Tuning Context Features with Genetic Algorithms[*]

Irena Spasić, Goran Nenadić, Sophia Ananiadou

Computer Science, University of Salford
Newton Building, Manchester, M5 4WT, UK
{I.Spasic, G.Nenadic, S.Ananiadou}@salford.ac.uk

Abstract

In this paper we present an approach to tuning of context features acquired from corpora. The approach is based on the idea of a genetic algorithm (GA). We analyse a whole population of contexts surrounding related linguistic entities in order to find a generic property characteristic of such contexts. Our goal is to tune the context properties so as not to lose any correct feature values, but also to minimise the presence of ambiguous values. The GA implements a crossover operator based on dominant and recessive genes, where a gene corresponds to a context feature. A dominant gene is the one that, when combined with another gene of the same type, is inevitably reflected in the offspring. Dominant genes denote the more suitable context features. In each iteration of the GA, the number of individuals in the population is halved, finally resulting in a single individual that contains context features tuned with respect to the information contained in the training corpus. We illustrate the general method by using a case study concerned with the identification of relationships between verbs and terms complementing them. More precisely, we tune the classes of terms that are typically selected as arguments for the considered verbs in order to acquire their semantic features.

1. Introduction

The automatic discovery of new knowledge encoded in text documents relies heavily on the identification of concepts, linguistically represented by domain specific terms (Maynard et al., 2000). New terms representing newly identified or created concepts appear rapidly due to the rapid growth of new knowledge and textual data describing it. This makes the automatic term extraction tools essential assets for efficient knowledge discovery. However, automatic term extraction itself is not the ultimate goal since the large and ever growing number of terms calls for a systematic way to access and retrieve the information about the terms. Therefore, the extracted terms need to be placed in an appropriate framework by establishing relations to other content words, primarily to other terms and domain specific verbs. These relations, like terms themselves, need to be extracted from text.

In corpus-based knowledge acquisition methods, the hypothesis about terms and relationships between them are formed by analysing the contexts of the terms. Not all word types found in the context are of equal importance in the process of reasoning about the terms: the most informative are verbs, noun phrases (especially terms) and adjectives. (Maynard et al., 1998) used a fixed-size context window containing the three word types in order to incorporate the context factor into an automatic term extraction procedure. The method is further improved by using the semantic knowledge about context terms (Maynard et al., 2000). (Hatzivassiloglou et al., 2002) used gene and protein names to classify domain specific verbs by counting the frequencies with which they co-occur as well as the total frequency of the verbs. A number of authors have been using pattern-based methods to identify pre-defined types of the relationships between terms. (Hearst, 1992) focused on the extraction of hyponym relation existing between terms based on manually defined patterns describing the syntactic structure of the word sequence encoding the relation. Similarly, (Thomas et al., 2000) used manually identified lexico-syntactic patterns as filters for extraction of protein interactions. (Agichtein et al., 2000), on the other side, extracted the patterns automatically from a corpus by providing the system with pairs of terms known to be in a considered relation, locating the word sequences in which the two terms appear in close proximity, and analysing the context connecting them. (Pustejovsky et al., 2002) differentiate between entity extraction and relation extraction. They use predicates (expressed by verbs and their nominalisations) as anchors in identifying relations. Complementation patterns are extracted from clustered predicate contexts based on their regularity.

In our previous work (Nenadic et al., 2001) we used a genetic algorithm (GA) based on a novel crossover operator to explore the context features of prepositions in Serbian by concentrating on case properties. The GA was able to automatically learn case constraints of noun phrases within specific preposition phrases by consulting non-disambiguated initially tagged corpus. In this paper we present a similar approach to identifying relationships between verbs and terms complementing them. We use a GA to perform reasoning about term classes allowed to be combined with specific verbs. The approach has been tested in the field of molecular biology. We aim at automatic tuning of the features of elements found in the context of such verbs (e.g. whether these elements are proteins or genes), by using an existing ontology as seed for learning. The results of the proposed methodology provide a platform for term clustering/classification, term sense disambiguation, verb subcategorisation, and verb class disambiguation.

The paper is organised as follows. In Section 2 we provide a brief overview of genetic algorithms accompanied with the specificities of the method that we have introduced. Section 3 illustrates the method by means of a case study related to the problem of complementation patterns for specific verbs in the domain

[*] This research is a part of the BioPATH research project coordinated by LION BioScience (http://www.lionbioscience.com) and funded by German Ministry of Research

of molecular biology. Further, in Section 4 we explain how the method may be used as a platform for solving a variety of problems in terminology management. Finally, we conclude the paper in Section 5.

2. General method

Genetic algorithms are meta-heuristics incorporating the principles of natural evolution and the idea of "survival of the fittest" (Reeves, 1996). A solution is encoded as a sequence of genes, referred to as an *individual*. In the initial phase of the GA a number of individuals is generated typically in a random manner. We, however, collect the individuals from a corpus in order to form the initial population. The following section describes the way in which this is done in more detail.

Operators typical of GAs, namely selection, crossover, mutation, and replacement, are applied, in that order, in each iteration of the GA. *Selection* is usually defined probabilistically: the better the solution, the higher the probability for that solution to be selected as a parent. We depart from this approach as all the individuals from the current population are selected, thus giving each of them a possibility to pass their genetic material onto their offspring.

Crossover is applied to a pair of parents resulting in their recombination, called children. We used a novel crossover operator based on the notion of dominant/recessive genes (Nenadic et al., 2001) in which recessive genes are passed to the offspring only in the absence of a dominant gene of the same type. Unlike in traditional GA algorithms, where two children are produced per one application of the crossover operator, our crossover operator results in one child, which necessarily inherits the dominant characteristics from its parents and inherits the recessive characteristics if they are not blocked by the dominant genes of the same type. Logically, we chose more desirable characteristics (depending on a specific application) to be denoted by dominant genes, as we wish them to "override" the less desirable ones when performing crossover between two individuals. The crossover operator based on dominant/recessive genes has to be defined individually for each specific problem and its representation.

The *mutation* operator introduces diversity into a population by modifying a small portion of newly formed solutions in a random manner. We do not use this operator since it would affect the results in an unwanted manner. Namely, we want to extract useful information contained in the corpus and mutation would distort this information by randomly changing it.

Generally in GAs, once all the new individuals have been evaluated, the fittest ones *replace* the appropriate number of the less fit old solutions, thus forming a new population. In our approach the evaluation of the fitness is redundant because the crossover based on dominant/recessive genes guarantees for the offspring to be fitter than both of its parents. It, therefore, replaces both of its parents in the next population. Each population formed in this way is referred to as a *generation*. This process is repeated from generation to generation. In each iteration, the number of individuals in the population is reduced by replacing a pair of parents with their child. By successively substituting recessive genes by dominant ones, we progressively refine a set of context features. Eventually, this results in a single individual that comprises all features determined by dominant genes present in the initial population. In other words, there is only one individual remaining, which is fitter than all of its antecessors, and which describes the context information that is tuned with respect to the information contained in the training corpus.

3. Case study: tuning verb complements

We will illustrate the proposed method by using a case study, which relates to the problem of complementation patterns for domain specific verbs. More precisely, we are interested in classes of terms that are typically selected as arguments for the considered verbs. We restricted the tests to the field of molecular biology.

3.1. Problem description

By looking at the context of an isolated verb occurrence it is difficult to predict all term classes that can be combined with the given verb. On the other hand, the whole "population" of terms complementing a specific verb is likely to provide a certain conclusion about that verb with respect to its complementation patterns. This was a primary motivation for using GA as it operates on a *population* of individuals (in our case, represented as sequences of terms) as opposed to a *single* individual. This fact also makes the approach robust since it does not rely solely on every specific instance of verb-term combination to be correctly recognised. A whole population of terms complementing a specific verb captures some properties of allowed term-verb combinations. Our goal was to tune these properties so as to preserve all the term classes that can complement a given verb, but to minimise overgeneralisation of the results at the same time.

3.2. Initial population

The initial population consists of terms collected from a corpus by using domain specific verbs as anchors. A number of most frequently used verbs are extracted from a corpus of 2008 abstracts retrieved from the MEDLINE database (MEDLINE, 2002). The number of verbs is than reduced by eliminating general verbs that are frequently used in scientific papers, but which, on the other hand, are not domain specific (e.g. `observe`, `demonstrate`, `explain`, etc.). We kept top 21 most frequent verbs, each of which is than analysed individually in order to deduce its domain specific frame (i.e. the type of terms that can be used as arguments of the verb in question).

We adopted a heuristic approach to extracting verb arguments, i.e. the terms that make up the initial population. First, we noted that transitive verbs dominated the list of the most frequent domain specific verbs (see Table 1). Only 2 out of 21 verbs were intransitive, though largely linking two entities based on the following pattern:

`<term> <verb> with <term>`
(e.g. `aryl_hydrocarbon_receptor interacts with estrogen_receptor_alpha`)

denoting the existence of a certain relationship between two terms. Further, purely transitive verbs mainly follow two patterns depending on whether they are used in an active or passive form. The pattern for the verb used in active is the following:

`<term> <verb> <term>`
(e.g. `C_terminal_tail inhibits the Pitx2_protein`)

The following pattern describes the passive usage of a verb:

`<term> <verb> by <term>`
(e.g. `chain_promoters were inhibited by C/EBPbeta_isoforms`)

Finally, the verbs that have both transitive and intransitive sense are predominantly used in the former sense in the corpus, in which case the same patterns mentioned for transitive verbs are applicable.

# of verbs	VI or VT	example
2	VI	compete, interact
8	VT	activate, induce, repress
11	both	bind, mediate, stimulate

Table 1: The distribution of domain specific verbs

We expected for each of the analysed verbs to be complemented by a term in both left and right context. However, it is not obligatory for a term to be a direct neighbour of a given verb, e.g.:

a **protein** that **activates** transcription of a large number of **viral_genes**

We, therefore, did not limit ourselves only to extraction of terms immediately preceding/following a verb or identifying its subject/object, but have instead used a knowledge-poor approach in which we extracted the terms closest to the verb in its left and right contexts respectively without crossing the sentence boundaries. This is done by defining the appropriate local grammars in a text mining software developed within the BioPATH project. Here is an example of a local grammar according to which the terms are to be extracted, where we covered all morphological forms of the verb `inhibit` including its nominalisation:

```
<right-context> → <verb> <non-term>* <term>
<verb> → inhibit( ε | s | ed | ing | ion )
```

Once the word sequences that match the defined pattern are identified in the corpus, the terms used in conjunction with the given verb are extracted and the duplicates are eliminated. These terms constitute the initial population of the GA.

3.3. Dominance relation

The output of the GA should be a sequence of terms corresponding to the concepts that include all other concepts denoted by terms in the initial population either as its (in)direct subclasses or instances. This output is obtained by iteratively applying a crossover operator based on a partial order relation induced by a domain specific ontology (Figure 1) developed within the BioPATH project.

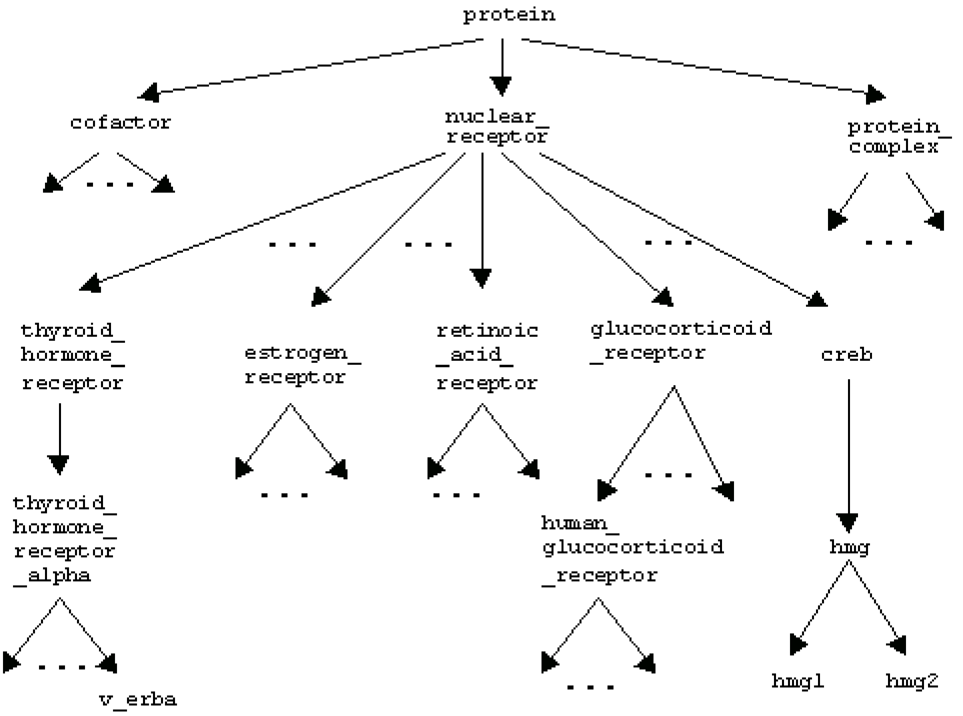

Figure 1: An excerption from a domain specific ontology

A few definitions are needed to describe the way in which the crossover is applied. Let us first define precisely the genotype representation that will be used henceforth. If $t_1,..., t_n$ are terms, then the disjunction $t_1 + ... + t_n$ is a *genotype*, where the disjuncts $t_1,..., t_n$ are *genes*. We use a GA in order to generate a hypothesis about the term classes used in conjunction with a specific verb, based on the information contained in the corpus and the ontology. More precisely, we are interested in a *minimal* genotype representation, where this optimality condition is two-dimensional. Namely, we want to minimise both the length of a genotype (i.e. to minimise n in $t_1 + ... + t_n$) and the depth of each individual gene in the ontology. In order to achieve this goal we use a selective breeding strategy based on the crossover operator that we now introduce.

In general, let terms t_1 and t_2 be the genes belonging to two different parents. We say that t_1 *dominates* t_2 (i.e. t_2 is *recessive* with respect to t_1) iff t_1 is an antecessor of t_2 in the ontology. A term in one parent is also *recessive* when it has no ancestor among terms in the other parent.

3.4. Crossover operator

Let us first intorduce the crossover operator through examples. If we apply the crossover operator to two parents:

```
glucocorticoid_receptor
human_glucocorticoid_receptor
```

we would like their child to have a genotype glucocorticoid_receptor since it is more general, i.e. placed higher in the ontology in the path connecting the two terms (see Figure 1). Therefore, we will say that the first gene dominates the second. Further, consider what we would like to be a child's genotype if the parents have the following genotypes:

```
glucocorticoid_receptor +
thyroid_hormone_receptor_alpha
human_glucocorticoid_receptor +
thyroid_hormone_receptor
```

The preferred result should be:

```
glucocorticoid_receptor +
thyroid_hormone_receptor
```

as glucocorticoid_receptor dominates human_glucocorticoid_receptor and thyroid_hormone_receptor dominates thyroid_hormone_receptor_alpha (see Figure 1). Finally, if the crossover operator should be applied to the following pair:

```
glucocorticoid_receptor + CREB
human_glucocorticoid_receptor +
thyroid_hormone_receptor
```

the result should be glucocorticoid_receptor + CREB + thyroid_hormone_receptor, as human_glucocorticoid_receptor is the only recessive gene blocked by a compatible dominant gene in this case. Generally, a child is determined by replacing all recessive genes by their dominant counterparts (i.e. their antecessors in the ontology), and then forming the union of all genes found in its parents. Note that it is not necessary for every recessive gene to have a dominant counterpart when to parents are combined. If that is the case, then such a recessive gene will be inherited by the child.[1]

The crossover operator can be formally introduced as follows. Let p_1 and p_2 denote two individuals $t_1^1 + ... + t_n^1$ and $t_1^2 + ... + t_m^2$ respectively. A child resulting from a crossover operator applied on the two individuals is obtained as a result of the following pseudo-code:

```
for all genes $t_i^1$ in $t_1$
    for all genes $t_j^2$ in $p_2$
        if $t_i^1$ is an antecessor of $g_j^2$
        then flag $g_j^2$;
        else if $t_j^2$ is an antecessor of $t_i^1$
            then flag $t_i^1$;
collect all non-flagged genes;
the child of $p_1$ and $p_2$ is a disjunction
of the collected genes;
```

Figure 2: The crossover operator

Since the dominance between genes is defined through the partial order relation induced by the ontology, which is transitive, the order in which individuals are bred is of no importance, i.e. the crossover operator has the associative property. This fact gives rise to parallel processing of corpora: it can be used to accelerate the processing of an individual corpus, but it also gives way to the possibility of merging the results obtained from several corpora.

Verb	Initial population sample	Result
bind	creb	protein
	estrogen_receptor	
	glucocorticoid_receptor	
	hmg	
	human_glucocorticoid_receptor	
	protein	
	retinoic_acid_receptor	
	thyroid_hormone_receptor	
	v_erba	
inhibit	glucocorticoid_receptor	protein
	mineralocorticoid_receptor	
	pml	
	protein	
	rar_rxr_heterodimer	
	thyroid_hormone_receptor_alpha	
mediate	androgen_receptor	protein
	creb	
	estrogen_receptor	
	glucocorticoid_receptor	
	human_glucocorticoid_receptor	
	mineralocorticoid_receptor	
	protein	
	rar_rxr_heterodimer	
	retinoic_acid_receptor_alpha	

Table 2: The results for verbs bind, inhibit, and mediate

[1] In future work we plan to experiment with a controled mutation operator that would replace the remaining recessive terms with their nearest common antecessor in case the distance between the antecessor and the given terms is less than a given threshold.

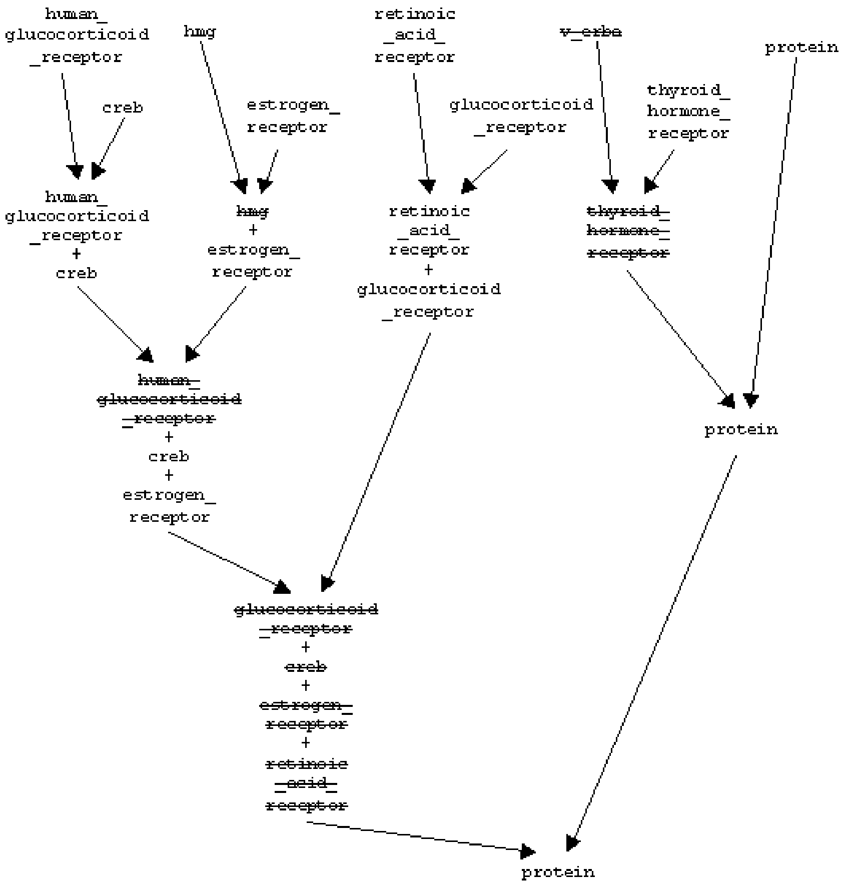
human_ glucocorticoid _receptor
hmg
retinoic _acid_ receptor
~~v_erba~~
protein
creb
estrogen_ receptor
glucocorticoid _receptor
thyroid_ hormone_ receptor
human_ glucocorticoid _receptor + creb
~~hmg~~ + estrogen_ receptor
retinoic _acid_ receptor + glucocorticoid _receptor
~~thyroid_ hormone_ receptor~~
~~human_ glucocorticoid _receptor~~ + creb + estrogen_ receptor
protein
~~glucocorticoid _receptor~~ + ~~creb~~ + ~~estrogen_ receptor~~ + ~~retinoic _acid_ receptor~~
protein

Figure 3: An example of crossover between the context terms for the verb `bind`

As an example, Figure 3 illustrates a sequence of generations of the GA, whose individuals represent terms that can complement the verb `bind`. The initial population is extracted from the corpus in the way described earlier in this section. The crossover operator is applied to the pairs of individuals,[2] as illustrated in Figure 3. Since the crossover operator is associative, the order in which we apply the crossover operator will not affect the result. We have run the proposed GA for the selected verbs (21 in total). In Table 2 we show the results for three of the verbs.

4. Applications

We have foreseen at least four potential applications of the described GA: term clustering and classification, term sense disambiguation, verb subcategorisation, and verb class disambiguation. Let us now discuss these applications in more detail.

4.1. Term clustering/classification

Automatically recognised terms should be related to existing knowledge and/or to each other. This entails the fact that terms should be classified or clustered so that semantically similar terms are grouped together. Classification and/or clustering of terms are indispensable for improving information extraction, knowledge acquisition, and document categorisation. Classification can also be used for efficient term management and populating and updating existing ontologies in a consistent manner.

All the results presented in the previous section have been obtained by running the GA on a set of terms complementing a specific verb, but which are also already classified within the ontology used to establish a dominance relation between terms. However, it is also possible to run the GA on a wider set of terms, in which case the newly recognised terms (i.e. the ones not found in the ontology) will be always passed to the offspring as non-classified terms cannot be dominated by other terms. This finally results in such terms being present as genes in the final population. All the genes defining an individual in the final population can be than divided into two groups based on the criterion of their (non)existence in the ontology. The terms that are present in the ontology are candidate classes for the newly recognised terms. For more fine-grained results the terms placed close to the root of the ontology should be removed from the initial population, thus being unable to override the terms found lower in the hierarchy. The depth up to which the terms are to be removed from the initial population may be user-specified, and it indirectly corresponds to the notion of distance in the classical clustering algorithms (Fasulo, 1999). In order to link the newly recognised terms to specific candidate classes, we can use a hybrid similarity measure (Nenadic et al., 2002). The candidate classes for each new term can be ranked based on the similarity value

[2] When there is an odd number of individuals in the population, one of them is simply passed to the next generation without being combined with another individual.

calculated for the new term and a term describing a candidate class (and/or its subordinate terms). This way the automatic ontology update can be supported straightforwardly (Bisson et al., 2000).

4.2. Term sense disambiguation

Dealing with term sense disambiguation is crucial for classifying terms and ontology populating. The appropriate term sense is usually discovered by examining the similarity between the given term and its context. The described GA can be used for term sense disambiguation as well, which is essential for resolving terminological confusion occurring in the field of molecular biology.

Term sense ambiguities occur due to one-to-many correspondence existing between terms and the concepts they describe (e.g. `GR` is an acronym for both `glucocorticoid_receptor` and `glutathione_reductase`). Verbs are useful in resolving this type of ambiguity once it is determined which classes of terms can be used as its arguments, which is exactly what is determined by our GA algorithm. For example, the two senses of the term mentioned above may be distinguished by analysing its context. Once the verb with which the term is used as an argument is identified, we can use the information attached to this verb to adopt the appropriate meaning of the term:

```
... GR catalyses electron transfer ...
```

The knowledge of `glucocorticoid_receptor` being a `nuclear_receptor` and `glutathione_ reductase` being an `enzyme` is represented in the ontology. In combination with the knowledge about complementation patterns for the verb `catalyse` (it is combined with the terms belonging to the class of enzymes) acquired from the corpus, we can eliminate `glucocorticoid_receptor` as an interpretation of `GR` in the given example.

4.3. Verb subcategorisation

Verb subcategorisation frames tend to vary in different sublanguages. The results of the GA can be used for domain-specific verb subcategorisation based on the similarity between the term classes complementing the verbs. More precisely, the disjunctions of term classes attached to the verbs by the GA can be compared to estimate the degree of their overlapping (common disjuncts and the existence of IsA relationship between the disjuncts in two respective terms). For instance, the three verbs shown in Table 2 can belong to the same class since all of them describe relations between the entities of the same type, `proteins` in this case.

4.4. Verb class disambiguation

Finally, verb class disambiguation can be performed in a way similar to term sense disambiguation described earlier. A term used in conjunction with an ambiguous verb is matched against the disjunctions of term classes attached to the verbs by the GA in order to point to the correct interpretation of the verb. In the following example:

```
the doctor interacts with a patient
```

the knowledge on the terms `doctor` and `patient` can be used to eliminate a potential interpretation of the verb `interact` referring to a biochemical reaction between proteins.

5. Conclusions

We described a general GA that can be easily adapted to operate on contexts of specific linguistic entities at various levels. At lexical level, specific words are in the centre of the context (Nenadic et al., 2001). At syntactic level, context of specific syntactic classes (e.g. nouns, transitive verbs, etc.) or structures (e.g. noun phrases, prepositional phrases, etc.) are considered. At semantic level, we deal with concepts denoted by semantic classes in domain specific corpora.

In this paper, we employed the algorithm at semantic level in order to learn complementation patterns for specific verbs in the domain of molecular biology. We explained how these patterns could be subsequently used as a platform for solving a variety of other problems in terminology management, e.g. term clustering/classification, term sense disambiguation, verb subcategorisation, and verb class disambiguation.

This case study together with the case study from previous work (Nenadic et al., 2001) shows that the concept of using dominant and recessive genes as a basis for a crossover operator in GAs is a simple idea that can be easily adapted for a variety of problems in NLP. This is done by choosing an appropriate problem representation and then defining dominant and recessive genes in terms of the preferred solution to the problem.

Our further work will be concerned with building applications, described in Section 4, on top of the described GA.

6. References

Agichtein, E., Gravano, L., 2000. Snowball: Extracting Relations from Large Plain-Text Collections. In *Proceedings of the 5th ACM International Conference on Digital Libraries* (DL'00).

Bisson, G., Nedellec, C., Canamero, D., 2000. Designing Clustering Methods for Ontology Building - The Mo'K Workbench. In S. Staab, A. Maedche, C. Nedellec, and P. Wiemer Hastings (Eds.): *Proceedings of the Workshop on Ontology Learning, 14th European Conference on Artificial Intelligence ECAI'00*, Berlin, Germany.

Fasulo, D., 1999. Analysis on Recent Work on Clustering Algorithms. *Technical Report #01-03-02*, Department of Computer Science and Engineering, University of Washington, Seattle.

Hatzivassiloglou, V., Weng, W., 2002. Learning Anchor Verbs for Biological Interaction Patterns from Published Text Articles. In R. Baud and P. Ruch (Eds.): *Proceedings of Workshop on Natural Language Processing in Biomedical Applications*, Nicosia, Cyprus.

Hearst, M.A., 1992. Automatic Acquisition of Hyponyms from Large Text Corpora. In *Proceedings of the 14th International Conference on Computational Linguistics*, Nantes, France.

Maynard, D., Ananiadou, S., 1998. Acquiring Contextual Information for Term Disambiguation. In *Proceedings of Computerm '98 Workshop on Computational Terminology* (COLING/ACL '98), Montreal, Canada, 86-91.

Maynard, D., Ananiadou, S., 2000. Identifying Terms by their Family and Friends. In *Proceedings of COLING 2000*, Saarbrucken, Germany, 530-536.

MEDLINE, 2002. National Library of Medicine. Available at: *http://www.ncbi.nlm.nih.gov/PubMed/*

Nenadic, G., Spasic, I., Ananiadou, S., 2001. Reducing Lexical Ambiguity in Serbo-Croatian by Using Genetic Algorithms. *Fourth European Conference on Formal Description of Slavic Languages FDSL-4*, Potsdam, Germany.

Nenadic, G., Spasic, I., Ananiadou, S., 2002. Automatic Discovery of Term Similarities Using Pattern Mining. Submitted.

Pustejovsky, J., Castaño, J., Zhang, J., Kotecki, M., Cochran, B., 2002. Robust Relational Parsing Over Biomedical Literature: Extracting Inhibit Relations. In *Proceedings of The Pacific Symposium on Biocomputing* (PSB 2002), Hawaii.

Reeves, C., 1996. Modern Heuristic Techniques. In: Rayward-Smith, V. et. al (Eds.) *Modern Heuristic Search Methods*. John Wiley & Sons Ltd.

Thomas, J., Milward, D., Ouzounis, C., Pulman, S., Carroll, M., 2000. Automatic Extraction of Protein Interactions from Scientific Abstracts. In *Proceedings of Pacific Symposium on Biocomputing 5*, 538-549.