Journal : SmallExtended 10710 | Article No : 9361 | Pages : 22 | MS Code : 9361 | Dispatch : 20-7-2019

Genetic Programming and Evolvable Machines
https://doi.org/10.1007/s10710-019-09361-5

Genetic programming for natural language processing

Lourdes Araujo[1,2]

Received: 15 October 2018 / Revised: 3 April 2019

Abstract

This work takes us through the literature on applications of genetic programming to problems of natural language processing. The purpose of natural language processing is to allow us to communicate with computers in natural language. Among the problems addressed in the area is, for example, the extraction of information, which draws relevant data from unstructured texts written in natural language. There are AQ1
also domains of application of particular relevance because of the difficulty in dealing with the corresponding documents, such as opinion mining in social networks, or because of the need for high precision in the information extracted, such as the biomedical domain. There have been proposals to apply genetic programming techniques in several of these areas. This tour allows us to observe the potential—not yet fully exploited—of such applications. We also review some cases in which genetic programming can provide information that is absent from other approaches, revealing its ability to provide easy to interpret results, in form of programs or functions. Finally, we identify some important challenges in the area. AQ2

Keywords Genetic programming · Grammatical evolution · Natural language processing · Applications · Challenges

1 Introduction

This article reviews some applications of techniques based on genetic programming and grammatical evolution to some of the main areas of NLP. It is not intended to be an exhaustive sample of the variety and importance of the applications of these techniques to natural language processing (NLP) tasks. Under the name of genetic programming (GP) [41] there is a class of evolutionary algorithms that evolve programs

✉ Lourdes Araujo
lurdes@lsi.uned.es

1 Natural Language Processing and Information Retrieval Group, Universidad Nacional de Educación a Distancia (UNED), Madrid, Spain

2 IMIENS: Instituto Mixto de Investigación, Escuela Nacional de Sanidad, Monforte de Lemos 5, 28019 Madrid, Spain

or functions usually represented as parse trees of variable size. Typical GP operators swap sub-trees between two parents, delete sub-trees in a parent, or perform changes at the nodes. Grammatical evolution (GE) [58, 59, 65] is a variant of GP that evolves individuals' genotypes represented as integer strings. To compute the fitness, the genotype is mapped to the phenotype or parse tree by means of a Backus–Naur form (BNF) grammar. The integer representation simplifies the application of the genetic operators.

Evolutionary algorithms in general have been applied to different NLP [2, 4] and information retrieval [20] tasks. However, they are not as many as one could expect from the complementary nature of GP techniques and NLP problems. This complementarity [2] relies on several facts. Statistical methods have become a fundamental approach to computational linguistics, bringing significant advances in tasks such as disambiguation, parsing or grammar induction. These methods are formulated as statistical models to be optimized, thus providing a natural fitness function when the problems are tackled with evolutionary algorithms. In addition, GP provides a natural way to integrate data representing the linguistic model as test cases. On the one hand, GP has been successfully applied to many classification problems [22], so it can also be applied to NLP tasks, which often involve classification problems. On the other hand, many NLP tasks are addressed by building rules more or less automatically, and GP has proven to have a great potential in generating rules for many problems. In fact, it is probably the most frequent application of GP to NLP.

The somewhat limited number of GP applications is possibly due to efficiency issues. GP requires evolving a population of complex structures. The computation of the fitness function for a set of training data is usually also a time-consuming process. However, GP has important advantages over other ML methods. One of them is the interpretability of the results. In general, it is important to understand the mechanisms a system has followed to achieve its results because this provides insights for further improvements. In addition, this is essential in some applications. For example, when extracting information in the biomedical domain, health care professionals need to know on which data the system's predictions are grounded, in order to evaluate its reliability. As a matter of fact, many NLP systems combine ML and rule-based techniques. For these reasons, and also because of an increasing computing capability, we can foresee an increase of GP applications to NLP.

In this work we review a few of these applications. Far from being exhaustive, we try to illustrate the NLP problems in which GP techniques have been most commonly applied. First of all, we review some of the first works where GP was applied to NLP problems. There are mainly related to the identification of the syntactic structure of the natural language. Later, we focus on what probably is the main area of application: extraction of information from documents. It includes works related to several of the main aspects of this topic, such as named entities recognition (NER), relationship extraction, and entity linking. Afterwards we review some representative works in natural language generation. Finally, we devote a section to some real-world applications of NLP that in some cases have been addressed using GP: detection of spam, opinion mining, and applications to the biomedical domain.

Figure 1 shows a scheme of the main topics covered in this review. There are two main areas of research in NLP. One of them is natural language understanding

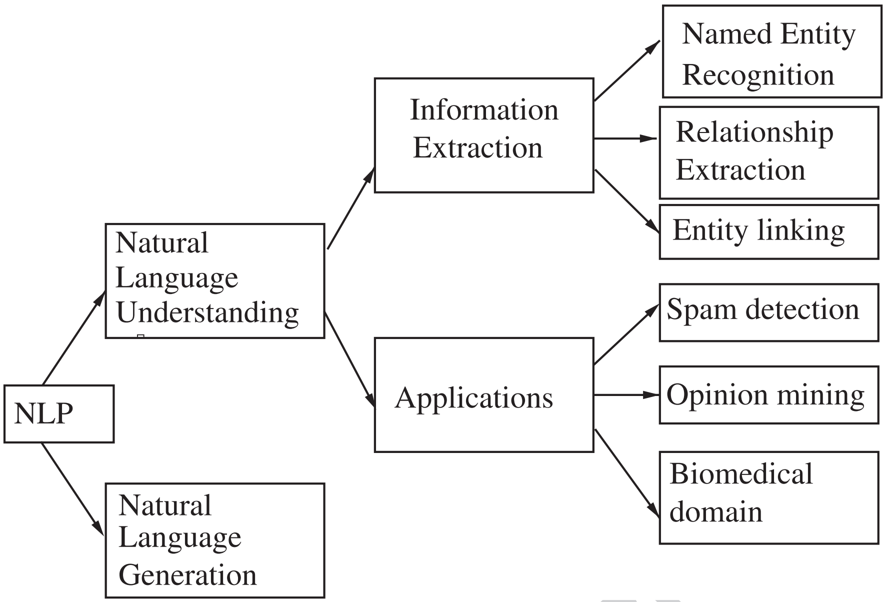

Fig. 1 Scheme of the main NLP topics with GP applications considered in this article

(NLU), often referred simply as NLP, which attempts to understand the meaning behind a text, and produces some kind of structured data with the information identified in the text. The other area is natural language generation (NLG), which starting from data tries to reflect them in a well written text. Most works in NLP are focused in understanding, although there are also many proposals for NLG. The NLU area comprises many important NLP tasks, such as parsing, word sense disambiguation, document classification, or information extraction, which in turn cover other sub-tasks. Most of them are applied to solve practical problems in the real world. The selection of topics is based on two basic requirements. The first one is the relevance of the topic itself in NLP. The second one is that the corresponding problem has been approached with several proposals based on GP. Regarding the applications of NLP to practical problems, apart from the previous conditions, it has also been taken into account that they were hot problems in the area of NLP, or that a particularly high number of contributions used GP to solve it. Opinion mining and information extraction in the biomedical domain are in the first case. Opinion mining is one of the main interest of the companies providing products based on NLP. For example, they offer other companies different ways of monitoring the opinion about their products. Applications to the biomedical domain has also become one of the main areas of application of NLP techniques due to the huge amount of text documents containing information relevant to health care. Spam detection, in addition to being a relevant topic in the field, seems to be a problem particularly well suited for applying GP, given the number of related proposals.

An area related to NLP is information retrieval (IR). IR seeks to recover from a collection a subset of relevant documents for a query, ranking them according to their relevance for the query, and is usually based on key-word search process. It differs from information extraction, which aims to extract from the content of a document or set of documents the salient facts, entity mentions or relationships. Some applications of GP to IR have been treated in a previous survey [20], and are not included in this review.

2 First applications of GP to NLP

Let us considered some of the first applications of GP to NLP. They were mainly related to the identification of the syntactic structure of the language, including topics like grammar induction, and natural language parsing. The fact that most of them dealt with syntactic aspects of the language is probably due to the analogy between the usual representation of the language syntax as a tree and the representation adopted in GP.

Grammar induction (GI) aims to learn the grammar underlying a collection of sentences. This process has many applications such as syntactic pattern recognition and machine translation. In turn, natural language parsing amounts to break down a sentence into groups of words with a particular linguistic function, such as subject or object of a verb, and to establish the relationship among those parts. Because the most natural data structures for representing this organization are trees, GP can be considered an appealing technique for dealing with the process of generating them. And actually several works have been proposed along this line.

Smith and Witten [67] proposed an evolutionary algorithm for GI which evolved a population of context-free grammars (CFG) represented as LISP AND-OR s-expressions. An example of grammar is:

(AND (OR a the)(OR dog cat)(OR saw bit) (OR a the)(OR dog cat))

which is able to parse a sentence like “the dog saw a cat” considered in this work.

Individuals were selected for reproduction and mutation in proportion to their size. The fitness of the whole population was measured by its ability to parse a training set. The system was able to infer simple natural language grammars for a small set of training examples. Some years later, Korkmaz and Ucoluk [39] presented another work which aimed to guide the recombination process by extracting global information from the potential solutions. This was done by introducing a control module which ran a classification algorithm to determine valid and invalid chromosomes. Experiments showed that the controlled search had a better performance compared to the straightforward application of GP.

Another early work related to parsing was developed by Rosé [63] who used GP to aid in recovery from parser failure in speech-to-speech machine translation. Araujo [1] proposed a GP system for natural language parsing which implemented a probabilistic bottom-up parser and evolved a population of partial parses. The proposal was extended in a later work [3] for performing Part-Of-Speech (POS) tagging and parsing simultaneously applying multiobjective GP to deal with both problems.

These works show the structural affinity between GP techniques and NLP problems concerning syntax, which suggests the simplicity for combining them.

Journal : SmallExtended 10710 | Article No : 9361 | Pages : 22 | MS Code : 9361 | Dispatch : 20-7-2019

However, despite the promising results obtained by these systems for some natural language fragments, their efficiency was limited by the GP computational cost, increased by the cost associated to the size of the grammar underlying the considered fragment of the language.

2.1 Summary

Table 1 shows the main features and the publication years of some of the first works applying GP to NLP. We can observe that from the very early days of GP, NLP-related works, such as the one by Rosé [63], began to appear. We can see that these works present some common features, such as the way of evaluating individuals, that in most cases amounts to comparing the tree representing the grammar or the parsing with a reference model for the sentences considered.

3 Information extraction

The aim of information extraction (IE) [36] is to convert unstructured information from texts into structured data, so that it can be easily used by other processes. One of the main tasks involved in IE is named entity recognition (NER). It amounts to identifying those words or phrases that correspond to a particular kind of concept (person names, organizations, diseases, etc.). Some kinds of entities that have been often considered are people, organizations, locations, diseases, drugs, genes, proteins, etc. Once the entities have been located, another interesting problem is extracting the relationships between them. In addition, sometimes the same concepts are referred in different sources in different manners. Because of this, the search for the links between entities have also been studied in various works.

One of the reasons behind the complementarity of GP and IE is the GP ability to find suitable patterns or functions to solve a problem. Most popular methods in IE are ML and pattern extraction, sometimes used together. Indirectly, GP has proven useful in improving ML systems, for example for feature selection. However, here

Table 1 Main features of the first GP applications to NLP

Topic	Proposal	Main features	Year
Grammatical induction	Smith and Witten	Grammars represented as LISP AND-OR s-expressions, CFG	1995
	Korkmaz and Ucoluk	GP-trees represented as vectors, allows combination of CFG	2004
Parsing	Rosé	LISP s-expressions	1990
	Araujo	Partial parse trees	2004
	Araujo	Partial parse trees, (multiobj.:NSGA)	2006

For each work, the third column indicates the representation of individuals used by the algorithm. The last row, for Araujo [3], indicates that the work applied a multiobjective approach. The fitness used in these works is based on comparing the individual with a reference standard for the parsing of the sentences considered

Journal : SmallExtended 10710 | Article No : 9361 | Pages : 22 | MS Code : 9361 | Dispatch : 20-7-2019

we focus on direct applications to IE, which are typically based on identifying patterns or regular expressions that characterize the information to be extracted.

3.1 Named entity recognition

NER is frequently approached via supervised techniques, such as ML [56], and more recently, deep learning [18]. However, there are other approaches that attempt to capture the patterns associated with the type of entities considered. Regular expressions are one of the main approaches to NER. Several works have investigated the application of GP or GE to the problem of identifying regular expressions in documents. A regular expression or *regex* is a way for representing string patterns precisely. They have multiple applications in tasks related to natural language, such as information search, data validation, and parsing. And they are the common way of representing the patterns for applications related to NER. Because generating the regex for a specific task is a highly complex and error-prone process, a number of approaches have been proposed for generating them automatically.

Gonzalez-Pardo and Camacho [27] applied GE for extracting regex matching url patterns. Grammatical evolution using CFGs may generate semantically incorrect individuals, because CFGs do not consider the context of the replaced non-terminal symbols. To deal with this problem, these authors evaluated four types of grammars: CFGs, CFGs with a penalized fitness function, extensible CFGs, and Christiansen grammars. In Christiansen grammars (CG) [16, 60] non-terminals have a set of attributes, each of them with a name and a value. The rules contain expressions to compute the value of the attributes, and allow the grammar to be modify during the evolution process, for example by adding or deleting rules. The best results were achieved using a Christiansen grammar.

Bartoli et al. [6] applied GP for extracting regex devoted to entity extraction applications. The system applied multiobjective GP to generate a regex for a task specified by a set of examples. Candidate solutions were represented as syntax trees where internal nodes were assigned regex operators. The adopted multiobjective approach was Non-Dominated Sorting Genetic Algorithm II (NSGA-II) which is used to optimize two fitness functions. One of these functions was the edit distance (minimum number of operations required to transform one string into the other) and the other one was the length of the regex. The proposal was evaluated on 12 extraction tasks including email addresses, IP addresses, web URLs, HTML headings, Twitter hashtags, and citations. The authors reported results of precision and recall that compare favorably with those of previous systems using the same data. In later works, Bartoli et al. [9] proposed an active learning approach in which the user acts as an oracle. Initially the user presents a few snippets to the system indicating the entities to be extracted. Then, a learner based on GP builds a solution, in the form of a regex, and examines the input text, selecting the most promising snippets according to the current model. Selected snippets are presented to the user, who indicates if they should be extracted or not. The system was evaluated on the datasets used in [6] and the authors concluded that active learning, starting with only one annotated match, is a viable approach for the considered application, and that it significantly decreases the required amount of user annotation. Bartoli et al. [8] have also explored the use of GE to the problem. Specifically, they considered the problem of learning similarity functions useful for syntax-based entity extraction from unstructured text streams. The input to the algorithm are pairs of strings and an indication of whether they correspond to the same syntactic pattern.

One important area of application of NLP techniques is the biomedical domain [14, 17, 32]. Because of the huge amount of documents produced in this domain, including scientific articles, and medical reports, IE techniques are needed to process them, and to exploit the knowledge they contain. As mentioned above a first step of the process is applying NER techniques.

Korkontzelos et al. [40] applied GP to reduce the amount of annotated data required to train a NER system. They proposed a voting system able to combine predictions from several recognizers. The system was evaluated on the PharmacoKinetic Corpus (PK corpus) [73], manually annotated and composed of 240 MEDLINE abstracts annotated with drug names, enzyme names and pharmacokinetic parameters. Results show that the system achieves state-of-the-art precision, but lower values of recall. In a second phase, and in order to improve recall, the authors applied GP to generate string patterns that can then be used as regex to capture additional drug names.

3.2 Relationships identification

Interesting patterns appear in the identification of relevant relationships in different domains. One of these domains is the biomedical one, although it is not the only one. Some of the relationships considered in this domain are protein–protein interaction [42], drugs and genes [61], drugs and adverse effects [44], and rare diseases and disabilities [23]. Most of these works focus on solving the problem at sentence level, i.e. they do not consider relationships between entities appearing in different sentences. Dealing with this problem requires both, identifying the entities that can be related, and verifying the existence of a relationship between the entities found, since the occurrence of two entities in the same sentence does not imply a relationship. Most works tackle both problems separately, or assume that the entities have been previously annotated, either manually or automatically. Many systems in the area have followed a supervised approach, applying different classifiers, and recently deep learning techniques [23, 44, 53] to the problem. There are however some interesting proposals considering GP, that apart from competitive results can provide more informative solutions.

In an early work on the subject, Bergström et al. [11] applied GP to find semantic relationships in texts from the web. They focused on the hyponym relation between nouns, i.e. a subordinate relation among nouns. Individuals in the population were syntactic trees. The fitness function was computed as the rate of related pairs of words that the individual captures according to Wordnet [52], a dictionary of nouns, verbs, adjectives and adverbs which organizes related concepts into synonym sets, representing concepts. The system was able to provide patterns that detect simple types of hyponyms.

Journal : SmallExtended 10710	Article No : 9361	Pages : 22	MS Code : 9361	Dispatch : 20-7-2019

More recently, in the biomedical domain, Bartoli et al. [7] have applied GP to identify sentences that contain descriptions of interactions between genes and proteins. Specifically, they used GP to obtain a model of syntax patterns composed of part-of-speech (POS) tags. The model consisted of a set of automatically learned regex. They used a dictionary of genes and proteins for the detection of entities. The system was evaluated on 456 sentences obtained from two corpora, both derived from genic–protein interactions extraction challenges in the biomedical domain. The GP system obtained an accuracy similar to the one reached by other methods used as baseline in the work. Also in the biomedical domain, Bootkrajang et al. [12] applied an evolutionary hypernetwork classifier to protein–protein interaction (PPI) sentence classification, i.e. to identify the text sentences in a dataset that mention a PPI. The authors stated that the proposed model provided good performance compared to ML systems, such as Naive Bayes and SVM.

3.3 Entity linking

Another task usually included in IE is entity linking (EL). EL refers to identify and connect the different ways in which the same entity is mentioned in the texts. EL is usually carried out by resorting to knowledge bases containing the entities corresponding to the different entity mentions. These knowledge bases may be organized as taxonomies or ontologies. EL helps to improve the performance of information retrieval systems as well as the search performance in document repositories. GP has been used in several works related to this task.

Carvalho et al. [21] applied GP to find effective deduplication functions, i.e. functions able to identify in a data repository entries referring to the same entity in spite of misspelling words, typos, or different writing styles. This problem has received a lot of attention since the presence of dirty data in the repositories degrades the performance. This work presented experiments on real data sets containing scientific article citations and restaurant catalog records. The authors showed that their approach was able to improve the results of a state-of-the-art SVM based approach. Later, Isele and Bizer [35] proposed other system on the subject trying to improve the previous results. They presented the ActiveGenLink algorithm, combining GP and active learning to learn linkage rules which included data transformations.

Tiddi et al. [70] used GP to search for a cost function able to detect the strength of the relationship between two given entities. Relationships in a Web of data can be represented as paths in the graph of linked data. This work builds and selects the functions that best perform in ranking sets of alternative relationship paths. The functions represented by the individuals are created on a set of features related to possible topological or semantic properties of the nodes and edges of the graph.

A topic related to EL is the construction of taxonomies and ontologies. Domain specific information is usually arranged hierarchically as taxonomies and ontologies. An example is the Hermes ontology [26], which is composed of concepts from the financial domain and is used in news classification and querying. These ontologies need to be kept up-to-date in an efficient way, and to include new selection patterns to extract concepts from new documents. IJntema et al. proposed the lexico-semantic Hermes Information Extraction Language (HIEL) [34], to include semantic elements in the extracted patterns. Because the construction of such patterns is a difficult task, in a later work [33] they proposed to apply GP for helping to build IE rules in the financial domain. Individuals are the tree structures used by the HIEL language. Fitness is provided as the F_measure computed by comparing the extracted information with manually annotated information. Araujo et al. [5] also presented a work devoted to generate hierarchical structures, or taxonomies from concepts from the Wikipedia by applying GE. Each Wikipedia article is assigned a topic and is linked by hyperlinks that connect related topics. The goal of this work was to identify taxonomies of concepts associated to linked Wikipedia pages. This was done by searching for functions that combine a set of features extracted from the contents of the Wikipedia pages.

Although in some cases the results reached by the mentioned works may be a bit lower than others using machine and deep learning approaches, they have the important advantage of supplying information about the knowledge used by the system to get the results. This information may be for example the elements that are part of the optimal program or function obtained by the GP algorithm. In many cases this information is of paramount importance for trusting in the results. For example, doctors need to know the knowledge applied by a system to provide relationships between medical entities to be able to rely on them for making diagnoses or prescribing treatments.

3.4 Summary

Table 2 presents a summary of the contributions mentioned in this section. Looking at the publication years, we can observe that, in general, these works are much more recent than those presented in the previous section, the latest publications being as recent as 2018. The third column in the table, describing the main features of the proposals, contains different information depending on the topic considered. The top part of the table, devoted to NER, includes works for different applications: url pattern, drug concepts, and the three works by Bartoli et al., evaluated on a set of different extraction tasks including email addresses, IP addresses, web URLs, HTML headings, Twitter hashtags, and citations. Although these last works have the same authors, all of them have been included because they are substantially different proposals. The first one proposes the use of regex and a multiobjective algorithm, the second one is devoted to obtaining functions for the computation of the similarity between regex using GE, and finally the last one left the evaluation to the very user. The central part of the table contains works focused on the extraction of relations. The table shows the concepts involved in the relationship sought. The bottom part of the table comprises several works devoted to entity linking. The table indicates the main goal of the work: looking for functions that identify duplicated entities [21], looking for connection between web entities [35], looking for functions to compute the strength of a relationship [70] or for building taxonomies, in the financial domain [33] or between Wikipedia pages [5], in this case using GE.

Table 2 Main features of GP applications to information extraction problems

Topic	Proposal	Main features	Year
Named entity recognition	Gonzalez-Pardo and Camacho	(Url patterns) (regex)	2011
	Korkontzelos et al.	Drug concepts (regex)	2015
	Bartoli et al.	Several concepts (regex) (multiobj.)	2014
	Bartoli et al.	Several concepts (encode similarity functions) (GE)	2016
	Bartoli et al.	Several concepts (regex) (user as oracle)	2018
Relationship identification	Bergström et al.	Hyponym relation	2000
	Bartoli et al.	(Genes and proteins) (regex)	2015
	Bootkrajang et al.	(Protein–protein) (evol. hypernetwork classifier)	2009
Entity linking	Carvalho et al.	Deduplication functions	2012
	Isele and Bizer	Web entities	2013
	Tiddi et al.	Strength of the relationship	2016
	IJntema et al.	Taxonomies (financial domain)	2012
	Araujo et al.	Taxonomies (Wikipedia pages), GE	2018

Journal : SmallExtended 10710 | Article No : 9361 | Pages : 22 | MS Code : 9361 | Dispatch : 20-7-2019

4 Natural language generation

Another well suited area for the application of GP is natural language generation (NLG). NLG aims to produce natural language text from the computer representation of information. The traditional approach to NLG is based on grammars and templates. Templates are particularly popular because of their simplicity. However, the design of templates (or grammars) able to generate high quality text, while preventing the generation of wrong sentences, is a difficult task. NLG [51] is an active area of research, with different applications, such as dialog systems. Recent proposals look for including mechanisms to enhance novelty in the generated texts [62] and GP may be one of these mechanisms.

One of the first works on this topic was proposed by Manurung [48] to generate poetry. The system, named McGONAGALL, characterized poetry by three features: meaningfulness, grammaticality and poeticness, which considered aspects such as metric and rhyme. The system was able to produce a text almost metrically perfect. An example of generated poetry is the following:

with a bandy very large waste with
the platinum lion , the mind is his waste with
the product . in a boy , with a african pole in
his bill with his whiskers , his platinum toad
in his bill in her dwells in his bean . his hippopotamus
will be the frog in a african child
in a soil with the fish with the tiger with the
grin in his bean.

In a later work by different authors, Manurung et al. [49] developed a GP system aiming to generate text presenting certain meter or patterns in the rhythm.

Another related area of application are conversational systems. Kim et al. [38] proposed a GP system to generate the answers that a conversational agent provides to user's queries. The system performs several preprocessing steps including keyword extraction. In this work, keywords are words appearing frequently on the particular domain. Keywords extracted from the query are compared with keywords in answer-scripts. Individuals in the GP population were trees representing patterns corresponding to Korean grammar structures. Fitness was computed by an interactive evaluation [69], in which the user was asked to provide a score for the generated replies. The authors claim that the replies of an agent introducing a fashion web site were more natural than those of other proposals. In a later paper, Lin and Cho [45] also proposed interactive GP for generating replies. In this case, instead of using grammars for encoding the trees in the population, the authors proposed sentence plan trees, trying to reduce the convergence time. Plan trees are binary trees whose leaves are labeled by pre-defined templates of simple sentences. The internal nodes were labeled with different joint operators, that allow to combine sentences.

Except for very restricted contexts, NLG still remains a hard task in the NLP area. There is not only a need of expressing a given content as a grammatically correct text, but it also has to be done in a natural and fluid way. Given the possibilities of GP to select individuals taking into account novelty and diversity, its application to NLG can be very interesting. However, the number of works in this line is still quite limited. NLG systems still need to reach a higher level of maturity to extend their use. The application of GP for building these systems will be conditioned to the amount of research devoted to design more sophisticated NLG systems in general.

4.1 Summary

Table 3 summarizes the selected works devoted to natural language generation. We can observe that they are not very recent, the last of them being published in 2008. The second column of the table, devoted to the specific domain for which the text is generated, indicates that they are very specific domains, such as poetry [48, 49], or question answering [38, 45]. Concerning the representation of individuals, the mentioned works use one form or another of trees representing grammars.

5 NLP applications

This section includes some works focused on specific problems of high interest, where NLP techniques have proven very useful. Among them are the detection of spam in emails, the mining of opinions, that allows a company to know the customers' satisfaction with a service or product, and applications to medicine.

5.1 Spam detection

The huge amount of spam, i.e. unsolicited electronic mails or text messages that are sent on the Internet, has made anti-spam filtering an active area of research. Spammers use a large number of different strategies to send illegal and fraudulent messages. This leads to anti-spam filters needing to be continually revised and updated to be adapted to new forms of attack [37]. The problem has been addressed with ML techniques, collaborative schemes and also by the identification of regex appearing in spam messages. Actually, popular anti-spam frameworks such as SpamAssassin [66] allow users to define regex to improve the system filtering. This is why it is so useful to automatically generate anti-spam filtering rules. Here some of them based on GP are considered.

Table 3 Main features of works applying GP to natural language generation

Proposal	Main features	Year
Manurung	Poetry	2003
Kim et al.	Conversational agents	2004
Lin and Cho	Reply generation	2005
Manurung et al.	Poetry	2008

Greenstadt and Kaminsky [30] were the first to propose the use of GP to generate regex for spam filtering. They performed different experiments to evaluate their system. The fitness function was a linear combination of the number of legitimate messages that match the regex and the number of spam e-mails that do not match with the regex. The evaluation was carried out on a small set of email messages and some false positive cases were detected. Conrad [19] tried to improve the previous proposal by defining a fitness function favoring those regex with minimal length and maximizing the matching with spam samples. This GP system, called GenRegex, generated a set of Perl Regular Expressions from spam and ham messages. In a later work, Basto-Fernades et al. [10] proposed the use of GE for the problem. They applied a multiobjective evolutionary mechanism, instead of linear combinations of the measures to be optimized in a single function. In a recent work, Ruano-Ordás et al. [64] have proposed DiscoverRegex. This system tries to avoid problems of the previous proposals, such as the minimizations of the length of the generated regex, that can exclude useful solutions. They also tried the reduce the generation of inefficient regex. Their proposal combines improvements in the evaluation of candidate regex and mechanisms to avoid the evaluation of a pattern more than once.

5.2 Opinion mining

Opinion mining [46], also known as sentiment analysis, refers to applying NLP techniques to study the attitude of the author of a text. Its purpose is to determine if the text expresses an emotion and whether it is positive or negative, as well as its intensity. Another related task is subjectivity analysis that discriminates the objective or subjective nature of a text. There are several aspects that made this problem difficult. For example, a word such as “low” may be associate with a positive opinion, as it happens in “low noise”, or to a negative one, as in “low performance”. Other difficulties come from the presence of negation, and speculation, that can change the sense of the words. Another problem is the fact that the same text can express positive and negative opinions regarding different aspects of the same product. This area has received a lot of attention in recent years due to the great relevance that has for companies that want to know the market response to their products, advertisements, etc. There have been some proposals applying GP to deal with it.

Graff et al. [28] applied semantic GP [55] to the problem. The key idea of this approach was creating the best offspring that can be produced by a linear combination of the parents. The system was tested on the data provided by an NLP evaluation campaign on sentiment analysis, TASS15, hold in 2015 [71]. According to the authors, the system reached results competitive with the performance of state-of-the-art classifiers. Moctezuma et al. [54] took part in the 2017 edition of the TASS campaign [50] using GP. Their approach was based on distant supervision, increasing the training data with new data labeled without human assistance. This was done by means of a set of heuristics based on dictionaries. Then, they used a set of classic classifiers trained with the two kind of datasets. Finally, they applied a GP system that combines all the decision values predicted by the classic classifiers. Specifically, the authors use EvoDag [29], a semantic

Journal : SmallExtended 10710 | Article No : 9361 | Pages : 22 | MS Code : 9361 | Dispatch : 20-7-2019

genetic programming python library. Winkler et al. [72] tested a number of ML methods, including GP, to identify the sentiment of sentences available in a German corpus of Amazon. The considered methods were decision trees and adaptive boosting, Gaussian processes, random forests, k-nearest neighbor classification, support vector machines, artificial neural networks, and GP. They found that a combination of classifiers was able to increase significantly the classification accuracy. But additionally, considering the results of the classifiers separately, GP was among the best ones.

5.3 Biomedical domain

The biomedical domain generates a large amount of information, including medical records, that is of high relevance to both health professionals and citizens. This has motivated great interest in the development of NLP techniques to process this information. These techniques will assist in tasks like clinical decision support (CDS) [14, 17], which helps health care professionals and citizens to make decisions by providing easily accessible health-related information. Among the documents considered in this domain are both, medical reports and scientific articles, that have very different nature. Several related works have already been mentioned in the section dedicated to IE. The availability of all these health data offers an unique opportunity to develop methods for extracting relationships among medical concepts, that can help to make diagnoses or predict adverse drugs effects, for example. We mention here some additional works related to the need of building systems that report on their behaviors.

Holzinger et al. [31] have presented an interesting study addressing the need in the medical domain of making predictions re-traceable in such a way that health care professional knew where the machine decisions come from. They mentioned a number of attempts of connecting the large databases of structured knowledge, such as the Unified Medical Language System (UMLS), with the distributional models, such as dense vector representations or embeddings. A path of research along this line is the integration of the interpretability of knowledge-based systems and the efficiency of neural approaches. There are some proposals along this line, like the one by Faruqui et al. [25] that proposed retrofitting neural embeddings with information from knowledge bases, or Faralli et al. [24] that suggested linking dense vector representations to lexical resources and knowledge bases. GP can be a useful alternative to build easy to interpret systems and to integrate different technologies in this domain. Interpretability in GP systems comes from the data and operators included in the program or function selected as the best solution.

There are also works, such as the one by Brameier and Banzhaf [13] comparing the performance of GP and neural networks, that have shown that GP is able to reach similar performances in classification and generalization in a number of problems related to diagnosis. Although the considered problems were not related to NLP tasks, these results indicate the ability of GP to reach comparable performance to neural networks.

Journal : SmallExtended 10710 | Article No : 9361 | Pages : 22 | MS Code : 9361 | Dispatch : 20-7-2019

5.4 Summary

Table 4 shows selected works applying GP to different real world problems. Although some of these works appeared quite a few years ago, most are recent. Works at the top part of the table are devoted to spam detection. A commonality they share is to use GP to generate regular expressions. This is one of the most common ways to address the problem, because it allows the solution to be adapted to the specific context being considered. GP, or GE in the case of [10], are used to look for an appropriate regex.

The central part of the table is devoted to opinion mining. The first two articles study the polarity in a Twitter dataset, while the third is evaluated on Amazon product reviews. In this topic, GP is applied in different ways, going from semantic GP to classification.

Finally, the bottom part of the table considers some works related to the biomedical domain. One of them [13] applies GP for classification in problems related to diagnosis. The other one [40] uses GP for generating a regex able to detect drug names in texts. We can observe that there are few works in this area, despite its relevance.

6 Opportunities

GP and GE have been applied to many different NLP problems, providing solutions different from those obtained with other more popular approaches in ML, such as classifiers (SVM, decision trees, etc.). There are several distinguishing features of evolutionary techniques, and of GP in particular, which make them particularly suitable for some applications.

Two of the most differentiating characteristics of evolutionary techniques are their ability to generate rules to solve specific problems, and their ability to generate diversity. Both features are fundamental for achieving improvements in many problems addressed in the NLP area. As a matter of fact, one line of research in which

Table 4 Main features of works applying GP to the considered NLP applications

Topic	Proposal	Main features	Year
Spam detection	Greenstadt and Kaminsky	(Regex)	2002
	Conrad	(Regex)	2007
	Basto-Fernades et al.	(Regex)(GE) (multiobjective)	2014
	Ruano-Ordás et al.	(Regex)	2018
Opinion mining	Graff et al.	Polarity in a tweet dataset (semantic GP)	2015
	Moctezuma et al.	Polarity in a tweet dataset (GP for emsamblig SVM classif.)	2017
	Winkler et al.	Amazon product reviews (classif. combination)	2015
Biomedical domain	Brameier and Banzhaf	Classif. for diagnosis	2015
	Korkontzelos et al.	NER biomedical	2015

Journal : SmallExtended 10710 | Article No : 9361 | Pages : 22 | MS Code : 9361 | Dispatch : 20-7-2019

GP could make very valuable contributions is the discovery of new knowledge. There are areas in which the great amount of available data makes it difficult to mine relationships mentioned in the data. An example is the biomedical area, in which the search for relationships between concepts, such as diseases and genes, interactions between proteins, drugs and adverse effects, etc., has great relevance. Some of these connections can be found in the texts, and NLP techniques aim at extracting them; but in other cases it is possible to identify hints of relationships between concepts that do not appear explicitly in the documents. They can be inferred, for example, by identifying certain patterns, and can lead to new knowledge. GP can be a way to pursue this search, since the programs, rules or functions given as solutions provide clues about new possible connections.

Another reason that makes the use of GP specially attractive for NLP problems is that it can provide results that can be interpreted more easily than those provided by other approaches. Neural networks have yielded a dramatic improvement on many problems, quite a few of them in the NLP area. However, the black box nature of these systems can limit the acceptance of their results. Many applications, such as those in the medical or financial domains, require an interpretation of the results and predictions of the system. In contrast, GP algorithms provide programs, rules or functions that are easy to interpret. They are composed of operators and data, which human beings can understand.

Evolutionary algorithms provide great adaptability, allowing the programmer to easily incorporate specific knowledge about a problem. This can be done in different elements of the algorithm, such as the representation of individuals, the fitness function, or the genetic operators. All these elements are frequently defined specifically for the considered problem, thus allowing to take advantage of all the available knowledge. In many NLP applications such knowledge is available. This knowledge, which allows NLP researchers to craft rules tailored to the specific framework of a problem, can be introduced into the design of evolutionary algorithms more easily than in other methods. For example, parsing systems based on GP [1] can easily include constraints on the size of the parse trees, based on linguistic knowledge of the most frequent forms of these trees. Similarly, systems generating regular expressions for detecting spam messages [19] also use knowledge on the problem by defining fitness functions that favor those expressions with a particular length range. In fact, rule-based and heuristic systems are quite popular in the NLP field. There are many problems for which the best solutions are achieved by hand-generated rules. There are different reasons for this. One may be the lack of sufficient data to generate well-trained ML models. But even in the presence of a large amount of data, the problem can arise from the huge amount of classes to classify them in, as it happens in problems like the assignment of medical codes to medical records (for example, ICD10 for diagnostic coding is composed of 68,000 different codes). In these cases, heuristics designed to fit the particular data may achieved the best results. As the hand-generation of rules is a difficult and expensive task, GP is an alternative to explore in all these cases.

There is another important reason that makes the complementarity of these two areas appealing, and offers an opportunity for the development of GP-based systems. This is the availability of evaluation data for a number of NLP tasks. For a long time, a number of evaluation campaigns or shared-tasks related to many NLP problems have been organized. The organizers of these evaluation campaigns define a precise framework in which to address a particular problem and provide training and test data. This way, the teams participating in the competition can compare their methods and results. Some of the best known organizations of competitions are TREC (https://trec.nist.gov/), CLEF (http://www.clef-initiative.eu/), SemEval (http://alt.qcri.org/semeval2018/) or SemEval (https://aclweb.org/aclwiki/SemEval_Portal). Many problems have been addressed in these campaigns: entity recognition, extraction of relationships, word sense disambiguation, opinion mining, etc. In addition, in the area of NLP it is common to develop collections of data (corpora) manually annotated by experts, which are used as reference for the development of systems. This availability of data makes NLP problems an attractive field in which GP techniques can be evaluated, as well as a major challenge due to the difficulty of working with real and extensive data.

Finally, in addition to direct applications of this type of heuristics to NLP there is also an indirect relationship. Machine learning (ML) techniques, are currently among the most popular in NLP [47]. Recently, as it has happened in other areas, there has been a explosion of applications of deep learning to NLP problems [18, 75]. These applications include machine translation [43] and named entities recognition [74], just to mention two of the most popular. At the same time, these ML techniques, both the classical ones and those based on deep learning, are using GP to improve their results [22, 68]. Thus, another possible way to improve NLP applications is to use GP to improve the design of ML and deep learning systems specific to the considered NLP application.

7 Challenges

Many challenges remain in the NLP area. Among them are the applications that have been considered in this work, such as opinion mining, detection of spam and extraction of information in domains such as health care, legal, journalistic, etc. In addition, the deep understanding of the language also has several pending aspects. Among them are, for example, the detection of negation (negated facts have to be identified) or word sense disambiguation, fundamental in information extraction. Although many of them have already been dealt with, there is plenty of room for improvement.

Another important challenge in understanding the language is to advance in the integration of the world knowledge that is required to capture the semantics of texts. Currently there are repositories such as Wikipedia or Babelnet [57] that allow us to connect concepts identified in the texts with additional knowledge about them. These connections can help, for example, to improve question-answering systems. In all these applications, and in many others, GP can help exploring new ways of understanding and approaching the problem.

Probably there are two main requirements for the proliferation of works pursuing the application of GP to more NLP problems. One is to improve the performance and the other is to facilitate the design of these applications. They may explain why there are fewer works that one could expect, given the potential of these techniques.

The now so popular systems based on deep learning have two very important advantages. One of them is their great performance and the other is the existence of tools for user friendly design. Tools such as Keras [15] have emerged, allowing easy and fast prototyping at a very high level of programming. Many deep learning systems for NLP applications use simple features such as the words in the text, their characters, and their assigned POS tags. Thus the system design is simple using these high level tools and appealing to many researchers.

GP systems need a greater effort in the design of the systems, which involves the selection of quite a few elements to be considered, going from the individual representation to the fitness function and including the data and operators that make up the generated programs or rules. Accordingly, a challenge to consider is the development of very high level tools that facilitate the quick and easy development of applications of GP to NLP.

Another big challenge GP has to face for dealing with NLP applications is to look for mechanisms to improve its performance. Certainly, the computational capacity of machines keeps increasing, spreading the use of computationally expensive techniques. This is what happened with deep learning. However, at the same time, the problems we face are becoming more complex and larger amounts of data are required to be processed. This is the case of NLP problems, which deals with real data collected from scientific articles, medical records, or opinions gathered from the Internet. One option is exploring specific designs for NLP. For example, the evaluation of individuals in algorithms working with parse trees, or trees representing taxonomies, is expensive. Mechanisms to reuse the evaluation of parts of trees that have already been evaluated could be very helpful.

8 Conclusions

This paper has reviewed some works in which GP techniques have been applied to NLP problems, providing interesting ideas. These works suggest that GP and NLP are a combination of techniques that match very well. Some reasons have been identified in Sect. 6. As it is stated in that section, there are quite a few open problems in NLP that offer an opportunity to explore GP techniques.

An inherent feature of GP and GE algorithms is that they do not guarantee optimality of the solutions. Yet, this feature does not have to be a handicap for many NLP applications. Human language has a strong subjective component. For example, the usual practice when annotating a linguistic corpus is that several experts annotate the same texts, in order to be able to compare their results and try to reach an agreement on the annotation criteria. Indeed, there are many ways to express the same ideas. In NLP a task is usually carried out by trying to approximate a specific reference model—for example the model for parsing can be given by the parse trees in a corpus. However, the task can become quite different if we consider a different reference model or corpus. Therefore, the approximate character of the solutions of an evolutionary algorithm is not a big deal for most NLP applications.

It has also been observed that the amount of work in this line is less than it might be expected from this complementarity between GP and NLP. The challenges section points out two possible research lines that might palliate this situation.

Acknowledgements This work has been partially supported by the Spanish Ministry of Science and Innovation within the Projects PROSA-MED (TIN2016-77820-C3-2-R).

References

1. L. Araujo, Genetic programming for natural language parsing, in *Proceedings of the European Conference on Genetic Programming (EuroGP2004), Lecture Notes in Computer Science*, vol. 3003 (Springer, Berlin, 2004), pp. 230–239
2. L. Araujo, Symbiosis of evolutionary techniques and statistical natural language processing. IEEE Trans. Evol. Comput. **8**(1), 14–27 (2004)
3. L. Araujo, Multiobjective genetic programming for natural language parsing and tagging, in *PPSN* (2006), pp. 433–442
4. L. Araujo, How evolutionary algorithms are applied to statistical natural language processing. Artif. Intell. Rev. **28**(4), 275–303 (2007)
5. L. Araujo, J. Martinez-Romo, A.D. Fernandez, Discovering taxonomies in Wikipedia by means of grammatical evolution. Soft Comput. **22**(9), 2907–2919 (2018)
6. A. Bartoli, G. Davanzo, A. De Lorenzo, E. Medvet, E. Sorio, Automatic synthesis of regular expressions from examples. Computer **47**(12), 72–80 (2014)
7. A. Bartoli, A. De Lorenzo, E. Medvet, F. Tarlao, M. Virgolin, Evolutionary learning of syntax patterns for genic interaction extraction, in *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, GECCO '15* (ACM, New York, 2015), pp. 1183–1190
8. A. Bartoli, A.D. Lorenzo, E. Medvet, F. Tarlao, Syntactical similarity learning by means of grammatical evolution, in *PPSN, Lecture Notes in Computer Science*, vol. 9921 (Springer, Berlin, 2016), pp. 260–269
9. A. Bartoli, A.D. Lorenzo, E. Medvet, F. Tarlao, Active learning of regular expressions for entity extraction. IEEE Trans. Cybern. **48**(3), 1067–1080 (2018)
10. V. Basto-Fernandes, I. Yevseyeva, R.Z. Frantz, C. Grilo, N.P. Díaz, M. Emmerich, An automatic generation of textual pattern rules for digital content filters proposal, using grammatical evolution genetic programming. Proc. Technol. **16**, 806–812 (2014)
11. A. Bergström, P. Jaksetic, P. Nordin, Enhancing information retrieval by automatic acquisition of textual relations using genetic programming, in *Proceedings of the 5th International Conference on Intelligent User Interfaces, IUI '00* (ACM, New York, 2000), pp. 29–32
12. J. Bootkrajang, S. Kim, B. Zhang, Evolutionary hypernetwork classifiers for protein–protein interaction sentence filtering, in *Genetic and Evolutionary Computation Conference, GECCO 2009, Proceedings, Montreal, Québec, Canada, July 8–12, 2009*, ed. by F. Rothlauf (2009), pp. 185–192
13. M. Brameier, W. Banzhaf, A comparison of linear genetic programming and neural networks in medical data mining. IEEE Trans. Evol. Comput. **5**(1), 17–26 (2001)
14. W.W. Chapman, K.B. Cohen, Current issues in biomedical text mining and natural language processing. J. Biomed. Inf. **42**(5), 757–759 (2009)
15. P. Charles, Project title. https://github.com/charlespwd/project-title (2013)
16. H. Christiansen, A survey of adaptable grammars. SIGPLAN Not. **25**(11), 35–44 (1990)
17. A.M. Cohen, W.R. Hersh, A survey of current work in biomedical text mining. Brief. Bioinf. **6**(1), 57–71 (2005)
18. R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa, Natural language processing (almost) from scratch. J. Mach. Learn. Res. **12**, 2493–2537 (2011)
19. E. Conrad, *Detecting Spam With Genetic Regular Expressions*, Technical report (SANS Technology Institute, 2007)
20. O. Cordón, E. Herrera-Viedma, C. López-Pujalte, M. Luque, C. Zarco, A review on the application of evolutionary computation to information retrieval. Int. J. Approx. Reason. **34**(2–3), 241–264 (2003)

Journal : SmallExtended 10710	Article No : 9361	Pages : 22	MS Code : 9361	Dispatch : 20-7-2019

21. M.G. de Carvalho, A.H.F. Laender, M.A. Goncalves, A.S. da Silva, A genetic programming approach to record deduplication. IEEE Trans. Knowl. Data Eng. **24**(3), 399–412 (2012)
22. P.G. Espejo, S. Ventura, F. Herrera, A survey on the application of genetic programming to classification. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. **40**(2), 121–144 (2010)
23. H. Fabregat, L. Araujo, J. Martinez-Romo, Deep neural models for extracting entities and relationships in the new RDD corpus relating disabilities and rare diseases. Comput. Methods Programs Biomed. **164**, 121–129 (2018)
24. S. Faralli, A. Panchenko, C. Biemann, S.P. Ponzetto, Linked disambiguated distributional semantic networks, in *International Semantic Web Conference (2). Lecture Notes in Computer Science, vol. 9982* (2016), pp. 56–64
25. M. Faruqui, J. Dodge, S.K. Jauhar, C. Dyer, E. Hovy, N.A. Smith, Retrofitting word vectors to semantic lexicons, in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (Association for Computational Linguistics, 2015), pp. 1606–1615
26. F. Frasincar, J. Borsje, F. Hogenboom, E-Business applications for product development and competitive growth: emerging technologies, chap., in *Personalizing News Services Using Semantic Web Technologies* (IGI Global 2011), pp. 261–289
27. A. González-Pardo, D. Camacho, Analysis of grammatical evolutionary approaches to regular expression induction, in *IEEE Congress on Evolutionary Computation* (IEEE 2011), pp. 639–646
28. M. Graff, E.S. Tellez, H.J. Escalante, S. Miranda-Jiménez, Semantic genetic programming for sentiment analysis, in *NEO, Studies in Computational Intelligence*, vol. 663 (Springer, Berlin, 2015), pp. 43–65
29. M. Graff, E.S. Tellez, S. Miranda-Jiménez, H.J. Escalante, Evodag: a semantic genetic programming python library, in *2016 IEEE International Autumn Meeting on Power, Electronics and Computing* (ROPEC, 2016), pp. 1–6
30. R. Greenstadt, M. Kaminsky, *Evolving Spam Filters Using Genetic Algorithms*, Technical Report 3836. (Massachusetts Institute of Technology, 2002)
31. A. Holzinger, C. Biemann, C.S. Pattichis, D.B. Kell, What do we need to build explainable AI systems for the medical domain? CoRR arXiv:1712.09923 (2017)
32. A. Holzinger, J. Schantl, M. Schroettner, C. Seifert, K. Verspoor, *Biomedical Text Mining: State-of-the-Art, Open Problems and Future Challenges* (Springer, Berlin, 2014), pp. 271–300
33. W. IJntema, F. Hogenboom, F. Frasincar, D. Vandic, A genetic programming approach for learning semantic information extraction rules from news, in *Web Information Systems Engineering—WISE 2014—15th International Conference, Thessaloniki, Greece, October 12–14, 2014, Proceedings, Part I, Lecture Notes in Computer Science*, vol. 8786, ed. by B. Benatallah, A. Bestavros, Y. Manolopoulos, A. Vakali, Y. Zhang (Springer, Berlin, 2014), pp. 418–432
34. W. IJntema, J. Sangers, F. Hogenboom, F. Frasincar, A lexico-semantic pattern language for learning ontology instances from text. Web Semant. Sci. Serv. Agents World Wide Web **15**(3), 37–50 (2012)
35. R. Isele, C. Bizer, Active learning of expressive linkage rules using genetic programming. Web Semant. Sci. Serv. Agents World Wide Web **23**, 2–15 (2013)
36. D. Jurafsky, J.H. Martin, *Speech and Language Processing*, 2nd edn. (Prentice-Hall Inc, Upper Saddle River, 2009)
37. A. Khorsi, An overview of content-based spam filtering techniques. Informatica (Slovenia) **31**(3), 269–277 (2007)
38. K.M. Kim, S.S. Lim, S.B. Cho, User adaptive answers generation for conversational agent using genetic programming, in *Intelligent Data Engineering and Automated Learning—IDEAL 2004*, ed. by Z.R. Yang, H. Yin, R.M. Everson (Springer, Berlin, 2004), pp. 813–819
39. E.E. Korkmaz, G. Üçoluk, A controlled genetic programming approach for the deceptive domain. IEEE Trans. Syst. Man Cybern. Part B **34**(4), 1730–1742 (2004)
40. I. Korkontzelos, D. Piliouras, A.W. Dowsey, S. Ananiadou, Boosting drug named entity recognition using an aggregate classifier. Artif. Intell. Med. **65**(2), 145–153 (2015)
41. J.R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection* (MIT Press, Cambridge, 1992)
42. M. Lan, C.L. Tan, J. Su, Feature generation and representations for protein–protein interaction classification. J. Biomed. Inf. **42**(5), 866–872 (2009)
43. Y. LeCun, Y. Bengio, G. Hinton, Deep learning. Nature **521**, 436 (2015)

44. F. Li, M. Zhang, G. Fu, D. Ji, A neural joint model for entity and relation extraction from biomedical text. BMC Bioinf. **18**(1), 198:1–198:11 (2017)
45. S. Lim, S. Cho, Language generation for conversational agent by evolution of plan trees with genetic programming, in *MDAI, Lecture Notes in Computer Science*, vol. 3558 (Springer, Berlin, 2005), pp. 305–315
46. B. Liu, L. Zhang, *A Survey of Opinion Mining and Sentiment Analysis* (Springer, New York, 2013), pp. 415–463
47. C.D. Manning, P. Raghavan, H. Schütze, *Introduction to Information Retrieval* (Cambridge University Press, New York, 2008)
48. H. Manurung, *An Evolutionary Algorithm Approach to Poetry Generation*, Ph.D. thesis (University of Edinburgh, School of Informatics, 2003)
49. R. Manurung, G. Ritchie, H. Thompson, An implementation of a flexible author-reviewer model of generation using genetic algorithms, in *Proceedings of the 22nd Pacific Asia Conference on Language, Information and Computation (PACLIC)* (De La Salle University (DLSU), Manila, 2008), pp. 272–281
50. E. Martínez-Cámara, M.C. Díaz-Galiano, M. Ángel García-Cumbreras García-Vega, M. Villena-Román, J.: Overview of TASS 2017, in *TASS@SEPLN, CEUR Workshop Proceedings*. CEUR-WS.org (2017), pp. 13–21
51. K.R. McKeown, *Text Generation—Using Discourse Strategies and Focus Constraints to Generate Natural Language Text. Studies in Natural Language Processing* (Cambridge University Press, Cambridge, 1992)
52. G.A. Miller, Wordnet: a lexical database for english. Commun. ACM **38**(11), 39–41 (1995)
53. M. Miwa, M. Bansal, End-to-end relation extraction using LSTMs on sequences and tree structures, in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7–12, 2016, Berlin, Germany, Volume 1* (Long Papers, 2016), pp. 1105–1116
54. D. Moctezuma, M. Graff, S. Miranda-Jiménez, E.S. Tellez, A. Coronado, CN. Sánchez, J. Ortiz-Bejar, A genetic programming approach to sentiment analysis for twitter: Tass17, in *TASS 2017: Workshop on Semantic Analysis at SEPLN* (CEUR, 2017), pp. 23–28
55. A. Moraglio, K. Krawiec, C.G. Johnson, Geometric semantic genetic programming, in *PPSN (1), Lecture Notes in Computer Science*, vol. 7491 (Springer, Berlin, 2012), pp. 21–31
56. D. Nadeau, S. Sekine, A survey of named entity recognition and classification. Linguist. Invest. **30**(1), 3–26 (2007)
57. R. Navigli, S.P. Ponzetto, BabelNet: the automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artif. Intell. **193**, 217–250 (2012)
58. M. O'Neill, C. Ryan, Under the hood of grammatical evolution, in *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation—Volume 2, GECCO'99* (Morgan Kaufmann Publishers Inc., Los Altos, 1999), pp. 1143–1148
59. M. O'Neill, C. Ryan, Grammatical evolution. IEEE Trans. Evol. Comput. **5**(4), 349–358 (2001)
60. A. Ortega, M. de la Cruz, M. Alfonseca, Christiansen grammar evolution: grammatical evolution with semantics. IEEE Trans. Evol. Comput. **11**(1), 77–90 (2007)
61. B. Percha, R.B. Altman, Learning the structure of biomedical relationships from unstructured text. PLoS Comput. Biol. **11**(7), e1004216 (2015)
62. R. Perera, P. Nand, Recent advances in natural language generation: a survey and classification of the empirical literature. Comput. Inf. **36**(1), 1–32 (2017)
63. C.P. Rose, A genetic programming approach for robust language interpretation, in *Advances in Genetic Programming*, vol. 3, ed. by L. Spector, W.B. Langdon, U.M. O'Reilly, P.J. Angeline (MIT Press, Cambridge, 1999), pp. 67–88
64. D. Ruano-Ordás, F. Fdez-Riverola, J.R. Méndez, Using evolutionary computation for discovering spam patterns from e-mail samples. Inf. Process. Manag. **54**(2), 303–317 (2018)
65. C. Ryan, J. Collins, J. Collins, M. O'Neill, Grammatical evolution: evolving programs for an arbitrary language, in *Lecture Notes in Computer Science, Proceedings of the First European Workshop on Genetic Programming*, vol. 1391 (Springer, Berlin, 1998), pp. 83–95
66. A. Schwartz, *SpamAssassin* (O'Reilly Media Inc., Newton, 2004)
67. T.C. Smith, I.H. Witten, A genetic algorithm for the induction of natural language grammars, in *Proceedings of the IJCAI-95 Workshop on New Approaches to Learning for Natural Language Processing* (1995), pp. 17–24

68. M. Suganuma, S. Shirakawa, T. Nagao, A genetic programming approach to designing convolutional neural network architectures, in *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '17* (ACM, New York, 2017), pp. 497–504
69. H. Takagi, Interactive evolutionary computation: fusion of the capabilities of EC optimization and human evaluation. Proc. IEEE **89**(9), 1275–1296 (2001)
70. I. Tiddi, M. d'Aquin, E. Motta, Learning to assess linked data relationships using genetic programming, in *International Semantic Web Conference (1). Lecture Notes in Computer Science*, vol. 9981 (2016), pp. 581–597
71. J. Villena-Román, J. García-Morera, MÁG. Cumbreras, E. Martínez-Cámara, MT. Martín-Valdivia, LAU. López, Overview of TASS 2015, in *TASS@SEPLN, CEUR Workshop Proceedings*, vol. 1397, CEUR-WS.org (2015), pp. 13–21
72. S. Winkler, S. Schaller, V. Dorfer, M. Affenzeller, G. Petz, M. Karpowicz, Data-based prediction of sentiments using heterogeneous model ensembles. Soft Comput. **19**(12), 3401–3412 (2015)
73. H.Y. Wu, S. Karnik, A. Subhadarshini, Z. Wang, S. Philips, X. Han, C. Chiang, L. Liu, M. Boustani, L.M. Rocha, S.K. Quinney, D. Flockhart, L. Li, An integrated pharmacokinetics ontology and corpus for text mining. BMC Bioinf. **14**, 35 (2013)
74. V. Yadav, S. Bethard, A survey on recent advances in named entity recognition from deep learning models, in *Proceedings of the 27th International Conference on Computational Linguistics* (Association for Computational Linguistics, 2018), pp. 2145–2158
75. T. Young, D. Hazarika, S. Poria, E. Cambria, Recent trends in deep learning based natural language processing. IEEE Comput. Int. Mag. **13**(3), 55–75 (2018)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.