Evolutionary Grammar Induction for Protein Relation Extraction

Dimitris Gavrilis[1], Ioannis Tsoulos[2] and Evangelos Dermatas[1]

[1]Department of Electrical & Computer Engineering, University of Patras
[2]Department of Computer Science, University of Ioannina
gavrilis@upatras.gr , sheridan@cs.uoi.gr , dermatas@george.wcl2.ee.upatras.gr

Abstract

A novel method is presented for protein relation extraction from scientific abstracts. The proposed method is based on Meta-Grammars, a novel method for grammar inference that uses genetic programming and a BNF description to discover a tree representation of sentence structure that can be used for information extraction. A series if transformations are applied to the original corpus before the Meta-Grammars genetic algorithm is applied. The proposed method is evaluated against extracting protein relations from scientific abstracts and it is shown that it requires a train corpus which has minimum requirements from field experts and giving precision of 79.165%.

Keywords: *Genetic Programming, Information Extraction, Grammatical Evolution, Protein Relations*

1. Introduction

The problem of information extraction from text and more specifically of protein relation extraction from scientific abstracts in biology still remains a crucial problem. Applications of information extraction from corpus will increase in the near feature since the rapid growth of the Internet and the number of natural language databases that are available mainly through the Internet.

In this direction, automatic generation of grammars accelerate the development time of natural language processing (NLP) tools and reduces significantly the production cost [1-4]. Among the initial publications for automatic grammar definition, the work of Horning [13] considered stochastic grammar inference using the Bayesian likelihood. The prior probability of a grammar is derived as the probability of obtaining it from an assumed grammar-generating MetaGrammar that favored the production of simple grammars over more complex ones. The notion of MetaGrammar (MG) was originally presented in [11] to automatically generate a Tree Adjoining Grammar (TAG) for French and Italian. A detail description of MetaGrammar methodology which has successfully been used for generating wide-coverage TAG for French is given in [12]: the grammar writer specifies, in a compact manner, syntactic properties that are usually encoded separately in the LFG machinery in the lexicon, in lexical rules and in rewriting rules. From this hierarchy an LFG is automatically generated. A number of desirable requirements for a MetaGrammar language are discussed in [14]: it should support disjunctions, to make it easy to express diathesis (such as active, passive), it should support conjunction so that complex descriptions can be assembled by combining several simpler ones, and it should support abstraction so that expressions can be named to facilitate reuse and avoid redundancy. A number of MetaGrammar tools are already available, such as:

1. The XDG development kit [15], offers a fexible method to define types of lexical entries, to build lexical abstractions, and to describe sets of lexical entries compactly. The MetaGrammar is processed to automatically generate all the entries of an XDG lexicon.
2. 2. The LORIA MG compiler is presented in [16] encoding Super-classes: A HyperTag captures the salient linguistic characteristics providing a set of quasi-nodes (i.e., variables), topological relations between these nodes (parent, dominates, precedes, equals), and a function for each quasi-nodes to decorate the tree (e.g., traditional agreement equations and/or LFG functional equations).
3. MetaGrammar tools describing in [17-20].

In fields such as biology, chemistry and biochemistry, new advances in technology (e.g. DNA microarrays) have allowed scientists to get quick results from experiments and report them on databases or in the form of scientific papers [7-10]. As a consequence, the number of papers and more generally the information that is made public on the WEB everyday increases with an enormous rate (hundreds of scientific papers are made available on PUBMED database every day). This phenomenon, forces scientists to spend enormous time in order to read and process this information while in some areas of research only specific information needs to be extracted.

Many attempts have been made in order to create automatic information extraction systems. Most of these systems have difficulties with scientific abstracts, especially in molecular biology, due to a huge number of technical and scientific terms and the complex sentence structure which used to describe proteins interaction in specific conditions. Another crucial problem in information extraction problems is the lack of tagged corpuses that can be used in order to train an

information extraction system for the new field of molecular biology. It should be noted that the construction of such corpuses is not only difficult but requires a lot of time and resources from experts. Furthermore, if such a corpus is created, it could target a limited number of domains.

Recently, a novel MetaGrammar Genetic Algorithm (mGGA) is presented in [5, 21] based on an evolvable grammar representation. In a number of benchmark problems the proposed evolutionary method the authors report significant performance gains when compared to static grammars.

In the present paper, a new approach for an information extraction system is presented which requires minimum information to achieve satisfactory results. This novel method, is based on grammatical evolution [4-5,20-21] and combines both syntactic and semantic information in order to extract protein relations from PUBMED abstracts. Among the main advantages of the proposed method is that the train corpus used requires only the sentences that contain the information to be tagged.

The structure of this paper is as follows. Section 2 provides a detailed description of the proposed MetaGrammar method, while in section 3 the Grammar induction algorithm is given. The evaluation process and a short discussion of the experimental results are presented in section 4. Finally some conclusions and additional details, concerning the directions of our future work, are given in the last section.

2. Method Description

The proposed method uses a series of transformations on the original corpus and an evolutionary approach to discover a tree representation of the sentence structure. The evolutionary approach uses genetic programming and a Backus Naur Form (BNF) grammar to induce the sentence tree. The use of genetic programming and BNF grammars is known as grammatical evolution [4]. In the present approach a BNF grammar is not directly used, but instead a meta-grammar is used to train the BNF grammar. The BNF grammar fits a context-free language model of the sentences that contain protein relations.

The train examples, that the meta-grammar uses, are not taken directly from the corpus but instead a series of transformations take place in order to achieve better generalization and classification performance. An overview of the algorithm and the corpus transformations follows:

1. *Tokenization:* The original corpus is split into tokens. The tokenization process involves splitting the words and the sentences using a standard tokenizer.
2. *Entity Tagging:* The biological entities are recognized by experts.
3. *Semantic Tagging:* Other entities that can improve the generalization capabilities and the performance of the proposed method are recognized and marked.
4. *Clustering:* The sentences are clustered based on their length and the entities they contain. For sentence clustering, the MetaGrammar algorithm is used. It is run iteratively for a small number of generations until some sentences are assigned to a template. The assigned sentences are removed and the algorithm is repeated on the remaining sentences. This process is repeated until all sentences are assigned to a group. A sentence is considered assigned to a group when the template can generate 70% of its length. This threshold is experimentally set.
5. *Grammar Induction:* For each cluster, the grammar is extracted.

The sentence clustering is applied in order to improve the system efficiency and generalization. Sentences with similar structure are grouped together and one grammar is extracted per group. The grammars are treated as templates in order to recognize sentences that contain protein relations and also to extract the actual relations from the sentences. The different tags improve greatly the algorithm's performance because it incorporates syntactic as well as semantic information in the extracted templates.

3. Grammar Induction

3.1. MetaGrammar description

The proposed method uses the Grammatical Evolution procedure to solve the grammar inference problem. In order to achieve this, a MetaGrammar must be used, which can represent any context–free language. Each production rule in a context–free language can be written in the form: $NT_K \rightarrow S_{K-0}S_{K-1}\ldots S_{K-N}$, where the symbol NT_K is a non–terminal symbol of the language and the symbols $S_{K-0}S_{K-1}\ldots S_{K-N}$ are terminal and non–terminal symbols. The symbol $\rightarrow$ in the above rule can be replaced with the function `Rule()` and the rule can be written using the following notation: $\texttt{Rule}(NT_K, S_{K-0}S_{K-1}\ldots S_{K-N})$.

Furthermore, each symbol in the rule can be replaced with a unique integer number, which yields the following form for the production rule: Rule(K,K-0,K-1,…,K-N). Therefore each context–free grammar can be written as a series of applications of the function Rule().

```
S::=<Rlist>
<Rlist>::=<R> | <R> <Rlist>
<R>::= <NT> <Right>
<Right>::=<T> | <NT> | <Right><Right>
<T>::=1 | 2 | 3 | … | TCOUNT
<NT>::=TCOUNT+1 | TCOUNT+2| … | TCOUNT+NTCOUNT
```

Fig. 1. Meta-Grammar BNF Description

The amount of terminal symbols in the language to be discovered are denoted by the symbol TCOUNT and the desired number of non–terminal symbols in the induced grammar is denoted by the symbol NTCOUNT. The description of the MetaGrammar used in this paper is shown in Fig. 1.

3.2. Fitness evaluation

In the induced grammar, the symbol P denotes the amount of positive examples and N denotes the amount of negative examples. The steps, for the estimation of the i chromosome fitness, are the following:

1. **Set** $v \leftarrow 0$.
2. **For** j=1,..,**P Do**
   If the positive sentence-j is not recognized by the grammar G_i, set $v \leftarrow v+1$.
   End For
3. **For** j=1,..,**N Do**
   If the negative sentence-j is recognized by the grammar G_i, set $v \leftarrow v+100$.
   End For
4. **Return** the value v as the fitness of the chromosome **i**.

A penalty value of 100 is added when a negative example is recognized. The process of sentence recognition was realized by the UNGER method [6], because the traditional approaches such as LL(1) and LR(1) parsers can not be used if the underlying grammar is not expressed in a specific form. For the estimation of the chromosome fitness, two alternative methods were implemented: Partial parsing and Semi-partial parsing. The first tries to match the sentence starting from the beginning and it is found to be generally superior in contrast to the latter method which tries to match the lengthier sentence starting from any point. The two methods are described below:

3.3. Partial parsing

The steps for the estimation of the chromosome i fitness, if the partial parsing method is used, is as follows:

1. **Set** $v \leftarrow 0$.
2. **For** j=1,..,**P Do**
   - Denote with a_j the percentage of the recognition of the positive senetnce-j, if the parser starts to match symbols from the most left element of the program.
   - $v \leftarrow v+a_j$.

   EndFor
3. **For** j=1,..,**N Do**
   - Denote with b_j the percentage of the recognition of the negative sentence-j, if the parser starts to match symbols from the most left element of the program.
   - $v=v \leftarrow 100*b_j$.

   EndFor
4. **Return** the value v as the fitness of the chromosome i.

3.4. Semi – partial parsing

The steps for the estimation of a chromosome i fitness, if the partial parsing method is used, are the following:

1. **Set** $v \leftarrow 0$.
2. **For** j=1,..,**P Do**
   - Denote with a_j the maximum percentage of the recognition of the positive sentence-j, if the parser starts to match symbols from any symbol of the sentence.
   - $V \leftarrow v+a_j$.

   EndFor
3. **For** j=1,..,**N Do**
   - Denote with b_j the percentage of the recognition of the negative sentence-j, if the parser starts to match symbols from any symbol of the sentence.
   - $V \leftarrow v+100*b_j$.

   EndFor
4. **Return** the value v as the fitness of the chromosome i.

4. Experimental Results

The proposed method is evaluated using 40 PUBMED abstracts extracted from molecular biology domain. From each abstract, the sentences describing protein relation is extracted manually. Each chromosome is evaluated using the partial parsing method.

The biological entities are shown in Table 1. The tagging process is completed by molecular biology experts. Additional semantic tags were used to annotate the extracted sentences by linguists. The complete set of semantic tags is shown in Table 2.

It is well-known that a grammar with too many free parameters cannot be estimated well from a relatively small set of training sequences. Attempts to estimate such a grammar will encounter the problem of overfitting, in which the grammar fits the training sequences well, but poorly fits related (test) sequences not included in the training set. One solution is to control the effective number of free parameters by regularization.

Table 1: Biological tags

BIOLOGICAL TAG	DESCRIPTION
PROTEIN	Genes and proteins
FUNCTION_1	Biological function (word end: -es)
FUNCTION_2	Biological function (word end: -ed)
FUNCTION_3	Biological function (word end: -ion)
SPECIES	Species
CTYPE	Cell type / Tissue
CLINE	Cell Line

The goal is to find those templates that can extract the protein relation information that is contained in the 40 samples. The samples are tokenized and their biological and semantic entities are tagged. Then they are split into 34 (used for training) and 6 (used for the evaluation).

The MetaGrammar is run iteratively until all samples are mapped onto a template. The 34 samples of the training set required 4 iterations. From the 4 iterations, 4 templates were extracted, which recognized 26 samples (8+8+4+6 respectively for each template). The remaining 8 samples were not recognized by any template. A sample is considered correctly recognized when the corresponding template can generate at least 70% of its length. According to this, the training error is 23.52%.

Table 2: Semantic tags

SEMANTIC TAG	DESCRIPTION
AND	And
BY	By
IN	In
THE	The
WORD	Common english word
BWORD	Other biology/chemical related

For the evaluation process a 4-fold evaluation test was performed, giving an overall test error of 20.835%. Samples taken from the original 40 abstracts that did not contain any protein relations were also tested against the templates extracted. None of the irrelevant samples were recognized by any template thus giving a recall rate of 100%.

5. Conclusions & Feature Work

An automated method for molecular biology information extraction from scientific abstracts has been presented. The proposed method requires minimum information in order to work and thus can be applied quickly to any problem. Field experts such as biologists for the present problem need only to mark the sentences that contain protein relations.

The method incorporates syntactic and semantic information in the recognition process and like all other NLP problems requires large corpora to give a detailed characterization of protein interactions. The results suggest that further work should be done especially with a larger corpus. Meta-Grammar is a relatively novel method and many improvements should be explored in the complex problem of modeling the molecular biology abstracts.

References

[1] Koza J., “Genetic Programming: On the programming of Computer by Means of Natural Selection”, MIT Press: Cambridge, MA, 1992.

[2] O'Neill M., and Ryan C., "Genetic code degeneracy: Implications for grammatical evolution and beyond," In Advances in Artificial Life, V.1674 of LNAI, Springer Verlag, p. 149, 1999.

[3] O'Neill R. and Ryan C., "Grammatical Evolution: Evolutionary Automatic Programming in an Arbitrary Language", V.4 of Genetic programming. Kluwer Academic Publishers, 2003.

[4] O'Neill R. and Ryan C., "Grammatical Evolution," IEEE Trans. Evolutionary Computation, Vol5, 349-358, 2001.

[5] Dempsey I., O'Neill M., Brabazon A., “Meta-grammar constant creation with grammatical evolution by grammatical evolution” 1665-1671, Electronic Edition (DOI: 10.1145/1068289) BibTeX

[6] Unger S.H., "A global parser for context-free phrase structure grammars", Commun. ACM, 11(4), p. 240-247, 1968.

[7] See-Kiong Ng, M. Wong, "Toward Routine Automatic Pathway Discovery from On-line Scientific Text Abstracts", Genome Inform Ser Workshop Genome Inform., 10: p. 104-112, 1999

[8] Gondy L., Hsinchun C., and Martinez J., "A shallow parser based on closed-class words to capture relations in biomedical text", J Biomed Inform. ,36(3), 145-58, 2003

[9] Skounakis M., Craven M., and Ray S., "Hierarchical Hidden Markov Models for Information Extraction", Proc. of the 17th International Joint Conference on Artificial Intelligence, IJCAI-2001

[10] 10. G. Demetriou, R. Gaizauskas, P. Artymiuk, and P. Willett., "Protein structures and information extraction from biological texts: The PASTA system.", Bioinformatics, 2002

[11] Candito, M. “A principle-based hierarchical representation of LTAGs”, In COLING-96, 1996

[12] Clément L., and Kinyon A., "Automating the generation of a wide-coverage LFG for French using a MetaGrammar", Proc. of Formal Grammar, C3, 33-46, 2003

[13] Horning J. "A procedure for grammatical inference". Proc. of IFIP Congress 71, Vol. 1, 519-523, Amsterdam. North-Holland, 1972.

[14] Crabb´e B., and Duchier D., "Metagrammar Redux", 43-58, CSLP 2004, Roskilde University, 2004

[15] Duchier, D.: MOGUL: the MOzart Global User Library (2004), http://www.mozart-oz.org/mogul/

[16] Gaiffe, B., Crabbe B., and Roussanaly A., “A new metagrammar compiler”, In Proc. TAG+6. Venice, 2002

[17] Clément, L. and Kinyon A.,”Generating LFGs with a MetaGrammar”. In Proc. LFG-03, 2003.

[18] Clément, L. and Kinyon A., “Generating parallel multilingual LFG-TAG grammars with a MetaGrammar”, In Proc. ACL-03, 2003.

[19] Dekang Lin and Pantel P., "Discovery of inference rules for question answering", Natural Language Engineering 7(4), 343-360, 2001

[20] Dempsey I., O’Neill M., and Brabazon A., “metaGrammar Constant Creation with Grammatical Evolution by Grammatical Evolution”, GECCO’05, 1665-1671, 2005

[21] O'Neill M., and Brabazon A. “mGGA: The meta-Grammar Genetic Algorithm”, EuroGP2005 & EvoCOP2005, EvoWorkshops2005, 2005