - $\mathit{any}^{10} \cdot \mathrm{DET} \cdot \mathit{any} \cdot \mathit{any}^* \cdot \mathit{any}^{14} \cdot (\mathrm{VBP} \cdot \mathit{any}^6)^* \cdot \mathit{any}^7$

### A.2.2 Linguistic Objects with a Generic Verb Object

The following are the rules learnt by the best individuals from the GP having the following set of linguistic objects:

VB, NOUN, ADJ, ADV, MOD, DET, FW, PREP, WHICH, TO

1. Best individual for VExp3; Satisfy all the following:
   - $any \cdot \text{NN} \cdot any^2 \cdot \text{NN}$
   - PREP
   - $\text{NN}^*$
   - $\text{AD} \cdot (\text{TO} + \text{AD})$
   - $\text{NN} \cdot (\text{TO} + \text{WHCH})$
   - $\text{AJ} \cdot \text{NN}$
   - $\text{TO} \cdot any \cdot \text{NN}$
   - $\text{TO}^*$
   - $\text{AD} \cdot \text{AD}^*$
2. Best individual for VExp4; Satisfy all the following:
   - $\text{TO} \cdot (\text{NN} + (any \cdot \text{AD}) + \text{AJ})$
   - $(\text{NN} \cdot (\text{NN} + ((\text{NN} + (\text{TO} \cdot \text{AD}) + \text{AJ}) \cdot \text{AD}) + \text{AJ}) \cdot \text{NN}) + \text{AJ}$
   - $\text{NN} \cdot any^2 \cdot \text{NN}$
   - AD
   - NN?
   - $\text{TO} \cdot any \cdot \text{NN}$
   - AD?
   - $\text{NN} \cdot any \cdot \text{NN}$
   - $\text{PREP} \cdot any \cdot \text{NN}$

- $\mathrm{AJ?} \cdot (\mathrm{NN} + \mathrm{AD} + (\mathrm{TO} \cdot (\mathrm{AD} + \mathrm{NN} + (((\mathrm{TO} \cdot \mathrm{NN} \cdot \mathrm{TO}) + \mathrm{AJ}) \cdot (\mathrm{AJ} + \mathrm{PREP})) + (\mathrm{TO} \cdot (\mathrm{NN} + ((\mathrm{NN} + (\mathrm{TO} \cdot \mathrm{AD}) + \mathrm{AD} + \mathrm{PREP}) \cdot (\mathrm{AJ} + \mathrm{PREP})) + \mathit{any}) \cdot (\mathrm{NN} + \mathrm{TO} + \mathrm{AJ} + (\mathrm{TO} \cdot \mathrm{NN} \cdot (\mathrm{NN} + (\mathrm{AJ} \cdot ((\mathrm{TO} \cdot (\mathrm{NN} + (\mathrm{TO}^2 \cdot (\mathrm{NN} + \mathrm{AD} + \mathrm{PREP}) \cdot \mathrm{TO} \cdot \mathrm{NN}) + \mathit{any})) + \mathrm{AJ})) + \mathrm{AJ})))) + \mathrm{AJ} + (\mathrm{TO} \cdot \mathrm{NN} \cdot \mathrm{AJ?}) + (\mathrm{TO} \cdot \mathrm{AJ?}) + (\mathrm{TO} \cdot (\mathrm{NN} + \mathrm{AD} + \mathrm{TO} + \mathrm{PREP}) \cdot (\mathrm{NN} + (\mathrm{TO} \cdot \mathrm{NN} \cdot \mathrm{TO}) + \mathrm{AJ})))) + \mathrm{AJ} + \mathrm{PREP} + (\mathrm{TO} \cdot (\mathrm{NN} + \mathrm{PREP}))) \cdot (\mathrm{NN} + \mathrm{AJ})$
- $\mathrm{AD} + ((\mathrm{NN} + \mathrm{TO?} + (\mathrm{TO} \cdot \mathrm{NN} \cdot (\mathrm{NN} + (\mathrm{AJ} \cdot \mathit{any}))) + \mathrm{AJ}) \cdot \mathrm{NN?}) + \mathrm{AJ} + \mathrm{PREP} + \mathrm{TO}^2$
- $\mathrm{TO} \cdot (\mathrm{NN} + \mathrm{AD} + (\mathrm{AD} \cdot ((\mathrm{TO} \cdot (\mathrm{AD} + (\mathrm{TO} \cdot (\mathrm{NN} + (\mathrm{TO} \cdot (\mathrm{NN} + \mathrm{AD} + (\mathrm{TO} \cdot (\mathrm{AD} + (((\mathrm{AD} \cdot (\mathrm{AJ} + \mathrm{TO})) + \mathrm{AJ}) \cdot ((\mathrm{TO} \cdot \mathrm{AJ?}) + (\mathrm{TO} \cdot (\mathrm{NN} + \mathrm{AD} + (\mathrm{TO}^2 \cdot (\mathrm{NN} + (\mathrm{NN} + (\mathrm{TO} \cdot \mathrm{NN} \cdot (\mathrm{NN} + (\mathrm{TO} \cdot \mathrm{AD}) + \mathrm{AD})) + \mathrm{AJ})? + \mathrm{PREP})) + \mathrm{AJ} + \mathrm{PREP})))) + \mathrm{AJ} + \mathrm{PREP} + (\mathrm{TO} \cdot (\mathrm{NN} + \mathrm{AJ?} + \mathrm{AJ})) + (\mathrm{TO} \cdot (\mathrm{AJ} + (\mathrm{TO} \cdot \mathrm{AJ?}))))) + \mathrm{AJ} + \mathrm{PREP}) \cdot (\mathrm{AJ} + \mathrm{NN} + \mathrm{AD} + (\mathrm{TO} \cdot (\mathrm{NN} + (\mathrm{TO} \cdot (\mathrm{AD} + ((\mathrm{AD} + (\mathrm{TO} \cdot (\mathrm{NN} + \mathrm{AD})) + \mathrm{AJ}) \cdot \mathrm{TO}) + \mathrm{AJ} + \mathrm{PREP} + (\mathrm{TO} \cdot \mathrm{AJ?})?)) + \mathrm{AJ})) + \mathrm{PREP})) + \mathrm{AJ})) + \mathrm{AJ})) + (\mathrm{AD} \cdot \mathrm{AJ?}) + \mathrm{AJ})) + \mathrm{PREP} + (\mathrm{AD} \cdot (\mathrm{NN} + \mathrm{AD}^2 + \mathrm{AJ})) + \mathrm{AJ}) \cdot (\mathrm{NN} + ((\mathrm{AD} + (\mathrm{TO} \cdot (\mathrm{NN} + (\mathrm{TO} \cdot \mathrm{AJ?}) + \mathrm{AJ})) + ((\mathrm{NN} + \mathrm{AD} + ((\mathrm{AD} + \mathrm{NN} + (\mathrm{TO} \cdot (\mathrm{AJ} + \mathrm{PREP})) + (\mathrm{TO} \cdot (\mathrm{NN} + ((\mathrm{NN} + (\mathrm{TO} \cdot \mathrm{AD}) + \mathrm{AD} + \mathrm{PREP}) \cdot (\mathrm{AJ} + \mathrm{PREP})) + \mathit{any}) \cdot (\mathrm{NN} + \mathrm{TO} + \mathrm{AJ} + (\mathrm{TO} \cdot \mathrm{NN} \cdot (\mathrm{NN} + (\mathrm{AJ} \cdot ((\mathrm{TO} \cdot (\mathrm{NN} + (\mathrm{TO}^2 \cdot (\mathrm{NN} + \mathrm{AD} + \mathrm{PREP}) \cdot \mathrm{TO} \cdot \mathrm{NN}) + \mathit{any})) + \mathrm{AJ})) + \mathrm{AJ})))) + \mathrm{AJ} + (\mathrm{TO} \cdot \mathrm{NN} \cdot \mathrm{AJ?}) + (\mathrm{TO} \cdot \mathrm{AJ?}) + (\mathrm{TO} \cdot (\mathrm{NN} + \mathrm{AD} + \mathrm{TO} + \mathrm{PREP}) \cdot (\mathrm{NN} + (\mathrm{TO} \cdot \mathrm{NN} \cdot \mathrm{TO}) + \mathrm{AJ})) + \mathrm{PREP} + (\mathrm{TO} \cdot (\mathrm{NN} + ((\mathrm{NN} + (\mathrm{TO} \cdot (\mathrm{NN} + \mathrm{TO} + \mathrm{AJ})) + \mathrm{AJ} + \mathrm{PREP}) \cdot \mathrm{AJ?}) + \mathrm{AJ}))) \cdot (\mathrm{NN} + (\mathrm{TO} \cdot (\mathrm{AJ} + (\mathrm{TO} \cdot \mathrm{AJ?}))) + \mathrm{AJ})) + \mathrm{AJ} + \mathrm{PREP}) \cdot \mathrm{AJ?}) + \mathrm{PREP}) \cdot \mathrm{AD}) + \mathrm{AJ})$
- $\mathrm{AD} + \mathrm{AJ}$
- $\mathrm{NN} + \mathrm{PREP}$

### A.2.3 Linguistic Objects with Verb and Word Categories

The following are the rules learnt by the best individuals from the GP having the following set of linguistic objects:

VBN, VBZ, VBD, VBG, VBP, VB, NOUN, ADJ, ADV, MOD, DET, FW, PREP, WHICH, TO, MEAN, DEFINE, RELATE, CALL, CONSIST, KNOWN

1. Best individual for VExp5; Satisfy all the following:
    - $\text{CALL} + \text{MEAN} + \text{DEFINE}$
2. Best individual for VExp6; Satisfy all the following:
    - $\text{CALL?}$
    - $\text{CALL}$
    - $(\text{PREP} + (\text{NN} \cdot \textit{any})) \cdot ((\text{NN} \cdot \text{PREP}) + \text{CALL})$
    - $\text{NN} \cdot \text{PREP}$
    - $(\text{CALL} + (\text{NN} \cdot \textit{any})) \cdot ((\text{NN} \cdot \text{PREP}) + \text{NN} + \text{CALL})$
    - $(\text{CALL} + ((\text{NN} + \text{CALL}) \cdot \textit{any})) \cdot ((\text{NN} \cdot \text{PREP}) + \text{PREP} + \text{CALL})$
3. Best individual for VExp7; Satisfy all the following:
    - $\text{DEFINE} + ((\text{CALL} + \text{VBG}) \cdot \text{PREP}) + \text{CALL} + (\text{PREP} \cdot \textit{any}^* \cdot \text{CALL} \cdot \textit{any}) + (\textit{any} \cdot \text{PREP}) + (\text{DEFINE} \cdot \textit{any}^*)$
    - $\text{CALL} + \text{PREP}$
    - $(\textit{any} \cdot \text{PREP} \cdot \textit{any}^* \cdot \text{CALL} \cdot \textit{any}) + \text{DEFINE}$
    - $\text{PREP}$

## A.3 Punctuation Category

1. Best individual for PExp1; Satisfy all the following:
    - $\text{PREP}$
    - $\text{NN} \cdot \text{COLON}$
    - $\text{NN} \cdot \text{COLON?} \cdot \text{NN}$
    - $\text{NN} \cdot \text{COLON} \cdot \text{DET}$
    - $\text{DET}^* \cdot \text{NN?}$
    - $\text{DET?}$
2. Best individual for PExp2; Satisfy all the following:

- $(\text{DET} \cdot \text{NN}) + (\text{COLON} + \text{AJ}^*)^*$
- AJ?
- $\text{NN} \cdot (\textit{any} + \text{PREP}^*) \cdot \text{DET}$
- $\text{COLON} \cdot \textit{any}? \cdot \text{DET}$
- $\text{NN} \cdot (\text{PREP} + \text{NN})$
- $(\text{AJ} + \text{COLON}) \cdot \text{NN}$
- $(\text{AJ} + \text{COLON})^*$
- $\text{COLON} \cdot (\textit{any} + \text{PREP}^*) \cdot \text{DET}$
- $(\text{AJ} + \text{COLON}) \cdot \text{NN}$
- $\text{NN}^*$
- $\text{NN} \cdot (\textit{any} + (\text{AJ} + \text{COLON})^*) \cdot \text{PREP}$
- $\text{VERB}^*$
- $\text{AJ}^*$
- $\text{NN} \cdot (\text{COLON} + \text{PREP}^*) \cdot \text{DET}$
- $\text{NN} \cdot (\textit{any} + \text{PREP}^*) \cdot \text{PREP}$
- $\text{NN} \cdot (\textit{any} + \text{PREP}^*) \cdot \text{NN}$
- $(\text{COLON}? \cdot \text{DET})^*$
- $\text{NN} \cdot \textit{any}? \cdot \text{PREP}$
- $\text{PREP}^*$
- $\text{COLON} + \text{VERB}$
- $(\text{AJ} + \text{COLON}) \cdot \text{DET}$
- $\text{COLON} \cdot \text{DET}$
- $\text{DET}^*$
- COLON?
- $\text{NN} \cdot (\text{COLON} + \text{NN})$

## A.4 All Categories

1. Best individual for AExp1; Satisfy all the following:

    - $\mathrm{IS} \cdot \mathrm{DET} \cdot ((\mathit{any}^5 \cdot ((\mathit{any}^5 \cdot \mathrm{IS} \cdot \mathrm{DET} \cdot \mathit{any} \cdot \mathrm{DET} \cdot ((\mathit{any}^4 \cdot ((\mathit{any}^6 \cdot \mathrm{IS}) + \mathit{any}) \cdot \mathit{any}^5) + \mathit{any})) + \mathit{any}) \cdot \mathit{any}^4) + \mathit{any}) \cdot \mathit{any} \cdot ((\mathit{any}^3 \cdot \mathrm{IS} \cdot \mathit{any}^2) + \mathit{any})$
    - $\mathrm{IS} \cdot \mathit{any}^5$
    - $\mathit{any}^5 \cdot ((\mathit{any} \cdot (\mathit{any} + \mathit{any}^3) \cdot \mathrm{IS} \cdot \mathit{any}^5 \cdot \mathrm{IS}) + \mathit{any}) \cdot \mathit{any}^2$

2. Best individual for AExp2; Satisfy all the following:

    - $\mathrm{CALL} + (\mathrm{CALL} \cdot (\mathrm{DET} + ((\mathit{any} + (\mathrm{IS} \cdot \mathit{any}) + \mathrm{IS}) \cdot \mathit{any}) + \mathrm{IS}^2)) + ((\mathrm{CALL} + (\mathrm{IS} \cdot \mathrm{DET})) \cdot \mathit{any})$
    - $(\mathrm{CALL} + (\mathrm{IS} \cdot (\mathrm{CALL} + ((\mathrm{DET} + \mathrm{CALL} + ((\mathrm{DET} + ((\mathit{any} + \mathrm{CALL} + ((\mathrm{CALL} + (\mathrm{DET} \cdot (\mathrm{CALL} + ((\mathrm{CALL} + (\mathrm{IS} \cdot \mathrm{CALL} \cdot \mathrm{DET})) \cdot \mathit{any}) + ((\mathrm{CALL} + ((\mathrm{CALL} + \mathit{any}^2 + (\mathrm{IS} \cdot \mathrm{DET}) + (\mathrm{IS} \cdot \mathrm{CALL} \cdot \mathrm{DET})) \cdot \mathit{any}) + (\mathrm{IS} \cdot \mathrm{CALL})) \cdot \mathrm{DET})) \cdot \mathit{any})) \cdot ((\mathrm{CALL} \cdot \mathrm{DET}) + (\mathrm{IS} \cdot \mathrm{DET}) + (\mathrm{IS} \cdot (\mathit{any} + \mathrm{CALL} + (\mathrm{IS} \cdot (\mathrm{CALL} + ((\mathrm{DET} + ((\mathit{any} + \mathrm{CALL} + (\mathrm{IS} \cdot \mathit{any}) + \mathrm{IS}) \cdot \mathit{any}) + \mathrm{IS}^2 + (\mathrm{IS} \cdot \mathrm{CALL} \cdot (\mathrm{CALL} + (\mathrm{IS} \cdot (\mathit{any} + ((\mathrm{CALL} + \mathrm{IS}^2 + (\mathrm{CALL} \cdot \mathrm{DET})) \cdot \mathrm{DET})) \cdot \mathit{any})) \cdot \mathit{any}^3)) \cdot \mathit{any}))) + ((\mathrm{DET} + \mathrm{CALL} + ((\mathrm{DET} + ((\mathit{any} + \mathrm{CALL} + ((\mathrm{CALL} + (\mathrm{IS} \cdot \mathit{any}^2)) \cdot (\mathrm{CALL} + \mathit{any}^2 + (\mathrm{IS} \cdot \mathrm{DET}) + (\mathrm{IS} \cdot (\mathit{any} + \mathrm{CALL} + (\mathrm{IS} \cdot \mathit{any}) + (\mathrm{IS} \cdot \mathrm{DET}) + (\mathrm{IS} \cdot (\mathrm{IS} + \mathrm{CALL} + (\mathit{any} \cdot \mathrm{DET}) + (\mathrm{IS} \cdot \mathrm{CALL})) \cdot \mathit{any})) \cdot \mathit{any})) \cdot \mathit{any}) + \mathrm{IS}) \cdot \mathit{any}) + \mathrm{IS}^2 + (\mathrm{IS} \cdot \mathrm{CALL} \cdot \mathrm{DET})) \cdot \mathit{any}) + \mathrm{IS}^2 + (\mathrm{IS} \cdot \mathrm{CALL} \cdot \mathrm{DET})) \cdot \mathrm{DET}) + (\mathrm{IS} \cdot (\mathit{any} + (\mathrm{IS} \cdot \mathrm{CALL})) \cdot \mathit{any})) \cdot \mathit{any})) \cdot (\mathit{any} + \mathrm{CALL} + (\mathrm{IS} \cdot \mathrm{DET}) + (\mathrm{IS} \cdot (\mathrm{IS} + \mathrm{CALL} + (\mathrm{IS} \cdot \mathrm{DET}) + (\mathrm{IS} \cdot \mathrm{CALL})) \cdot \mathit{any})) \cdot \mathrm{CALL}) + \mathrm{IS}) \cdot \mathit{any}) + \mathrm{IS}^2 + (\mathrm{IS} \cdot \mathrm{CALL} \cdot \mathrm{DET})) \cdot \mathit{any}) + \mathrm{IS}^2 + (\mathrm{IS} \cdot \mathrm{CALL} \cdot \mathrm{DET})) \cdot \mathit{any}) + ((\mathrm{CALL} + \mathrm{IS} + (\mathrm{IS} \cdot \mathrm{DET})) \cdot \mathrm{DET})) \cdot \mathit{any})) \cdot (\mathrm{CALL} + \mathit{any}^2 + (\mathrm{IS} \cdot \mathrm{DET}) + (\mathrm{IS} \cdot (\mathrm{DET} + \mathrm{CALL} + ((\mathrm{DET} + ((\mathrm{CALL} + \mathrm{IS} + (\mathrm{CALL} \cdot \mathrm{DET}) + ((\mathit{any} + \mathrm{CALL} + (\mathrm{IS} \cdot \mathit{any}) + \mathrm{IS}) \cdot \mathit{any}) + ((\mathrm{CALL} + (\mathrm{IS} \cdot \mathit{any}^2)) \cdot (\mathrm{CALL} + \mathit{any}^2 + (\mathrm{IS} \cdot \mathrm{DET}) + (\mathrm{IS} \cdot (\mathit{any} + \mathrm{CALL} + (\mathrm{IS} \cdot \mathit{any}) + (\mathrm{IS} \cdot \mathrm{CALL}) + (\mathrm{IS} \cdot (\mathrm{IS} + ((\mathrm{CALL} + (\mathrm{IS} \cdot (\mathrm{CALL} + ((\mathrm{DET} + ((\mathrm{CALL} + ((\mathit{any} + (\mathrm{IS} \cdot \mathrm{CALL} \cdot \mathrm{DET})) \cdot \mathit{any}) + ((\mathrm{DET} + \mathrm{CALL} + ((\mathrm{DET} + ((\mathit{any} + \mathrm{CALL} + ((\mathrm{CALL} + ((((\mathrm{CALL} + (\mathrm{CALL} \cdot \mathit{any}) + (\mathrm{IS} \cdot \mathrm{DET}) + (\mathrm{IS} \cdot (\mathit{any} + \mathrm{CALL} + (\mathrm{IS} \cdot \mathit{any}) + (\mathrm{IS} \cdot \mathrm{DET}) + (\mathrm{IS} \cdot (\mathrm{IS} + \mathrm{CALL} + \mathrm{DET} + (\mathrm{IS} \cdot \mathrm{CALL})) \cdot \mathit{any})) \cdot \mathit{any})) \cdot \mathit{any}) + \mathrm{CALL} + (\mathrm{IS} \cdot \mathit{any}) + \mathrm{IS}) \cdot \mathit{any}) +$

$(\mathrm{IS}\cdot\mathrm{DET})+(\mathrm{IS}\cdot\mathrm{CALL}\cdot\mathrm{IS}\cdot(\mathit{any}+\mathrm{CALL}+\mathrm{IS}^2+(\mathrm{IS}\cdot\mathrm{DET})+(\mathrm{IS}\cdot\mathit{any}))\cdot$
$(\mathrm{DET}+(\mathrm{IS}\cdot\mathit{any})+\mathrm{IS}^2+(\mathrm{IS}\cdot\mathrm{CALL}\cdot\mathrm{DET}))\cdot\mathit{any}))\cdot\mathit{any})+((\mathrm{CALL}+(\mathrm{IS}\cdot$
$(\mathrm{CALL}+((\mathrm{DET}+((\mathit{any}+\mathrm{CALL}+(\mathrm{IS}\cdot\mathit{any})+\mathrm{DET}+((\mathit{any}+\mathrm{CALL}+(\mathrm{IS}\cdot$
$\mathit{any})+\mathrm{IS})\cdot\mathit{any}))\cdot\mathit{any})+(\mathrm{IS}\cdot\mathrm{CALL}\cdot\mathrm{DET}\cdot\mathrm{IS})+(\mathrm{IS}\cdot\mathrm{CALL}\cdot\mathrm{DET}))\cdot\mathit{any})+$
$((\mathrm{CALL}+(\mathrm{CALL}\cdot\mathit{any})+(\mathrm{IS}\cdot\mathrm{CALL}))\cdot\mathrm{DET}))\cdot\mathit{any}))\cdot(\mathrm{CALL}+((\mathit{any}+$
$((\mathrm{CALL}+\mathit{any}^2)\cdot\mathit{any})+\mathrm{IS})\cdot\mathit{any})+(\mathrm{IS}\cdot\mathrm{DET})+(\mathrm{IS}\cdot(\mathit{any}+\mathrm{CALL}+\mathrm{IS}^3+$
$(\mathrm{IS}\cdot(\mathrm{CALL}+(\mathrm{IS}\cdot\mathit{any})+(\mathrm{IS}\cdot(\mathrm{IS}+\mathrm{CALL}+(\mathrm{IS}\cdot\mathrm{DET}))\cdot\mathit{any})))+(\mathrm{IS}\cdot(\mathrm{IS}+$
$\mathrm{CALL}+(\mathrm{IS}\cdot\mathrm{DET})+(\mathrm{IS}\cdot\mathrm{CALL}))\cdot\mathit{any}))\cdot\mathit{any}))\cdot(\mathit{any}+\mathrm{CALL}+(\mathrm{IS}\cdot(\mathrm{DET}+$
$((\mathit{any}+\mathrm{CALL}+(\mathrm{IS}\cdot\mathit{any})+\mathrm{IS})\cdot\mathit{any})))+(\mathrm{IS}\cdot\mathrm{DET})+(\mathrm{IS}\cdot(\mathrm{IS}+\mathrm{CALL}+$
$((\mathrm{CALL}+((\mathrm{CALL}+(\mathrm{IS}\cdot\mathrm{DET})+\mathit{any}^2+\mathrm{IS})\cdot\mathrm{IS}\cdot\mathrm{DET})+(\mathrm{IS}\cdot\mathrm{DET})+\mathit{any})\cdot$
$\mathit{any})+(\mathrm{IS}\cdot\mathrm{CALL}))\cdot\mathit{any}))\cdot\mathrm{CALL})+\mathrm{IS})\cdot\mathit{any})+\mathrm{IS}^2+(\mathrm{IS}\cdot\mathrm{CALL}\cdot\mathrm{DET}))\cdot$
$\mathit{any})+\mathrm{IS}^2+(\mathrm{IS}\cdot\mathrm{CALL}\cdot\mathrm{DET}))\cdot\mathit{any}))\cdot\mathrm{DET})+((\mathrm{DET}+((\mathit{any}+\mathrm{CALL}+$
$((\mathrm{CALL}+(\mathrm{IS}\cdot(\mathit{any}+\mathrm{IS}+(\mathrm{IS}\cdot\mathrm{DET}))))\cdot(\mathrm{CALL}+\mathit{any}^2+(\mathrm{IS}\cdot\mathrm{DET})+(\mathrm{IS}\cdot$
$(\mathit{any}+\mathrm{CALL}+(\mathrm{IS}\cdot(\mathrm{CALL}+((\mathrm{DET}+(\mathrm{IS}\cdot\mathrm{DET})+\mathrm{IS}^2+(\mathrm{IS}\cdot\mathrm{CALL}\cdot\mathrm{DET}))\cdot$
$\mathit{any})))+(\mathrm{IS}\cdot\mathrm{DET})+(\mathrm{IS}\cdot(\mathrm{IS}+\mathrm{CALL}+(\mathrm{IS}\cdot\mathrm{DET})+(\mathrm{IS}\cdot\mathit{any}))\cdot\mathit{any}))\cdot\mathit{any}))\cdot$
$\mathrm{CALL}\cdot\mathrm{DET}\cdot\mathrm{CALL})+\mathrm{IS})\cdot\mathit{any})+\mathrm{IS}^2+(\mathrm{IS}\cdot\mathrm{CALL}\cdot\mathrm{DET}))\cdot\mathit{any})+\mathrm{IS}^2+$
$(\mathrm{IS}\cdot\mathrm{CALL}\cdot\mathrm{DET}))\cdot\mathit{any})+((\mathrm{CALL}+((\mathrm{CALL}+((\mathrm{CALL}+(\mathrm{IS}\cdot(\mathrm{CALL}+$
$((\mathrm{DET}+((\mathit{any}+\mathrm{CALL}+(\mathrm{IS}\cdot\mathrm{CALL})+\mathrm{IS})\cdot\mathit{any})+\mathrm{IS}^2+(\mathrm{IS}\cdot\mathrm{CALL}\cdot\mathrm{DET}))\cdot$
$\mathit{any})+((\mathrm{CALL}+((\mathrm{CALL}+((\mathit{any}+\mathrm{CALL}+(\mathrm{IS}\cdot\mathit{any})+\mathrm{IS})\cdot\mathit{any})+(\mathrm{IS}\cdot$
$\mathrm{DET})+(\mathrm{IS}\cdot\mathrm{CALL}\cdot\mathrm{DET}))\cdot\mathit{any})+(\mathrm{IS}\cdot\mathrm{DET}))\cdot\mathrm{DET}))\cdot\mathit{any}))\cdot(\mathrm{CALL}+$
$\mathit{any}^2+(\mathrm{IS}\cdot\mathrm{DET})+(\mathrm{IS}\cdot(\mathit{any}+\mathrm{DET}+((\mathit{any}+\mathrm{CALL}+((\mathrm{CALL}+(\mathrm{IS}\cdot$
$(\mathrm{CALL}+((\mathrm{DET}+((\mathit{any}+\mathrm{CALL}+(\mathrm{IS}\cdot\mathit{any})+\mathrm{IS})\cdot\mathit{any})+\mathrm{IS}^2+((\mathrm{CALL}+$
$((\mathrm{CALL}+((\mathit{any}+\mathrm{CALL}+(\mathrm{IS}\cdot\mathit{any})+\mathrm{IS})\cdot\mathit{any})+(\mathrm{IS}\cdot\mathrm{DET}))\cdot\mathit{any})+(\mathrm{IS}\cdot$
$\mathrm{DET}))\cdot\mathrm{DET})+(\mathrm{IS}\cdot\mathrm{CALL}))\cdot\mathit{any})+((\mathrm{CALL}+((\mathrm{CALL}+(\mathrm{IS}\cdot\mathrm{DET})+$
$(\mathrm{IS}\cdot\mathrm{CALL}\cdot\mathrm{DET}))\cdot\mathit{any})+\mathrm{IS})\cdot\mathrm{DET}))\cdot\mathrm{IS}\cdot\mathrm{CALL}))\cdot(\mathrm{CALL}+\mathit{any}^2+(\mathrm{IS}\cdot$
$\mathrm{DET})+(\mathrm{IS}\cdot(\mathrm{CALL}+(\mathrm{CALL}\cdot\mathit{any})+((\mathrm{CALL}+(\mathit{any}\cdot\mathrm{IS})+(\mathrm{IS}\cdot\mathrm{DET}))\cdot$
$\mathit{any})+(\mathrm{IS}\cdot\mathrm{DET})+(\mathrm{IS}\cdot(\mathit{any}+\mathrm{CALL}+((\mathrm{DET}+((\mathit{any}+\mathrm{CALL}+(\mathrm{IS}\cdot\mathit{any})+$
$\mathrm{IS})\cdot\mathit{any})+\mathrm{IS}^2+(\mathrm{IS}^2\cdot\mathrm{DET}))\cdot\mathit{any}))\cdot\mathit{any}))\cdot(\mathrm{CALL}+\mathit{any}^2+(\mathrm{IS}\cdot\mathrm{DET})+$
$(\mathrm{IS}\cdot(\mathit{any}+\mathrm{CALL}+(\mathrm{IS}\cdot(\mathrm{IS}+((\mathrm{DET}+\mathrm{CALL}+(\mathrm{IS}\cdot\mathrm{DET})+(\mathrm{IS}\cdot(\mathit{any}+$
$\mathrm{CALL}+(\mathrm{IS}\cdot\mathit{any}))\cdot\mathit{any}))\cdot\mathit{any})+(\mathrm{IS}\cdot\mathrm{DET}))\cdot\mathrm{DET})+(\mathrm{IS}\cdot\mathrm{DET})+(\mathrm{IS}\cdot(\mathrm{IS}+$
$\mathrm{CALL}+(\mathrm{IS}\cdot\mathrm{DET})+(\mathrm{IS}\cdot\mathrm{CALL}))\cdot\mathit{any}))\cdot\mathrm{CALL}))))\cdot\mathit{any})+\mathrm{IS})\cdot\mathit{any})+\mathrm{IS}+$
$(\mathrm{IS}\cdot\mathrm{CALL}\cdot\mathrm{DET})+(\mathrm{IS}\cdot\mathit{any})+\mathrm{CALL}+(\mathrm{IS}\cdot\mathrm{DET})+(\mathrm{IS}\cdot(\mathrm{IS}+\mathrm{CALL}+(\mathrm{IS}\cdot$
$\mathrm{DET})+(\mathrm{IS}\cdot\mathrm{CALL}))\cdot\mathrm{CALL}))\cdot\mathrm{CALL}))\cdot\mathit{any})+\mathrm{IS}+(\mathrm{IS}\cdot\mathrm{CALL}\cdot\mathrm{DET}))\cdot$

$\mathit{any})+\mathrm{DET}^2)\cdot\mathrm{DET}))\cdot\mathit{any}))\cdot(\mathrm{CALL}+\mathit{any}^2+(\mathrm{IS}\cdot\mathrm{DET})+(\mathrm{IS}\cdot(\mathrm{DET}+$
$\mathrm{CALL}+((\mathrm{DET}+((\mathit{any}+\mathrm{CALL}+((\mathrm{CALL}+(\mathrm{IS}\cdot\mathit{any}^2))\cdot(\mathrm{CALL}+\mathit{any}^2+$
$(\mathrm{IS}\cdot\mathrm{DET})+(\mathrm{IS}\cdot(\mathit{any}+\mathrm{CALL}+(\mathrm{IS}\cdot\mathit{any})+(\mathrm{IS}\cdot\mathrm{DET})+(\mathrm{IS}\cdot(\mathrm{IS}+\mathrm{CALL}+$
$(\mathit{any}\cdot\mathrm{DET})+(\mathrm{IS}\cdot\mathrm{CALL}))\cdot\mathit{any}))\cdot\mathit{any}))\cdot\mathit{any})+\mathrm{IS})\cdot\mathit{any})+\mathrm{IS}^2+(\mathrm{IS}\cdot\mathrm{CALL}\cdot$
$\mathrm{DET}))\cdot\mathit{any})+\mathrm{IS}^2+(\mathrm{IS}\cdot\mathrm{CALL}\cdot\mathrm{DET})+\mathrm{IS})\cdot\mathit{any}))\cdot\mathit{any})+(\mathrm{IS}\cdot\mathrm{CALL}))\cdot$
$\mathit{any}))\cdot\mathit{any}))\cdot\mathit{any}))\cdot\mathit{any})+\mathrm{IS}^2+(\mathrm{IS}\cdot\mathrm{CALL}\cdot\mathrm{DET}))\cdot\mathit{any})+\mathrm{IS}^2+(\mathrm{IS}\cdot\mathrm{CALL}\cdot$
$\mathrm{DET})+(\mathrm{IS}\cdot(\mathrm{IS}+\mathrm{CALL}+(\mathrm{IS}\cdot\mathrm{DET})+(\mathrm{IS}\cdot\mathrm{CALL}))\cdot\mathit{any}))\cdot\mathit{any}))\cdot\mathit{any}$

3. Best individual for AExp3; Satisfy all the following:

    - $\mathrm{IS}\cdot\mathit{any}\cdot((\mathrm{IS}\cdot(\mathrm{IS}\cdot\mathrm{DET}\cdot((\mathit{any}\cdot\mathrm{IS}\cdot\mathrm{DET}\cdot\mathit{any}\cdot\mathrm{IS}^2\cdot\mathit{any}\cdot((\mathrm{IS}\cdot\mathrm{DET}\cdot(\mathit{any}+$
      $\mathit{any}?)\cdot\mathit{any}^3\cdot\mathrm{DET}^2\cdot\mathit{any}\cdot\mathrm{IS}^2\cdot\mathrm{DET}?\cdot(\mathrm{IS}^2\cdot\mathit{any}^4\cdot\mathrm{DET}\cdot(\mathit{any}^3+\mathit{any})\cdot\mathit{any}^3\cdot$
      $\mathrm{DET}\cdot\mathit{any}?\cdot\mathit{any}\cdot\mathrm{IS}^2\cdot(\mathrm{DET}+(\mathrm{DET}\cdot\mathit{any}?))\cdot\mathit{any}^9)?\cdot\mathit{any}^3)?\cdot\mathit{any}^2)?)+\mathit{any})\cdot$
      $\mathit{any}^3\cdot\mathrm{DET}\cdot\mathit{any}\cdot((\mathit{any}\cdot\mathrm{IS}\cdot(\mathit{any}^2+\mathit{any})\cdot\mathit{any}\cdot\mathrm{DET}\cdot\mathit{any}^2)+\mathit{any}?)\cdot\mathrm{IS}\cdot\mathrm{DET}\cdot$
      $\mathit{any})?\cdot\mathit{any}^6\cdot\mathrm{DET}\cdot\mathit{any}?\cdot\mathit{any}\cdot\mathrm{IS}\cdot\mathit{any}^2\cdot\mathit{any}^6?\cdot\mathit{any}\cdot\mathrm{IS}\cdot\mathit{any}\cdot\mathrm{DET}\cdot\mathrm{IS}\cdot\mathrm{DET}\cdot$
      $\mathrm{IS}\cdot\mathit{any}\cdot(\mathit{any}^3?\cdot\mathrm{IS}\cdot\mathit{any}\cdot((\mathit{any}\cdot(\mathit{any}\cdot(((\mathit{any}^3+\mathit{any})\cdot\mathit{any})?\cdot\mathit{any}^2)?\cdot\mathrm{IS}\cdot\mathrm{DET}\cdot$
      $(\mathit{any}^3+\mathit{any})\cdot\mathit{any}^3)?\cdot\mathit{any}\cdot(\mathit{any}^3+\mathit{any})\cdot\mathit{any}^2\cdot\mathrm{IS}\cdot\mathrm{DET}?\cdot\mathit{any}^5\cdot\mathrm{DET}\cdot(\mathrm{DET}\cdot$
      $\mathit{any}?)?\cdot\mathit{any}\cdot\mathrm{IS}^2\cdot\mathrm{DET}?\cdot\mathit{any}^3\cdot(((\mathrm{IS}\cdot\mathit{any}\cdot(\mathit{any}^3\cdot\mathrm{DET}?\cdot((\mathrm{DET}?\cdot\mathrm{DET}\cdot$
      $(\mathit{any}+(\mathrm{IS}\cdot\mathrm{DET}\cdot(\mathrm{DET}\cdot((\mathit{any}\cdot\mathrm{IS}\cdot\mathit{any})+\mathit{any}))?\cdot\mathrm{IS}\cdot\mathit{any}))\cdot\mathit{any}\cdot\mathrm{IS}^2\cdot(\mathrm{DET}+$
      $\mathrm{IS})\cdot\mathit{any}^9)+\mathrm{IS})\cdot\mathit{any}^2)?\cdot\mathit{any}^3)+(\mathrm{IS}\cdot\mathit{any}\cdot\mathit{any}^3?))\cdot\mathit{any}\cdot\mathrm{IS}\cdot\mathit{any}^4)?\cdot\mathrm{IS}\cdot\mathit{any}^4)?\cdot$
      $\mathit{any}^2)?\cdot\mathit{any}^4)?\cdot\mathit{any}^3\cdot\mathrm{DET}\cdot\mathit{any}^5\cdot\mathrm{IS}\cdot\mathit{any}^4)?\cdot\mathrm{IS}\cdot\mathit{any}\cdot(\mathrm{IS}\cdot\mathrm{DET}\cdot\mathit{any}^2\cdot\mathrm{IS}\cdot\mathit{any}\cdot$
      $((\mathit{any}^3\cdot(\mathit{any}?\cdot\mathit{any}^2)?\cdot\mathit{any}^3\cdot\mathrm{DET}^2\cdot\mathit{any}\cdot\mathrm{IS}^2\cdot\mathrm{DET}\cdot\mathit{any}^3\cdot((\mathrm{IS}\cdot\mathrm{DET}\cdot(\mathit{any}^3+$
      $\mathit{any})\cdot\mathrm{DET}\cdot(\mathit{any}^3+\mathit{any})\cdot\mathit{any}^2\cdot\mathrm{DET}\cdot\mathit{any}^2\cdot\mathrm{DET}\cdot((\mathit{any}\cdot((\mathrm{DET}\cdot\mathrm{IS}^2\cdot((\mathrm{IS}\cdot$
      $\mathrm{DET}\cdot((\mathit{any}\cdot\mathrm{IS}\cdot\mathit{any}\cdot\mathrm{IS}^2\cdot\mathit{any}\cdot((\mathrm{IS}\cdot\mathrm{DET}\cdot(\mathit{any}^3+\mathit{any}?)\cdot\mathit{any}^3\cdot\mathrm{DET}^2\cdot\mathit{any}\cdot$
      $\mathrm{IS}\cdot\mathit{any})?\cdot(\mathit{any}^3+\mathit{any})\cdot\mathit{any})?)+\mathit{any})\cdot\mathit{any}^3\cdot\mathrm{DET}\cdot\mathit{any}^2\cdot\mathrm{IS}\cdot\mathrm{DET}\cdot\mathit{any})+$
      $\mathrm{DET})\cdot\mathit{any}^6\cdot(\mathit{any}^3+\mathit{any}))?\cdot\mathit{any}^2)?)+\mathit{any})\cdot\mathit{any}^3)?\cdot\mathit{any}\cdot\mathrm{IS}\cdot(\mathrm{DET}+(\mathrm{IS}^2\cdot$
      $\mathit{any}\cdot\mathit{any}^2?\cdot\mathit{any}^3\cdot\mathrm{IS}?\cdot(\mathit{any}^3+\mathit{any})))\cdot(\mathrm{IS}\cdot\mathit{any}\cdot((\mathrm{DET}\cdot((\mathrm{IS}\cdot\mathrm{DET}\cdot((\mathrm{IS}^2\cdot\mathit{any}^4\cdot$
      $(\mathrm{DET}+\mathrm{IS})\cdot\mathit{any}^9)+\mathit{any}))+\mathit{any})\cdot\mathrm{IS}^2\cdot\mathrm{DET}?\cdot\mathit{any}^3\cdot\mathrm{IS}\cdot\mathrm{DET}?\cdot\mathit{any}^4\cdot(\mathit{any}?\cdot$
      $\mathit{any}\cdot((\mathrm{DET}\cdot\mathit{any})?\cdot\mathit{any}^2)?\cdot\mathit{any}^3\cdot\mathrm{DET}\cdot\mathit{any}?\cdot\mathit{any}\cdot\mathrm{IS}^2\cdot\mathrm{DET}?\cdot\mathit{any}^3\cdot((\mathrm{IS}\cdot$
      $\mathrm{DET}\cdot((\mathrm{IS}^2\cdot\mathrm{DET}?\cdot(\mathrm{IS}\cdot\mathrm{DET}\cdot\mathit{any}^4\cdot\mathrm{DET}\cdot\mathit{any}?\cdot(((\mathit{any}^2\cdot\mathrm{DET}\cdot\mathit{any}?\cdot\mathit{any}\cdot\mathrm{IS}\cdot$
      $\mathit{any}\cdot\mathrm{DET}?\cdot\mathit{any}^5)+\mathit{any})\cdot\mathit{any}^2)?\cdot\mathrm{IS}^2\cdot\mathit{any}^4\cdot\mathrm{DET}?\cdot\mathit{any}^9)?\cdot\mathit{any}\cdot(\mathrm{IS}\cdot\mathrm{DET}\cdot$
      $(\mathit{any}^3+\mathit{any})\cdot\mathit{any}\cdot((\mathit{any}^3\cdot(((\mathit{any}^5\cdot\mathrm{DET})+\mathit{any})\cdot\mathit{any}^2)?\cdot\mathit{any})+\mathit{any}))?\cdot$
      $\mathit{any}^3)+\mathit{any})\cdot\mathrm{DET}\cdot(\mathit{any}^3+\mathit{any})\cdot\mathit{any}^2\cdot\mathrm{DET}\cdot\mathit{any}?\cdot\mathrm{IS}\cdot\mathit{any}^2\cdot\mathrm{DET}\cdot((\mathit{any}\cdot$
      $((((\mathrm{IS}\cdot\mathrm{DET}\cdot\mathit{any}\cdot\mathrm{IS}\cdot\mathit{any}^6\cdot\mathrm{IS}\cdot\mathit{any}^4)?\cdot\mathit{any}^2)?\cdot\mathit{any}\cdot\mathrm{IS}^2\cdot\mathrm{DET}?\cdot\mathit{any}^6\cdot\mathrm{IS}\cdot(\mathrm{IS}\cdot$

$\mathrm{DET}\cdot((\mathit{any}\cdot\mathrm{IS}\cdot\mathrm{DET}\cdot\mathit{any}\cdot\mathrm{IS}^2\cdot\mathit{any}\cdot((\mathrm{IS}\cdot\mathrm{DET}\cdot(\mathit{any}^3+\mathit{any})\cdot\mathit{any}^3\cdot\mathrm{DET}^2\cdot$
$\mathit{any}\cdot\mathrm{IS}^2\cdot\mathrm{DET}?\cdot(\mathrm{IS}\cdot\mathit{any}\cdot\mathrm{IS}\cdot\mathit{any}^4\cdot\mathrm{DET}\cdot(\mathit{any}^3+\mathit{any})\cdot\mathit{any}^3\cdot\mathrm{DET}\cdot\mathit{any}?\cdot\mathit{any}\cdot$
$\mathrm{IS}^2\cdot(\mathrm{DET}+(\mathrm{DET}\cdot\mathit{any}?))\cdot\mathit{any}^{11})?\cdot\mathit{any}^3)?\cdot(\mathit{any}^3+\mathit{any})\cdot\mathit{any})?)+\mathit{any})\cdot$
$\mathit{any}^3\cdot\mathrm{DET}\cdot\mathit{any}^2\cdot\mathrm{IS}\cdot\mathrm{DET}\cdot\mathit{any})?\cdot\mathit{any}^5\cdot\mathrm{IS}\cdot\mathrm{DET}\cdot(\mathit{any}+(\mathrm{IS}\cdot\mathrm{DET}\cdot(\mathrm{IS}\cdot(\mathrm{DET}\cdot$
$(\mathrm{IS}\cdot\mathit{any}\cdot\mathrm{IS}\cdot(\mathrm{IS}\cdot\mathit{any}\cdot((\mathrm{DET}\cdot\mathrm{IS}^2\cdot\mathrm{DET}?\cdot\mathrm{DET}\cdot\mathit{any}^5\cdot\mathrm{IS}\cdot\mathit{any}^2\cdot\mathrm{IS}\cdot\mathit{any}\cdot(\mathit{any}^3?\cdot$
$\mathit{any}^2)?\cdot\mathit{any}^2)?\cdot\mathit{any}^2)?\cdot\mathit{any}^4)?\cdot(\mathit{any}^3\cdot\mathrm{DET}\cdot(\mathrm{IS}\cdot\mathit{any})?\cdot(\mathit{any}^3+\mathit{any})\cdot\mathit{any}^3\cdot$
$\mathrm{DET}\cdot\mathit{any}?\cdot\mathit{any}\cdot\mathrm{IS}\cdot\mathrm{NN}\cdot\mathit{any}^4\cdot\mathrm{DET}?\cdot\mathit{any}^2\cdot((((\mathit{any}^3\cdot\mathrm{DET})+\mathit{any})\cdot\mathit{any}^2)+$
$\mathit{any})\cdot\mathit{any})?\cdot\mathit{any}^3)?\cdot\mathit{any})?\cdot((\mathrm{IS}\cdot\mathit{any}^3\cdot(\mathit{any}^3+\mathit{any}))?\cdot\mathit{any}^4\cdot\mathrm{DET}\cdot\mathit{any}?\cdot\mathit{any}\cdot$
$\mathrm{IS}^2\cdot\mathrm{DET}?\cdot\mathit{any}^3\cdot((\mathit{any}^2\cdot\mathrm{DET}\cdot\mathit{any}?\cdot\mathrm{IS}\cdot\mathit{any}\cdot\mathrm{DET}\cdot((\mathit{any}\cdot((\mathrm{IS}\cdot\mathit{any}\cdot\mathrm{DET}\cdot$
$\mathit{any}?\cdot\mathit{any}^5\cdot\mathrm{IS}\cdot(\mathrm{IS}\cdot\mathrm{DET}\cdot((\mathit{any}\cdot\mathrm{IS}\cdot\mathrm{DET}\cdot\mathit{any}\cdot\mathrm{IS}^2\cdot\mathit{any}\cdot((\mathrm{IS}\cdot\mathrm{DET}\cdot(\mathit{any}^3+$
$(\mathrm{IS}\cdot\mathrm{DET}\cdot(\mathit{any}^3+\mathit{any})\cdot\mathit{any}))\cdot\mathit{any}^3\cdot\mathrm{DET}^2\cdot\mathit{any}\cdot\mathrm{IS}^2\cdot\mathrm{DET}?\cdot(\mathrm{IS}\cdot\mathit{any}\cdot\mathrm{IS}\cdot$
$\mathit{any}^4\cdot\mathrm{DET}\cdot(\mathit{any}^3+\mathit{any})\cdot\mathit{any}^3\cdot\mathrm{DET}\cdot\mathit{any}?\cdot(\mathrm{IS}\cdot\mathit{any}^6\cdot\mathrm{IS}\cdot(\mathrm{IS}\cdot\mathrm{DET}\cdot\mathit{any}^3)?\cdot$
$\mathit{any}^2\cdot\mathrm{IS}\cdot\mathit{any}^5)?\cdot\mathrm{IS}^2\cdot(\mathrm{DET}+(\mathrm{DET}\cdot\mathit{any}?))\cdot\mathit{any}^6\cdot\mathit{any}^*\cdot\mathit{any}\cdot\mathrm{IS}\cdot\mathrm{DET}\cdot$
$(\mathit{any}^3+\mathit{any})\cdot\mathrm{IS}\cdot\mathit{any}^6)?\cdot\mathit{any}^3)?\cdot(\mathit{any}^3+\mathit{any})\cdot\mathit{any}^2)?)+\mathit{any})\cdot\mathit{any}^3\cdot\mathrm{DET}\cdot$
$\mathit{any}^2\cdot\mathrm{IS}\cdot\mathrm{DET}\cdot\mathit{any})?\cdot\mathit{any}^6\cdot\mathrm{DET}\cdot\mathit{any}^2\cdot(\mathit{any}^3+\mathit{any}))?\cdot\mathit{any}^2)?)+\mathit{any})\cdot$
$\mathit{any}\cdot\mathrm{IS}^2\cdot\mathit{any}^2)?\cdot\mathit{any}\cdot\mathrm{IS}\cdot(\mathrm{DET}+(\mathrm{IS}^2\cdot\mathit{any}\cdot(\mathit{any}\cdot\mathrm{DET})?\cdot\mathit{any}\cdot\mathrm{IS}\cdot\mathit{any}\cdot((\mathrm{IS}^2\cdot$
$\mathrm{DET}?\cdot\mathit{any}\cdot\mathrm{IS}\cdot((\mathit{any}^2+\mathit{any})\cdot\mathit{any}^2)?\cdot\mathit{any}^5\cdot\mathrm{IS}\cdot\mathrm{DET}?\cdot\mathit{any}^5\cdot((\mathit{any}\cdot((\mathrm{DET}?\cdot$
$((\mathrm{DET}\cdot\mathit{any}^2\cdot\mathrm{IS}\cdot\mathrm{DET}?\cdot\mathit{any}^2)+\mathit{any})\cdot\mathit{any}\cdot(\mathit{any}+(\mathrm{IS}\cdot\mathit{any}^3))\cdot(\mathit{any}^3+\mathit{any})\cdot$
$\mathit{any}\cdot((\mathit{any}^2\cdot\mathrm{DET}\cdot\mathit{any}^2)+\mathit{any})\cdot\mathit{any}^3\cdot\mathrm{DET}\cdot\mathit{any}?\cdot\mathit{any}\cdot\mathrm{IS}^2\cdot\mathrm{DET}?\cdot\mathit{any}^4\cdot$
$\mathrm{IS}\cdot\mathrm{DET}\cdot\mathit{any}^3\cdot\mathrm{DET}\cdot\mathit{any}?\cdot\mathit{any}^5)?\cdot\mathit{any}^2)?\cdot\mathrm{DET}?\cdot(\mathit{any}+(\mathrm{IS}\cdot\mathrm{DET}\cdot\mathit{any}))\cdot$
$\mathit{any})+\mathit{any})\cdot\mathit{any}^2\cdot\mathrm{IS}\cdot((\mathrm{DET}\cdot\mathrm{IS}\cdot\mathit{any}^2\cdot\mathrm{DET})?\cdot\mathit{any}^2)?\cdot\mathit{any}^3\cdot\mathrm{DET}?\cdot\mathit{any}\cdot\mathrm{IS}\cdot$
$\mathrm{DET}?\cdot\mathit{any}^6\cdot\mathit{any}?\cdot\mathit{any})?\cdot\mathit{any}\cdot(\mathrm{IS}\cdot\mathrm{DET}\cdot((\mathrm{DET}\cdot\mathit{any}^2)+\mathit{any})\cdot\mathit{any}^3\cdot\mathrm{IS}\cdot\mathit{any}^2\cdot$
$\mathrm{IS}^2\cdot\mathrm{DET}?\cdot\mathit{any}^9)?\cdot\mathit{any}^2)?\cdot\mathit{any}\cdot\mathrm{IS}?\cdot(\mathit{any}^3+(\mathit{any}\cdot\mathrm{DET}\cdot\mathit{any})+\mathit{any})))\cdot(\mathrm{IS}\cdot$
$\mathit{any}\cdot(((\mathit{any}^4\cdot\mathrm{DET}\cdot\mathit{any}\cdot\mathrm{DET}\cdot\mathit{any}?\cdot\mathrm{IS}^2\cdot\mathrm{DET}?\cdot\mathit{any}^4\cdot\mathrm{IS}\cdot\mathit{any}^4)+(((\mathit{any}^3\cdot$
$\mathrm{DET})+\mathit{any})\cdot\mathit{any}^2))\cdot\mathit{any}^2)?\cdot\mathit{any}\cdot\mathrm{DET}?\cdot\mathit{any}\cdot\mathrm{IS}\cdot(\mathit{any}?+\mathit{any})\cdot\mathit{any}^5)?\cdot$
$\mathit{any}^2)?\cdot\mathrm{IS}\cdot\mathit{any}^4)?\cdot\mathrm{IS}\cdot\mathrm{DET}\cdot\mathit{any}^6)?))\cdot\mathit{any}\cdot(\mathit{any}^3+\mathit{any}))?\cdot\mathit{any}^2)?)+\mathit{any})\cdot$
$\mathit{any}\cdot\mathrm{IS}^2\cdot\mathit{any}^2)?\cdot\mathit{any}\cdot\mathrm{IS}\cdot\mathit{any}\cdot(\mathrm{DET}\cdot\mathit{any}\cdot\mathrm{DET}\cdot\mathit{any}?\cdot\mathit{any}\cdot\mathrm{IS}^2\cdot\mathrm{DET}?\cdot((\mathit{any}\cdot$
$\mathrm{DET}\cdot\mathit{any}?\cdot\mathrm{IS}\cdot\mathrm{DET}\cdot((\mathit{any}\cdot\mathrm{IS}\cdot\mathit{any}\cdot\mathrm{IS}\cdot(\mathit{any}^4+\mathit{any})\cdot\mathit{any}\cdot\mathrm{DET}\cdot\mathit{any}^4\cdot\mathrm{IS}\cdot\mathit{any}\cdot$
$(\mathit{any}\cdot(\mathit{any}\cdot(((\mathit{any}^3+\mathit{any})\cdot\mathit{any})?\cdot\mathit{any}^2)?\cdot\mathrm{IS}\cdot\mathrm{DET}\cdot(\mathit{any}^3+\mathit{any})\cdot\mathit{any}^3)?\cdot$
$\mathit{any}\cdot(\mathit{any}^3+\mathit{any})\cdot\mathit{any}\cdot(\mathit{any}^2\cdot(((\mathrm{IS}\cdot((\mathit{any}^2\cdot\mathrm{IS}\cdot\mathrm{DET}?\cdot\mathit{any}^2)+\mathit{any})\cdot((((\mathrm{IS}\cdot$
$\mathrm{DET}\cdot(\mathit{any}^3+\mathit{any})\cdot\mathrm{IS}\cdot\mathit{any}^2\cdot\mathrm{IS}^2\cdot\mathit{any}^6\cdot((\mathit{any}^3\cdot\mathrm{DET})+\mathit{any})\cdot\mathit{any}\cdot\mathrm{DET}\cdot\mathrm{IS}\cdot$
$\mathrm{DET}?\cdot\mathit{any}^5\cdot\mathrm{DET}\cdot\mathit{any}^4\cdot(\mathit{any}^5+\mathit{any}^3+\mathit{any}))+\mathit{any})\cdot\mathit{any}^2)+\mathrm{DET})\cdot\mathit{any}^2\cdot$
$\mathrm{IS}\cdot\mathit{any}^4)+\mathit{any})\cdot\mathit{any}^2)?\cdot\mathit{any})?)?)+\mathit{any})\cdot\mathit{any}^4\cdot\mathrm{DET}\cdot(\mathrm{IS}^2\cdot\mathit{any}^5)?\cdot\mathit{any}\cdot\mathrm{IS}\cdot$

$\mathit{any}^4\cdot\mathrm{DET}\cdot(\mathrm{IS}\cdot\mathit{any}\cdot((\mathit{any}^4\cdot\mathrm{DET}\cdot\mathit{any}?\cdot\mathit{any}\cdot\mathrm{IS}^2\cdot(\mathrm{DET}+(\mathrm{IS}\cdot\mathit{any}\cdot(\mathrm{IS}?\cdot$
$\mathrm{IS}\cdot\mathit{any}\cdot((\mathit{any}?\cdot\mathit{any}\cdot\mathrm{DET})+\mathrm{DET})\cdot\mathit{any}^3)?\cdot\mathit{any}^2))\cdot\mathit{any}^6\cdot\mathrm{DET}\cdot\mathit{any}^4)?\cdot$
$\mathit{any}^2)?\cdot\mathit{any}^4\cdot\mathrm{DET}\cdot(\mathrm{IS}\cdot\mathrm{DET}\cdot(\mathit{any}+(\mathrm{IS}\cdot\mathit{any}\cdot((\mathit{any}^2\cdot\mathrm{IS}\cdot(\mathrm{IS}\cdot\mathit{any}\cdot((\mathit{any}^4\cdot$
$\mathrm{DET}\cdot\mathit{any}?\cdot\mathit{any}\cdot\mathrm{IS}^2\cdot\mathit{any}\cdot\mathrm{DET}?\cdot\mathit{any}^4\cdot\mathrm{IS}\cdot\mathit{any}^6\cdot\mathrm{DET}\cdot\mathit{any}^4)?\cdot\mathit{any}^2)?)?\cdot$
$((\mathrm{IS}^2\cdot\mathit{any}^3\cdot\mathrm{DET}\cdot(\mathit{any}+(\mathrm{IS}\cdot\mathrm{DET}?\cdot\mathrm{IS}^2\cdot\mathrm{DET}?\cdot\mathrm{IS}\cdot\mathrm{DET}\cdot\mathit{any}^2\cdot\mathrm{IS}\cdot\mathit{any}))\cdot$
$\mathit{any}\cdot\mathrm{IS}^2\cdot\mathrm{DET}?\cdot\mathit{any}^9)+\mathrm{IS})\cdot\mathit{any}^2)+\mathrm{DET})\cdot\mathit{any})))?\cdot\mathrm{IS}\cdot\mathrm{DET}?\cdot\mathit{any}^4)?\cdot$
$\mathit{any})+\mathit{any})\cdot\mathit{any}^3)?\cdot\mathit{any}^2)?\cdot\mathrm{IS}\cdot\mathit{any}^4)?\cdot\mathit{any}^3)?\cdot\mathit{any}^2)?\cdot\mathrm{DET}\cdot\mathit{any}?\cdot\mathit{any}\cdot\mathrm{IS}\cdot$
$(\mathit{any}?+\mathit{any})\cdot\mathit{any}^5)?\cdot\mathit{any}^2)?\cdot\mathrm{IS}\cdot\mathit{any}^4)?\cdot\mathit{any}\cdot\mathrm{DET}\cdot\mathit{any}^4)?\cdot\mathit{any}^3)?\cdot\mathit{any}^2)?$

- $\mathrm{IS}\cdot\mathrm{DET}\cdot(\mathrm{IS}\cdot\mathit{any}\cdot(\mathit{any}^3+\mathit{any}))?$

- $\mathit{any}\cdot((\mathit{any}\cdot\mathrm{IS}\cdot\mathrm{DET}\cdot(\mathit{any}\cdot\mathrm{DET}\cdot(\mathit{any}^3+\mathit{any}))?\cdot(\mathit{any}^3+((((\mathit{any}^3\cdot(\mathit{any}^3+$
  $\mathit{any}))?\cdot\mathrm{IS}\cdot((\mathit{any}^3\cdot(\mathit{any}+(\mathrm{DET}\cdot(\mathrm{IS}\cdot\mathrm{DET}\cdot(\mathit{any}^3+\mathit{any}+((\mathrm{IS}\cdot\mathrm{DET}\cdot$
  $(\mathit{any}^3+\mathit{any})\cdot\mathrm{DET}\cdot(\mathit{any}^3+\mathit{any})\cdot\mathit{any}^2\cdot\mathrm{DET}\cdot\mathit{any}?\cdot\mathrm{IS}^2\cdot\mathit{any}^5\cdot\mathrm{DET}\cdot((\mathit{any}^2\cdot$
  $\mathrm{IS}\cdot\mathrm{DET}?\cdot\mathit{any}^4)+\mathit{any})\cdot\mathit{any}^4\cdot\mathrm{IS}\cdot\mathit{any}^2\cdot\mathrm{IS}\cdot\mathit{any}^2\cdot\mathrm{DET}\cdot\mathit{any}?\cdot\mathit{any}\cdot\mathrm{IS}\cdot\mathit{any}\cdot$
  $\mathrm{IS}^2\cdot\mathit{any})?\cdot\mathit{any}^2)))?))\cdot\mathit{any}^2)+\mathit{any})\cdot((((\mathrm{IS}\cdot\mathit{any}^2\cdot\mathrm{DET}\cdot\mathit{any}\cdot((\mathit{any}^2\cdot\mathrm{IS}\cdot$
  $\mathrm{DET}?\cdot\mathit{any}^2)+(\mathrm{IS}\cdot\mathrm{DET}?\cdot\mathrm{IS}^2\cdot\mathrm{DET}?\cdot\mathrm{IS}\cdot(\mathrm{DET}+(\mathrm{DET}\cdot\mathit{any}?))\cdot\mathit{any}^6\cdot\mathrm{IS}\cdot$
  $\mathit{any}^2)+\mathit{any}+\mathit{any}^2))+\mathit{any}?)\cdot\mathit{any}^2)+\mathrm{DET})\cdot\mathit{any}^4\cdot\mathrm{DET}\cdot\mathit{any}?\cdot\mathit{any}\cdot\mathrm{IS}^2\cdot$
  $\mathrm{DET}?\cdot\mathit{any}^3\cdot((\mathrm{DET}\cdot\mathit{any}\cdot\mathrm{IS}^2\cdot\mathit{any}^2)?\cdot\mathit{any}\cdot\mathrm{IS}\cdot(\mathrm{DET}+(\mathrm{IS}^2\cdot\mathit{any}\cdot(\mathit{any}\cdot$
  $\mathrm{DET})?\cdot\mathit{any}^3\cdot\mathrm{IS}?\cdot(\mathit{any}^3+\mathit{any})))\cdot(\mathrm{IS}\cdot\mathit{any}\cdot(((\mathit{any}^4\cdot\mathrm{DET}\cdot\mathit{any}\cdot\mathrm{DET}\cdot\mathit{any}?\cdot$
  $\mathrm{IS}\cdot\mathit{any}?\cdot\mathit{any}\cdot\mathrm{IS}\cdot\mathit{any}^4)+(((\mathit{any}^3\cdot\mathrm{DET})+\mathit{any})\cdot\mathit{any}^2))\cdot\mathit{any}^2)?\cdot\mathit{any}^2\cdot\mathrm{IS}\cdot$
  $(\mathit{any}?+\mathit{any})\cdot\mathit{any}\cdot\mathrm{IS}\cdot\mathit{any}^5\cdot\mathrm{IS}\cdot\mathrm{DET}\cdot(\mathit{any}^2+(\mathrm{IS}\cdot\mathrm{DET}\cdot\mathit{any}^4))\cdot\mathrm{DET}?\cdot\mathit{any}\cdot$
  $\mathrm{IS}\cdot\mathrm{DET}\cdot\mathit{any}^2)?\cdot\mathit{any}^2)?\cdot\mathit{any}^4)+\mathrm{IS})\cdot\mathrm{IS}\cdot\mathrm{DET}\cdot\mathit{any}^5))\cdot\mathit{any}^3\cdot\mathrm{DET}\cdot\mathit{any}?\cdot$
  $\mathit{any}^2\cdot\mathrm{DET}\cdot\mathit{any}?\cdot\mathit{any}\cdot\mathrm{IS}^2\cdot\mathit{any}\cdot((\mathit{any}?\cdot\mathit{any}\cdot\mathrm{IS}\cdot(\mathrm{IS}\cdot\mathit{any}\cdot\mathrm{DET}\cdot(\mathrm{DET}+$
  $(\mathit{any}^2?\cdot\mathit{any}^2)?)\cdot\mathit{any})?\cdot\mathit{any}^3)+\mathrm{DET})\cdot\mathit{any}^3\cdot\mathit{any}?\cdot\mathit{any}^5\cdot\mathrm{IS}\cdot\mathit{any}\cdot\mathrm{DET}\cdot$
  $\mathrm{IS}\cdot\mathrm{DET}\cdot\mathit{any}\cdot\mathrm{IS}\cdot\mathit{any}^4)?\cdot\mathit{any}^2)?\cdot\mathit{any}\cdot(\mathrm{IS}\cdot\mathrm{DET}\cdot((\mathit{any}\cdot\mathrm{IS}\cdot\mathrm{DET}?\cdot\mathit{any})+$
  $\mathit{any}+(\mathit{any}^4?\cdot\mathit{any}^2))\cdot\mathit{any}^5\cdot\mathrm{IS}^2\cdot\mathit{any}\cdot((\mathit{any}^3\cdot\mathrm{DET}\cdot(\mathit{any}^3+\mathit{any})\cdot\mathit{any}^4\cdot\mathrm{IS}\cdot$
  $\mathit{any}\cdot((\mathrm{IS}\cdot\mathit{any}\cdot((\mathit{any}^4\cdot\mathrm{DET}\cdot\mathrm{IS}\cdot(\mathrm{IS}\cdot\mathrm{DET}\cdot\mathit{any}^4)?\cdot\mathit{any}^3\cdot\mathrm{DET}\cdot\mathit{any}?\cdot\mathrm{IS}^3\cdot$
  $\mathit{any}^6)?\cdot\mathrm{IS}\cdot\mathit{any}\cdot((\mathit{any}\cdot\mathrm{IS}\cdot\mathrm{DET}\cdot\mathit{any}^2\cdot\mathrm{IS}\cdot\mathit{any}\cdot(\mathrm{DET}\cdot\mathit{any}^2)?\cdot\mathit{any}^2\cdot\mathrm{DET}\cdot$
  $\mathit{any}?\cdot\mathit{any}\cdot\mathrm{DET})?\cdot\mathit{any}^2)?\cdot\mathit{any}^3)?\cdot(\mathrm{IS}\cdot\mathrm{DET}\cdot(\mathit{any}^3+\mathit{any}?)\cdot\mathit{any}^3\cdot\mathrm{IS}\cdot\mathit{any}\cdot$
  $\mathrm{IS}^2\cdot\mathrm{DET}?\cdot(\mathrm{IS}\cdot\mathrm{DET}\cdot\mathit{any}^4\cdot\mathrm{DET}\cdot\mathrm{IS}\cdot(((\mathit{any}^3\cdot\mathrm{IS}\cdot\mathit{any}\cdot\mathrm{DET}?\cdot\mathrm{DET}\cdot\mathit{any}^3\cdot$
  $\mathrm{IS}\cdot\mathit{any}^4)+(\mathrm{IS}\cdot\mathrm{DET}\cdot\mathit{any}\cdot\mathrm{DET}?\cdot\mathit{any}^2))\cdot\mathit{any}^2)?\cdot\mathrm{IS}^2\cdot\mathit{any}^4\cdot\mathrm{DET}?\cdot\mathit{any}^9)?\cdot$
  $\mathit{any}\cdot(\mathrm{IS}\cdot\mathrm{DET}\cdot(\mathit{any}^3+\mathit{any})\cdot\mathit{any}\cdot((\mathrm{IS}^2\cdot\mathit{any}^5\cdot\mathrm{DET}\cdot((\mathit{any}^2\cdot\mathrm{IS}\cdot\mathrm{DET}?\cdot$

$any^5) + any) \cdot any^3 \cdot \mathrm{IS} \cdot \mathrm{DET} \cdot any^5 \cdot (((any^2 \cdot \mathrm{DET}) + any) \cdot any^2)? \cdot any) + any))? \cdot any)? \cdot any^4)? \cdot any^2)? \cdot \mathrm{DET}? \cdot any? \cdot any^7) + any) \cdot any^3)? \cdot any$

4. Best individual for AExp4; Satisfy all the following:

   - $any^3 \cdot (\mathrm{DET} + any^2) \cdot any^2 \cdot (any + any^4)$
   - $any^3 \cdot (\mathrm{DET} + any^3) \cdot any^3$
   - $any^3 \cdot (\mathrm{DET} + any^5)$
   - $\mathrm{DET} \cdot any^2$
   - $any \cdot \mathrm{IS} \cdot any^5$
   - $\mathrm{IS} \cdot any^5$
   - $\mathrm{DET} \cdot any^4$
   - $any \cdot \mathrm{IS} \cdot (\mathrm{DET} + \mathrm{MEAN})$

# References

Baker, J.E. (1985). Adaptive selection methods for genetic algorithms. In *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*, Erlbaum. 51

Baker, J.E. (1987). Reducing bias and inefficiency in the selection algorithm. In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application*, Erlbaum. 49

Belz, A. & Eskikaya, B. (1998). A genetic algorithm for finite state automata induction with an application to phonotactics. In *Proceedings of the ESSLLI-98 Workshop on Automated Acquisition of Syntax and Parsing*. 37, 38

Blair-Goldensohn, S., McKeown, K.R. & Schlaikjer, A.H. (2004). *New Directions in Question Answering*, chap. Answering Definitional Questions – A Hybrid Approach. AAAI Press. 8, 27, 30, 32, 33, 135

De Jong, K.A. (1975). *An Analysis of the Behaviour of a Class of Genetic Adaptive Systems*. Ph.D. thesis, University of Michigan, ann Arbor. 50

de la Maza, M. & Tidor, B. (1991). Boltzmannn Weighted Selection Improves Performance of Genetic Algorithms. Tech. rep., Massachusetts Institute of Technology, Cambridge, MA, USA. 91

Degórski, Ł., Marcińczuk, M. & Przepiórkowski, A. (2008). Definition extraction using a sequential combination of baseline grammars and machine learning classifiers. In E.L.R.A. (ELRA), ed., *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*. 23, 31, 33, 67

Fahmi, I. & Bouma, G. (2006). Learning to Identify Definitions using Syntactic Features. In *Workshop of Learning Structured Information in Natural Language Applications, EACL, Italy*. 8, 22, 23, 30, 31, 32, 33, 67, 135

Finkel, J.R., Grenager, T. & Manning, C. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, 363–370. 62

Forrest, S. (1985). Scaling fitness in the genetic algorithm, In Documentation for PRISONERS DILEMMA and NORMS Programs That Use the Genetic Algorithm. 49

Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA. 34, 54

Gray, H. (1918). *Anatomy of the human body*. Philadelphia: Lea and Febiger. 135

Holland, J.H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor. 34, 49, 54, 59

Ide, N. & Suderman, K. (2002). XML Corpus Encoding Standard Document XCES 0.2. Tech. rep., Department of Computer Science, Vassar College, and Equipe Langue et Dialogue, Vandoeuvre-lés-Nancy. 62

Joho, H., Liu, Y.K. & Sanderson, M. (2001). Large scale testing of a descriptive phrase finder. In *HLT '01: Proceedings of the first international conference on Human language technology research*, 1–3. 26, 27

Joshi, A.K. & Schabes, Y. (1997). Tree-Adjoining Grammars. In *Handbook of Formal Languages*, vol. 3, 69–124, Springer. 41

Keller, B. & Lutz, R. (1997). Learning Stochastic Context-Free Grammars from Corpora Using a Genetic Algorithm. In *Workshop on Automata Induction Grammatical Inference and Language Acquisition (ICML-97)*, Nashville, Tennessee. 38, 39

KLAVANS, J.L., POPPER, S. & PASSONNEAU, R. (2003). Tackling the internet glossary glut: Automatic extraction and evaluation of genus phrases. In *SIGIR'03 Workshop on Semantic Web*. 23, 25, 33

KOBYLIŃSKI, Ł. & PRZEPIÓRKOWSKI, A. (2008). Definition Extraction with Balanced Random Forests. In *proceedings of GoTAL 2008*. 23, 30, 31, 33, 67

KOZA, J.R. (1992). *Genetic Programming: On the Programming of Computers by means of Natural Selection*. MIT Press, Cambridge, MA. 34, 40, 55, 80

KOZA, J.R., KEANE, M.A., STREETER, M.J., MYDLOWEC, W., YU, J. & LANZA, G. (2005). *Genetic Programming IV: Routine Human-Competitive Machine Intelligence*. Springer. 55, 58, 59

LANKHORST, M.M. (1994). Breeding Grammars: Grammatical Inference with a Genetic Algorithm. Tech. Rep. CS-R9401, Department of Computer Science, University of Gronigen, PO Box 800, 9700 AV Groningen, The Netherlands. 35, 41

LIU, B., CHIN, C.W. & NG., H.T. (2003). Mining Topic-Specific Concepts and Definitions on the Web. In *Proceedings of the Twelfth International World Wide Web Conference (WWW'03)*. 24, 25, 30, 33, 132

LOSEE, R.M. (1996). Learning Syntactic Rules and Tags with Genetic Algorithms for Information Retrieval and Filtering: An Empirical Basis for Grammatical Rules. In *Information Processing and Management*, vol. 32. 36, 41

MALAISÉ, V., ZWEIGENBAUM, P. & BACHIMONT, B. (2004). Detecting semantic relations between terms in definitions. In *COLING CompuTerm 2004: 3rd International Workshop on Computational Terminology*, 55–62. 19, 33, 111, 132

MANNING, C.D., RAGHAVAN, P. & SCHÜTZE, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press. 17

MARCUS, M.P., MARCINKIEWICZ, M.A. & SANTORINI, B. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, **19**, 313–330. 111, 119

McCarthy, J. (2007). What is artificial intelligence? Stanford University. 44

Miliaraki, S. & Androutsopoulos, I. (2004). Learning to Identify Single-Snippet Answers to Definition Questions. In *Proceedings of Coling 2004*, 1360–1366, COLING, Geneva, Switzerland. 26, 32, 33

Mitchell, M. (1998). *An Introduction to Genetic Algorithms*. MIT Press. 48, 50, 59, 91

Monachesi, P., Lemnitzer, L. & Simov, K. (2007). Language Technology for eLearning. In *First European Conference on Technology Enhanced Learning*. 6, 61

Muresan, S. & Klavans, J.L. (2002). A method for automatically building and evaluating dictionary resources. In *Proceedings of the Language Resources and Evaluation Conference*. 18, 30, 33, 111, 132

Park, Y., Byrd, R.J. & Boguraev, B.K. (2002). Automatic glossary extraction: beyond terminology identification. In *Proceedings of the 19th international conference on Computational linguistics*, 1–7, Association for Computational Linguistics, Morristown, NJ, USA. 18, 30, 33, 111

Prager, J., Radev, D. & Czuba, K. (2001). Answering what-is questions by Virtual Annotation. In *HLT '01: Proceedings of the first international conference on Human language technology research*, 1–5. 26, 27

Przepiórkowski, A., Degórski, Ł., Spousta, M., Simov, K., Osenova, P., Lemnitzer, L., Kubon, V. & Wójtowicz, B. (2007). Towards the automatic extraction of definitions in Slavic. In *Proceedings of the BSNLP workshop at ACL 2007*. 21, 33, 132

Sang, E.T.K., Bouma, G. & de Rijke, M. (2005). Developing Offline Strategies for Answering Medical Questions. In *Proceedings of the AAAI-05 Workshop on Question Answering in Re stricted Domains*, 41–45. 29, 31, 33

Shaw, W.C. (1922). *The Art of Debate*. Allyn and Bacon. 3

Smith, T.C. & Witten, I.H. (1995). A Genetic Algorithm for the Induction of Natural Language Grammars. In *Proceedings IJCAI-95 Workshop on New Approaches to Learning for Natural Language Processing, Canada*, 17–24. 40

Spasić, I., Nenadić, G. & Ananiadou, S. (2004). Learning to Classify Biomedical Terms through Literature Mining and Genetic Algorithms. In *Intelligent Data Engineering and Automated Learning (IDEAL 2004)*, vol. 3177 of *LNCS*, 345–351, Springer-Verlag. 39

Storrer, A. & Wellinghoff, S. (2006). Automated detection and annotation of term definitions in german text corpora. In *Language Resources and Evaluation Conference*. 19, 20, 30, 33, 132

Tobin, R. (2005). Lxtransduce A replacement for fsgmatch. Tech. rep., University of Edinburgh. 62, 80

Toutanova, K. & Manning, C.D. (2000). Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger. In *Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000), Hong Kong*. 62

Walter, S. & Pinkal, M. (2006). Automatic Extraction of Definitions from German Court Decisions. In *Workshop on Information Extraction Beyond The Document*, 20–28. 20, 21, 30, 33

Westerhout, E. & Monachesi, P. (2007a). Combining pattern-based and machine learning methods to detect definitions for elearning purposes. In *Natural Language Processing and Knowledge Representation for eLearning Environments Workshop*. 22, 30, 31, 67

Westerhout, E. & Monachesi, P. (2007b). Extracting of Dutch Definitory Contexts for elearning purposes. In *CLIN 2007*. 33

Westerhout, E. & Monachesi, P. (2008). Creating glossaries using pattern-based and machine learning techniques. In *LREC2008*. 23, 30, 111

Witten, I.H. & Frank, E. (2005). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann. 23, 135

Wyard, P.J. (1991). Context Free Grammar Induction Using Genetic Algorithms. In *ICGA*, 514–519. 40