Soft Comput
DOI 10.1007/s00500-017-2544-4

Discovering taxonomies in Wikipedia by means of grammatical evolution

Lourdes Araujo[1] · **Juan Martinez-Romo**[1] · **Andrés Duque**[1]

Abstract This work applies grammatical evolution to identify taxonomic hierarchies of concepts from Wikipedia. Each [1] article in Wikipedia covers a topic and is cross-linked by hyperlinks that connect related topics. Hierarchical taxonomies and their generalization to ontologies are a highly useful resource for many applications since they enable semantic search and reasoning. Thus, the automatic identification of taxonomies composed of concepts associated with linked Wikipedia pages has attracted much attention. We have developed a system which arranges a set of Wikipedia concepts into a taxonomy. This technique is based on the relationships among a set of features extracted from the contents of the Wikipedia pages. We have used a grammatical evolution algorithm to discover the best way of combining the considered features in an explicit function. Candidate functions are evaluated by applying a genetic algorithm to approximate the optimal taxonomy that the function can provide for a number of training cases. The fitness is computed as an average of the precision obtained by comparing, for the set of training cases, the taxonomy provided by the evaluated function with the reference one. Experimental results show that the proposal is able to provide valuable functions to find high-quality taxonomies. [2]

Communicated by V. Loia.

✉ Lourdes Araujo
lurdes@lsi.uned.es

Juan Martinez-Romo
juaner@lsi.uned.es

Andrés Duque
aduque@lsi.uned.es

[1] Universidad Nacional de Educación a Distancia (UNED), Madrid, Spain

Keywords Grammatical evolution · Genetic algorithm · Wikipedia taxonomies · Information extraction

1 Introduction

A key step toward the full Semantic Web functionality is the efficient organization of human knowledge in ontologies. These usually large and handmade structures have to be adapted to new knowledge in an efficient and reliable way.

There are a wide range of ontology and taxonomy applications. They include summarization (Morales et al. 2008), terminology translation (Navigli et al. 2003), detection of relevant features from textual resources, useful in classification and clustering applications (Vicient et al. 2013), classification of the relevance of the answers for a query (Galitsky 2013), machine translation (Hovy 1998), automatic query expansion (Bhogal et al. 2007), document classification (Camous et al. 2007), word sense disambiguation (Prokofyev et al. 2013), to name a few.

In this work, we propose a method for automatically organizing parts of a wide spread and constantly updated source of knowledge, which is Wikipedia. Nowadays, Wikipedia is the most popular and largest reference work. This freely available encyclopedia is collaboratively edited on the Internet. Information in Wikipedia is organized in articles, and each of them devoted to a particular topic. Wikipedia articles are cross-linked by hyperlinks inserted in the text. An interesting question that arises when considering linked Wikipedia pages is the kind of relationship between the linked concepts. In particular, we are interested in identifying the "is a" relationship between Wikipedia concepts in order to organize them into a taxonomy or hierarchy. This kind of relationship does not always explicitly appear in the content of the articles. For example, the Wikipedia page for *animal* has a link to the Wikipedia page entitled *mammal*. However, the page *mammal* does not explicitly say that a mammal is an animal: *Mammals are a clade of endothermic amniotes distinguished from the reptiles and the birds by the possession of hair, three middle ear bones, mammary glands in females, and a neocortex (a region of the brain)*... Thus, we need to resort to other methods to identify this kind of relationship.

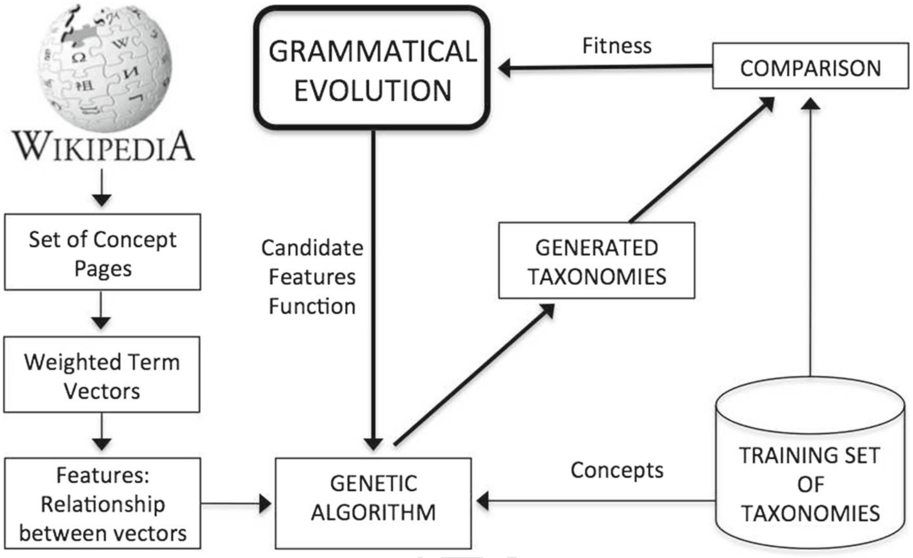
Medelyan et al. (2009) made an in-depth review of the different uses that the research community has given Wikipedia, such as information extraction and ontology building. Actually, there are several efforts to construct ontologies from Wikipedia pages. Several works focus on deriving relations from article text. Ruiz-Casado et al. (2005) used WordNet for mining the patterns that capture the semantic relation between Wikipedia entities. Given two co-occurring semantically related WordNet nouns, the text that appears between them in Wikipedia articles is used to find relations missing from WordNet. Other works (Herbelot and Copestake 2006; Sucha nek et al. 2006; Nguyen et al. 2007) use a different kind of parsers to identify the concepts and the relationships between them.

There are also works investigating the relationships among Wikipedia categories. Chernov et al. (2006) studied whether links between Wikipedia categories bear semantic meaning. They find that the hyperlink connectivity between articles in two categories correlates with the semantic relatedness of those categories. Nakayama et al. (2007) also exploited this idea and built a large association thesaurus, without specifying the kind of relationship. YAGO, Yet Another Great Ontology (Suchanek et al. 2007), is a large taxonomy created by mapping Wikipedia's leaf categories onto the WordNet taxonomy of synsets and adding the articles belonging to those categories as new elements. Khalatbari and Mirroshandel (2015) proposed the construction of a prototype ontology in the animal domain using the Infoboxes in Wikipedia pages to extract facts. As this information is often incomplete, they use Google searches to look for the missed facts. Ben Aouicha et al. (2016) proposed a method for obtaining an "is a" taxonomy from the Wikipedia Categories Graph (WCG). This graph is constructed by volunteers who link Wikipedia categories without explicitly specifying the kind of the relation. They exploit expression patterns, such as BY (as in *Songs by songwriter*), to identify the kind of relationship. For example, the relation between a category whose name contains BY and its descendants is qualified as "is a". Another example of ontology related to Wikipedia is the DBpedia ontology (Lehmann et al. 2014). DBpedia is a project aiming to extract structured information from Wikipedia and to make this information available on the emerging Web of Data. The DBpedia project maps Wikipedia infoboxes from different language editions to a single shared ontology. The DBpedia ontology is a shallow, cross-domain ontology, which has been manually created based on the most commonly used infoboxes within Wikipedia. The ontology currently covers 529 classes which form a subsumption hierarchy. Wikipedia has also been used for expanding existing ontologies. Schlegel et al. (2015) resorted to Wikipedia as a source of synonyms to expand SNOMED CT, an ontology of clinical terminology commonly used for processing clinical documents. The authors propose methods for aligning concepts in SNOMED CT with Wikipedia articles in order to find synonyms that may be added to SNOMED CT. Ali and Raghavan (2015) used Wikipedia to extend the Simple Knowledge Organization System (SKOS) (Miles and Bechhofer 2008). It is a W3C recommendation for representing taxonomies, as well as any structured controlled vocabulary. The authors propose the annotation model SKOS-Wiki, using the structure of the Wikipedia network and the template within the Wikipedia pages to define different types of concepts.

These and other works (Wu and Weld 2007; Weber and Buitelaar 2006; Ponzetto and Strube 2007) indicate the actual need of discovering and organizing the relationships within the encyclopedic knowledge of Wikipedia.

Given the complexity of the problem, metaheuristic approaches, such as evolutionary algorithms, are among the methodologies used to deal with the generation of taxonomies. We have to take into account that the number of possible trees with a fixed set of N nodes is N^{N-2} (Cayley's tree formula) (Clarke 1958). Even for a small number of nodes, the amount of possible trees is huge, and thus, heuristic methods are required. Some works applying metaheuristic approaches have been devoted to the hierarchical multi-label classification (HMC) problem of assigning functions to proteins, being each function represented by a class (term) in the gene ontology (GO). Cerri et al. (2014) applied a genetic algorithm, while Otero et al. (2009) proposed an ant colony optimization algorithm. Their methods discover classification rules which are able to predict GO terms. Othman et al. (2007) combined semantic similarity measures and a genetic algorithm to search semantically similar terms in the gene ontology. The genetic algorithm is employed to perform batch retrievals while handling the large search space of the gene ontology graph. Mao (2001) proposed to use formal semantics of ontology to improve genetic algorithms and make them more adaptive for semantic-based problems. He illustrated the usage of the algorithm with a traditional Chinese medicine ontology. Isele and Bizer (2013) presented the ActiveGenLink tool which combines genetic programming and active learning to generate expressive linkage rules interactively. The ActiveGenLink algorithm automates the generation of linkage rules, and then, the user can either confirm or decline a number of link candidates. Most of these approaches are focused on a gene ontology with a controlled vocabulary.

Fig. 1 System scheme. The input data to the algorithm are the Wikipedia pages associated with different concepts, and the training set of taxonomies. The grammatical evolution algorithm uses a genetic algorithm to compute the fitness of the candidate functions



There are also some recent works applying evolutionary approaches to deal with problems related to the one we are considering. Bartoli et al. (2016) proposed an algorithm based on grammatical evolution for learning a similarity function suitable for extracting syntactic patterns from unstructured text streams. Forsati and Shamsfard (2016) addressed the ontology mapping problem of identifying semantically aligned entities in different ontologies. They build a similarity matrix from different similarity measures. This matrix is used as fitness function in a search process based on a harmony search (HS) algorithm (Geem et al. 2001). HS algorithms are an optimization method which imitates the music improvisation process.

In this work, we have developed a system which arranges a set of Wikipedia concepts into a taxonomy. Wikipedia's articles are devoted to a particular topic, and related articles are connected by hyperlinks. Our proposal is based on the relationships among a set of features extracted from the contents of the Wikipedia pages. We apply grammatical evolution (GE), a kind of evolutionary algorithm, to discover the best way of combining the considered features in an explicit function. Candidate functions are evaluated by applying a genetic algorithm to approximate the optimal taxonomy that the function can provide for a number of training cases.

The remainder of the paper presents the model, its implementation and its evaluation. Section 2 shows a general overview of the system, whose elements are detailed in the following sections. Section 3 describes the features that are extracted from the Wikipedia pages content to define an evaluation function for the taxonomy in which the corresponding concepts should be arranged. Section 4 is devoted to the grammatical evolution algorithm which optimizes the candidate functions of features. The genetic algorithm used to compute the fitness of the GE algorithm is described in Sect. 5. Section 6 presents the experimental framework and results obtained. Finally, conclusions are drawn in Sect. 7.

2 System overview

Our system searches for a function capable of selecting a particular arrangement of a set of Wikipedia concepts in a taxonomy. The chosen arrangement should optimize a number of relationships among the concepts.

The GE algorithm works with a population of candidate functions which compete to be selected in the next generation according to its fitness. The candidate functions being evaluated should approximate the hierarchical relationships between the concepts of the considered taxonomy. Fitness is computed as the average, for the set of training taxonomies, of the precision obtained when comparing the taxonomy that presents the highest score according to the function, with the reference one. In order to obtain the highest score taxonomy that a candidate function can provide, we need to perform an optimization process which is, in turn, implemented by a genetic algorithm. We have used different parts of the DBpedia ontology for training and evaluation. Specifically, we have used a set of taxonomies extracted from the *Species* part of the DBpedia ontology for training.

Figure 1 shows a scheme of the system. Wikipedia provides the linked pages of articles related to a set of concepts. From the terms contained in each of these documents, we compute a weighted term vector associated with the corresponding concept. Different relationships can be expected to be fulfilled between the vectors associated with related concepts. Then, a function that appropriately combines these features can detect the hierarchical relation between two concepts. The grammatical evolution algorithm evolves functions combining the considered features. Fitness of a candidate function is computed by comparing the approximate best taxonomy that the function can obtain for a number of training cases, with the taxonomy of reference of each case. The best taxonomy for a function and training case is obtained applying a genetic algorithm, which uses the term vectors representing the documents to compute the value of the features appearing in the function. A very preliminary account of part of this work was presented in an abstract elsewhere (Araujo et al. 2015).

The contributions of this paper are threefold:

1. We explore new measures which capture not only similarity between concepts, but also the trend of a concept to be the parent or the child in a particular relationship, thus helping to determine the direction of the relationship between two given concepts.
2. We propose a novel approach based on grammatical evolution to arrange a set of concepts in the most appropriate taxonomy taking into account the relationships among pairs of concepts. Additionally, this approach produces explicit functions of features ready to be applied to new sets of concepts. Moreover, these functions provide insights into the relevance of different relations considered among concepts.
3. We introduce innovations in the evaluation of candidate functions in the grammatical evolutionary approach, using a GA to find a good approximation to the reference taxonomy required for evaluation. We also propose some optimizations to reduce the GE execution time that can be used for other applications of GE.

3 The Wikipedia taxonomies problem

As we can not trust that the article of a concept, such as *mammal*, contains an explicit expression indicating that *mammal* **is an** *animal*, we resort to statistical techniques to represent the pages and analyze their relationships.

In the vector space model (Salton et al. 1975), text documents are represented as vectors of terms. This representation is used in information extraction, information retrieval, indexing and relevance rankings. Each position in the vectors associated with documents corresponds to a term i in the set of documents:

$$d_j = (w_{1,j}, w_{2,j}, \ldots, w_{t,j})$$

The value of each term ($w_{i,j}$) indicates the relevance of the term as representative of the document d_j. If a term does not occur in the document, its value in the vector is zero. There are different ways of computing the value corresponding to each term, i.e., its weight. We represent each Wikipedia article by a vector of weights, each corresponding to a term in the collection of articles of the considered Wikipedia pages. We use one of the most common measures for weighting each term: TF-IDF (term frequency–inverse document frequency), where TF, $\mathrm{tf}(t, d)$, stands for the frequency of a term t in a document d, and IDF, $\mathrm{idf}(t, D)$, for the inverse document frequency of a term t in the considered collection D. In the case of the term frequency $\mathrm{tf}(t, d)$, we use the augmented frequency (Manning et al. 2008) to prevent a bias toward longer documents, i.e., raw frequency $f(t, d)$ (the number of times that term t occurs in document d) divided by the maximum raw frequency of a term in the document:

$$\mathrm{tf}(t, d) = 0.5 + \frac{0.5 \times f(t, d)}{\max\{f(w, d) : w \in d\}}$$

The inverse document frequency is a measure of whether the term is common or rare across all documents. It is obtained by dividing the total number of documents by the number of documents containing the term and then taking the logarithm of that quotient:

$$\mathrm{idf}(t, D) = \log \frac{|D|}{|d \in D : t \in d|}$$

where $|D|$ is the number of documents in the corpus or collection, and d is the number of documents where the term t appears. We have used an English Wikipedia articles dump[1] as reference collection. Then, tf-idf is computed as:

$$\text{tf-idf}(t, d, D) = \mathrm{tf}(t, d) \times \mathrm{idf}(t, D)$$

3.1 Relationships

The next point to tackle is to identify some relationships which tend to be met between two linked pages (i.e., their corresponding vectors) with a hierarchical relationship. We have considered the following features:

- COS (cosine) The most popular similarity measure is the cosine coefficient, which measures the angle between two document vectors. It is commonly used to detect whether two documents are really related to each other.
- *DIFSIM (differences in similarity)* This measure gives an approximation to the similarity between the intersection S of two vectors A and B, and any of the vectors, A or B. If there exists a hierarchical relationship between two concepts, one can expect that a large part of the content of one of them is included in the other one. The common part of the two concepts can be computed as:

$$s_i = a_i + b_i - (a_i \times b_i)$$

---

[1] http://download.wikimedia.org/enwiki/.

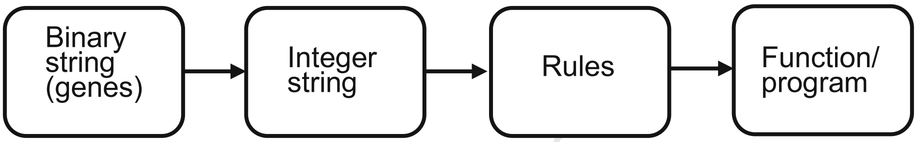


Fig. 2 Scheme of the mapping process in grammatical evolution

where a_i represents the components of the vector corresponding to concept A, and b_i those of concept B. The common part can be similar to any of the concepts. Then, we expect to have $\text{cosine}(\mathbf{S}, \mathbf{A}) \approx 1$ if the intersection of the two concepts is similar to A and $\text{cosine}(\mathbf{S}, \mathbf{B}) \approx 1$ if the intersection is similar to B. Then, DIFSIM feature measures the difference between both cosines. High values of DIFSIM may indicate a hierarchical relationship, distinguishing it from a sibling relationship.

- *Distinct terms* The DIFSIM feature is useful to detect whether there is a hierarchical relationship between two concepts, but not its direction. Accordingly, we have explored the terms non-shared (distinct) by the two concepts. One can expect that the relative degree of generality of two linked concepts affects the relevance of the particular terms of the concept. Specifically, we have considered the following measures related to the distinct terms:
  - *Average weight of distinct terms (AWD)* It is calculated as the average weight of terms appearing in one concept but not in the other one. This feature measures the relevance of the exclusive terms of a concept.
  - *Standard deviation of distinct terms (SDD)* It is calculated as the standard deviation of the weight of distinct terms in each concept. This feature measures the dispersion from the average of the exclusive terms of a concept.

  These features tend to adopt higher values for the parent–child relationship than for the child–parent one.

We now need to combine these features in a function able to detect the tendency of Wikipedia linked concepts to present a hierarchical relationship.

4 The proposal

Grammatical evolution (GE) (O'Neill and Ryan 2001) is an evolutionary algorithm that evolves programs using a Backus Naur Form (BNF) grammar to describe the output language and presents potential capacity for parallelization (He et al. 2016). In this way, GE does not perform the evolutionary process on actual programs, but on variable-length binary strings. A mapping process generates programs in any formal language by using the binary strings to produce integer strings, which are used to select production rules in a BNF grammar definition. The result is the construction of a syntactically correct program that can be evaluated by a fitness function. More precisely, variable-length binary string genomes are used with each codon or group of 8 bits representing an integer value. The integer values are then used in a mapping function to select an appropriate production rule from the BNF definition, the numbers generated always representing one of the rules. GE does not suffer from the problem of having to ignore codon integer values because it does not generate illegal values. Figure 2 outlines the mapping process.

As the population is composed of binary strings, we do not need any special crossover or mutation operators. The algorithm adopted in this case is a variable-length genetic algorithm. Individual initialization is achieved by randomly generating variable-length binary strings within a pre-specified range of codons. In the experiments conducted in this paper, we use the initialization range of ten codons, where a codon is a group of 8 bits. We adopt the standard genetic operators of one point mutation and one point crossover, as it is done by O'Neill and Ryan (2001).

The BNF grammar (Fig. 3) has been designed so as to include the features that have been identified as indicators of possible hierarchical relationships, such as cosine similarity (COS), the difference in the similarity of each concept and the intersection of both (DIFSIM), and the relevance of the distinct terms (AWD) and their deviation (SDD).

In this work, we adopt the standard approach to constant creation in genetic programming (GP), having values chosen randomly within a pre-specified range (Koza 1992). More sophisticated methods (Dempsey et al. 2007) have been proposed for the constant creation in GE. However, the values of the constants of our problem are limited to a small range—the range in which the features take values—and we have observed in the experiments that results are not too sensitive to small changes. Therefore, in this case the standard approach is valid, though it can be improved in the future.

One of the GE parameters is the allowed maximum depth for the trees representing the candidate functions. During the evaluation process, individuals that exceed the maximum depth are discarded. After each generation, the population is completed with new individuals to restore the required size.

The fitness of the GE algorithm is computed as the average precision achieved by comparing, for a number of training cases, the taxonomy provided by the candidate function and the reference taxonomy. In order to obtain the taxonomy which optimizes the value of the candidate function for a

```
expr
<expr> ::= <op> <var> <var>
         | if <cond> <expr> <expr>
         | <var>
<op> ::= +
         | -
         | /
         | *
<cond> ::= <var> = <var>
         | <var> < <var>
         | <var> > <var>
         | <var> >= <var>
         | <var> <= <var>
         | <var> = <cst>
         | <var> < <cst>
         | <var> > <cst>
         | <var> >= <cst>
         | <var> <= <cst>
<var> ::= COS
         | DIFSIM
         | AWD1
         | AWD2
         | SDD1
         | SDD2
<cst> ::= 0.05 | 0.1 | ... | 0.9 | 0 | 1 | ... | 9
```

Fig. 3 BNF grammar for the algorithm

particular set of concepts, we have resorted to a genetic algorithm.

5 Genetic algorithm for computing the fitness of the grammatical evolution algorithm

The input of this algorithm is a set of concepts that have to be arranged in a taxonomy. More specifically, the input is the features computed from the weighted term vectors of each pair of concepts in the input set. Individuals in this GA are taxonomies represented as vectors in which we can easily identify the descendant nodes of a given node and perform swapping between nodes.

Each position in the vector representing a taxonomy is devoted to a concept, and registers:

- The position of the parent node
- The number of children
- A vector with the positions of the children
- The level within the tree

5.1 Crossover operator

The crossover operator combines two different hierarchical arrangements of the same set of nodes. This is done by choosing at random a node different from the root and then swapping the subtrees under the nodes corresponding to the selected concept at each parent. However, we have to take into account that some of the nodes coming from the other parent *B* may be already present in the current parent *A*. In this case, the repeated nodes are erased from the coming subtree. Analogously, the coming subtree may lack some nodes which were present in the substituted subtree. Then, these nodes are included in the coming subtree as other children.

Due to the nature of the problem, in which all the individuals have to contain all the involved concepts, the crossover operator is somehow similar to a mutation operator.

5.2 Mutation operator

We have implemented four different mutation operators, which are randomly chosen when mutation is applied.

- Swap of two nodes, without their subtrees. The nodes are chosen at random.
- Swap of the root node with another node chosen at random. Changes in the root node have more influence on the results, and thus, we have introduced this specific operator for the root in order to favor the exploration of alternatives to the root.
- Search of the best swap for a node chosen at random.
- Swap of subtrees under two nodes chosen at random. This operator in somehow similar to the proposed crossover operator, but in this case there is no exchange of information between individuals.

5.3 Fitness function

The fitness function is computed as the sum, for all the nodes in the taxonomy, of the score assigned to the relationship between the node and its parent.

$$\sum_{node \in tax.} score(rel(node, parent))$$

The score of the relationship between the node and its parent is computed by applying the feature function being evaluated in the GE algorithm to these nodes.

The computation of this function is efficient since all the relationships between concepts are calculated and registered in advance in the initialization of the GE algorithm.

6 Experimental framework

We have focused the training for obtaining a set of functions able to arrange a set of concepts in a taxonomy, on the part of the DBpedia ontology concerning *species* that appears in

- Species
    - Archaea
    - Bacteria
    - Eukaryote
        - Animal
            - Amphibian
            - Arachnid
            - Bird
            - Crustacean
            - Fish
            - Insect
            - Mammal
            - Mollusca
            - Reptile
        - Fungus
        - Plant
            - Lycopodiopsida
            - Cycad
            - Fern
            - FloweringPlant
            - Ginkgo
            - Gnetophytes
            - GreenAlgae
            - Moss

Fig. 4 DBpedia ontology: Species part

Fig. 4. Later on, we have used other taxonomies, also from the DBpedia ontology, for evaluating the obtained functions. The concepts covered in the *species* taxonomy are very specific, and thus, they can provide less noisy results. We have downloaded the Wikipedia pages corresponding to the concepts in this hierarchy.[2]

We have built a training set composed of five taxonomies extracted from the *species* taxonomy shown in Fig. 4: the taxonomy corresponding to the concept *animal*, the one corresponding to the concept *plant*, other two taxonomies (*partial animal* and *partial plant*) which are subsets of the *animal* and *plant* taxonomies, respectively, and another one (*partial species*) which includes concepts from the whole *species* taxonomy, all of them composed of less than 14 concepts. These taxonomies are shown in Fig. 5. The reason for training with subsets of the *species* taxonomy instead of using the whole taxonomy is that the problem is too difficult for large taxonomies, as we have noticed in preliminary experiments, and has to be tackled considering relatively small sets of nodes. This difficulty is also shown by a baseline that corresponds to the precision—rate of relationships correctly detected—achieved by randomly generating each of the training taxonomies. Table 1 presents this baseline as the average and standard deviation of the precision achieved in 20 random generations of each taxonomy in the training set. The low values, below 1%, obtained for the larger taxonomies, *animal*, *plant* and *partial species*, indicate the difficulty of the problem.

Let us assume to illustrate the evaluation process that the following candidate function has been generated by the GE algorithm:

$$\begin{aligned} &expr(if(cond(var(AWD2), <=, var(SDD1)),\\ &\quad expr(op(+), var(DIFSIM), var(COS)),\\ &\quad expr(op(/), var(DIFSIM), var(SDD1)))) \end{aligned} \tag{1}$$

In order to compute its fitness, the function is applied to the subset of nodes of the training cases. Let us consider a training set composed of only two training cases corresponding to the *animal* and *plant* taxonomies appearing in Fig. 5. Let us assume that the GA used for computing the fitness produces the taxonomies shown in Fig. 6 as an approximation to the taxonomies with the highest score that the candidate function can provide for these two training sets. Then, the fitness of the function is computed as the average of the precision for the two cases. In the *animal* taxonomy, the function has been able to capture 8 out of 10 relations between concepts of the taxonomy, achieving 80% of precision. In the *plant* taxonomy, the function has been able to capture 5 out of 9 relationships, achieving 55.5% of precision. Thus, the fitness would be 67.75, the average of both values.

6.1 Obtaining the functions that evaluate Wikipedia taxonomies

We have to take into account that this GA is run each time that an individual of the GE algorithm has to be evaluated. Therefore, we have to look for a set of parameters which provide good enough individuals in a short time.

After a number of tests, we have selected the values appearing in Table 2. The algorithm is run until convergence or reaching the maximum number of generations. We can observe that the mutation rate is higher than the crossover rate. Due to the nature of the problem, in which all the trees of the population are composed of the same set of concepts, mutation and crossover are quite similar. Besides, we have introduced a variety of mutation operators which aim to provide different kinds of information exchange. Thus, we favor the application of this operator.

Table 3 shows the parameters adopted for the GE algorithm. The last parameter corresponds to the maximum depth allowed in the trees representing the candidate functions. We have observed that a maximum depth of 40 is enough for the algorithm to generate the most useful functions.

In order to reduce execution time, we have introduced some optimizations. First, all the data the GA fitness computation requires are calculated and registered in advance for each pair of concepts involved in the test. In addition, we record the sequences of grammar rules that have already appeared during the execution and their fitness. Specifically,

---

[2] They are available at http://nlp.uned.es/~lurdes/wikipedia_data.

- Animal
  - Amphibian
  - Arachnid
  - Bird
  - Crustacean
  - Fish
  - Insect
  - Mammal
  - Mollusca
  - Reptile

Animal Taxonomy

- Plant
  - Lycopodiopsida
  - Pinophyta
  - Cultivar
  - Cycad
  - Fern
  - FloweringPlant
  - Ginkgo
  - Gnetophyta
  - GreenAlgae
  - Moss

Plant Taxonomy

- Animal
  - Amphibian
  - Arachnid
  - Crustacean
  - Fish
  - Insect
  - Mollusca
  - Reptile

Partial Animal Tax.

- Plant
  - Lycopodiopsida
  - Cultivar
  - Cycad
  - Fern
  - FloweringPlant
  - Gnetophyta
  - Moss

Partial Plant Tax.

- Species
  - Eukaryote
    - Plant
      - Cultivar
      - Cycad
      - FloweringPlant
      - Moss
    - Animal
      - Arachnid
      - Bird
      - Mammal
      - Mollusca
      - Reptile

Partial Species Tax.

Fig. 5 Taxonomies in the training set

Table 1 Baseline (precision) for 20 random generations for each taxonomy in the training set

Taxonomy	Average	SD
Animal	0.09	0.10
Plant	0.075	0.09
Partial animal	0.22	0.14
Partial plant	0.12	0.15
Partial species	0.07	0.07

- Animal
  - Amphibian
  - Arachnid
  - Bird
  - Fish
  - Insect
  - Mammal
  - Mollusca
    - Crustacean
  - Reptile

(a)

- Moss
  - Plant
    - Lycopodiopsida
    - Cycad
    - Fern
    - FloweringPlant
    - Ginkgo
  - Gnetophytes
  - GreenAlgae

(b)

Fig. 6 Examples of taxonomies that could be generated with the GA for the candidate function of Eq. 1 for *animal* (**a**) and *plant* (**b**)

Table 2 Parameters of the GA used in the GE fitness evaluation

Parameter	Value
Population size	20
N. generations	50
Crossover rate	10%
Mutation rate	50%

Table 3 Parameters of the GE algorithm

Parameter	Value
Population size	20
N. generations	50
Crossover rate	40%
Mutation rate	10%
Max. depth of the tree	40

we register the different sequences of integers used to select the BNF grammar rules (after applying the module function to the integers corresponding to the binary strings of the GE algorithm) associated with each candidate function, along with the best fitness obtained in three evaluations. Then, when an individual has to be evaluated, we check whether its genotype corresponds to a phenotype that has already been evaluated three times. In this case, the individual does not need to be evaluated again.

6.1.1 Best functions found

Table 4 shows some of the best functions found by the algorithm along different runs. Table 5 shows the results achieved with these functions for the training set. We can observe that all of them provide average results around 80%, which is a high value if we consider the difficulty of the problem, as Table 1 indicates. However, none of the functions considered has been able to provide the reference taxonomy for all the training taxonomies, being the best precision of all of them around 85%.

In order to analyze these results, Table 6 shows the precision achieved for each taxonomy in the training set. This table shows the best and average fitness of 20 runs for each of the first five functions shown in Table 4 for the training set. Though all the functions appearing in Table 4 provide high precision for the training taxonomies, we have selected a subset of five of them, whose differences in the results are statistically significant, as we will see later. We can observe that all the selected functions achieve high results for the training taxonomies. In fact, the first three functions are able to find the four first reference taxonomies in some of the runs. However, none of them has been able to produce the reference taxonomy for the *partial species* case. This one is not only the larger one, but it also includes the most general concepts, as *species*, which makes the problem more difficult.

Table 4 Best functions found

ID	Function
F1	expr(if(cond(var(AWD2),<=,var(SDD1)), expr(op(+), var(DIFSIM), var(COS)), expr(op(/),var(DIFSIM),var(SDD1))))
F2	expr(if(cond(var(SDD2),>,cte(0.6)), expr(op(/), var(AWD1), var(AWD2)), expr(if(cond(var(SDD1),>=,var(SDD2)), expr(var(COS)), expr(op(/),var(AWD1),var(AWD2))))))
F3	expr(op(/),var(AWD1),var(SDD2))
F4	expr(if(cond(var(SDD2),>=,var(SDD1)), expr(op(*), var(COS), var(COS)), expr(var(DIFSIM))))
F5	expr(if(cond(var(SDD2),<=,var(AWD1)), expr(op(*), var(AWD1), var(DIFSIM)), expr(op(/),var(SDD1),var(AWD1))))
F6	expr(op(/),var(SDD1),var(SDD2))
F7	expr(if(cond(var(AWD1),>=,var(SDD2)), expr(var(DIFSIM)), expr(var(COS))))
F8	expr(op(/),var(SDD1),var(AWD2))
F9	expr(if(cond(var(AWD1),<=,var(DIFSIM)), expr(var(SDD1)), expr(op(/),var(SDD1),var(SDD2))))
F10	expr(if(cond(var(SDD1),>=,var(AWD2)), expr(op(+), var(AWD1), var(DIFSIM)), expr(op(/),var(DIFSIM),var(DIFSIM))))

Table 5 Precision achieved for each function in Table 4 for the training set

Func.	Best	Average	SD
F1	0.85	0.82	0.02
F2	0.84	0.81	0.02
F3	0.86	0.80	0.03
F4	0.86	0.77	0.04
F5	0.84	0.79	0.03
F6	0.86	0.81	0.02
F7	0.86	0.75	0.04
F8	0.87	0.80	0.04
F9	0.86	0.82	0.02
F10	0.83	0.74	0.06

The first column shows the best result, the second column the average, and the last one the standard deviation of 20 executions

6.2 Results for the test set

Once we have obtained a set of functions, we have tested them on a different set of taxonomies also extracted from DBpedia ontology, which is our reference for evaluation. The test set of taxonomies appears in Fig. 7. They correspond to concepts related to *time periods*, *musical works*, *means of transport* and *person*. Table 7 shows a baseline for the results. These values are the average and standard deviation achieved in 20 random generations of each taxonomy in the test set. In all cases, we can see very low values, below 0.2. The values are particularly low for the *person* taxonomy. These data indicate the difficulty of the problem for the test set.

Table 8 shows the results obtained by the five selected functions for the test set. We can observe that the results of each function depend on the test taxonomy, since the Wikipedia pages for each of them present different features. There has been at least one function able to produce the reference taxonomy for each the four test taxonomies. However, none of the functions has been able to produce the reference taxonomy for all the test taxonomies. Functions F2 and F3 have the best behavior in average as their results are above 50% for all the test taxonomies, and above 75% for three of them.

Table 9 shows the Wilcoxon test results for the considered functions in the case of the *person* taxonomy. We can see that the differences are statistically significant ($p < 0.05$).

Table 6 Precision achieved for the five selected functions in Table 4 for each taxonomy in the training set

Tax.	F1		F2		F3		F4		F5	
Animal	1	0.89(0.04)	1	0.92(0.06)	1	0.88(0.08)	1	0.77(0.11)	0.88	0.83(0.05)
Plant	1	0.90(0.03)	1	0.86(0.10)	1	0.88(0.07)	0.9	0.86(0.06)	0.9	0.87(0.04)
P. An.	1	0.85(0.04)	1	0.89(0.07)	1	0.87(0.07)	1	0.83(0.08)	1	0.82(0.07)
P. Pl.	1	0.90(0.06)	1	1(0)	1	0.97(0.06)	1	0.90(0.06)	1	0.87(0.04)
P. Sp.	0.75	0.55(0.09)	0.5	0.37(0.05)	0.5	0.39(0.06)	0.66	0.48(0.11)	0.66	0.56(0.09)

The first column shows the best result, and the second column the average and the standard deviation between parentheses, of 20 executions
P. An. partial animal, *P. Pl.* partial plant, *P. Sp.* partial species

Time period Tax.

- TimePeriod (Time)
  - GeologicalPeriod
  - HistoricalPeriod (History by Period)
  - PeriodOfArtisticStyle (ArtPeriod)
  - PrehistoricalPeriod (Prehistory)
  - Year
  - YearInSpaceflight (LightYear)

Musical Work Tax.

- MusicalWork (Musical composition)
  - Album
  - ArtistDiscography (Discography)
  - ClassicalMusicComposition (Classical Music)
  - NationalAnthem
  - Opera
  - Single
  - Song
    - Eurovision Song Contest

Person Tax.

- Person
  - Ambassador
  - Artist
  - Astronaut
  - Celebrity
  - Farmer
  - HorseTrainer
  - Journalist
  - Judge
  - Militarypersonnel
  - Model
  - Philosopher
  - Playboy
  - Playmate
  - Presenter
  - Psychologist
  - Referee

Means of Transportation Tax.

- MeanOfTransportation (Mode of transport)
  - Aircraft
  - MilitaryVehicle (Armoured fighting vehicle)
  - Automobile (Car)
  - Motorcycle
  - Ship
  - Spacecraft
  - Train

Fig. 7 Test set of taxonomies from DBpedia ontology

Table 7 Baseline for 20 random generations for each taxonomy in the test set

Taxonomy	Average	SD
Person	0.07	0.10
Time period	0.175	0.22
Musical work	0.1	0.11
Transport	0.19	0.19

Results indicate that the features extracted from the content of the Wikipedia pages are valuable indicators of the hierarchical relationships between linked pages. They also indicate that it is better to search for each part of the taxonomy separately, i.e., considering groups of concepts corresponding to Wikipedia pages directly connected. Taxonomies with several levels are too noisy for the functions to find appropriate arrangements.

As the taxonomy to be found becomes larger, the difficulty of the problem increases a lot. However, the evaluation functions found by the grammatical evolutionary algorithm during the training phase are valid for dealing with larger taxonomies. To show this fact, we have chosen the *person* taxonomy, one of the largest included in the Dbpedia ontology. This taxonomy has an only level, being all the nodes offspring of the "person" node, which is the case properly captured by the system. In order to analyze the scalability of the functions provided by the GE algorithm, we have run one of the best function found, $F3$ in Table 4, on a larger version of the person taxonomy, composed of up to 25 nodes, shown

Table 8 Best, average precision and deviation of each function in Table 4 for each considered test set

F.	Person			Time period			Musical work			Transportation		
	B.	Av.	St.	B.	Av.	St.	B.	Av.	St.	B.	Av.	St.
F1	0.75	0.58	0.14	1	0.61	0.15	1	0.79	0.20	0.42	0.13	0.13
F2	0.91	0.78	0.12	0.83	0.56	0.15	1	0.86	0.08	0.85	**0.84**	0.04
F3	1	**0.86**	0.08	0.83	0.53	0.12	1	**0.88**	0.05	1	0.83	0.08
F4	0.75	0.66	0.04	0.83	**0.63**	0.16	0.87	0.65	0.13	0.57	0.38	0.10
F5	0.83	0.70	0.07	1	0.60	0.13	1	0.94	0.13	0.71	0.34	0.12

Average and deviation of 20 executions

Table 9 Wilcoxon test results for *person* taxonomy

	F1	F2	F3	F4	F5
F1	1				
F2	$9.5\,e^{-9}$	1			
F3	$1.3\,e^{-12}$	0.007	1		
F4	0.04	$3.8\,e^{-9}$	$8.5\,e^{-15}$	1	
F5	0.0006	$2.9\,e^{-5}$	$2.2\,e^{-10}$	0.001	1

in Fig. 8. Figure 9 shows the results of accuracy obtained with different number of nodes. Specifically, it shows the best result obtained in ten runs (Best), the average of ten runs (Average) and also the baseline. Looking at the baseline, we can see how the difficulty of the problem significantly increases as we add more nodes to the set of concepts to be arranged in a taxonomy, leading to a decrease in accuracy. However, despite the difficulty of the problem, the results obtained by the function provided by the GE algorithm are still valuable, obtaining accuracy values which range from 1 to 0.5 for the best taxonomy found by the GA.

Fig. 8 Larger version of the person taxonomy

7 Conclusions

The GE algorithm presented in this work is able to produce functions that correctly identify some taxonomies among Wikipedia concepts, such as *plant*, *animal*, *person*, *time period*, *musical work* and *means of transport*. Even in the cases in which the obtained taxonomy does not match the DBpedia ontology used as reference, we can see that the method is able to detect real relationships such as the ones between *insect* and *arachnid*, *crustacean* and *fish*, *playboy* and *celebrity* or *discography* and *EurovisionSongContest*. Best results are obtained between groups of concepts which are directly connected in Wikipedia. Results get worse for the most general concepts, such as *species*, the top of the considered part of the DBpedia ontology in the training set. The GE algorithm has been able to provide valuable functions that combine the considered features extracted from the Wikipedia pages. However, other features can be extracted from the Wikipedia pages, and the proposed algorithm can be used to find the best function to combine them.

There is a lot of work that can be done to improve the results of this proposal, apart from the mentioned introduction of additional features from the Wikipedia pages. We can also look for different ways of evaluating the taxonomy in the GA used by the GE algorithm. In this work, the candidate functions have been evaluated by applying it to each couple of nodes connected by a parent–child relationship in the taxonomy being evaluated. However, other relationships can be considered, as those between a node and all its ancestors. Concerning the GE algorithm, we plan to explore possible improvements in the generation of constants for the candidate functions. Though the proposed algorithm has been applied to linked Wikipedia pages in order to eval-

Journal: **500** MS: **2544** ☐ TYPESET ☐ DISK ☐ LE ☐ CP Disp.:**2017/3/15** Pages: **13** Layout: **Large**

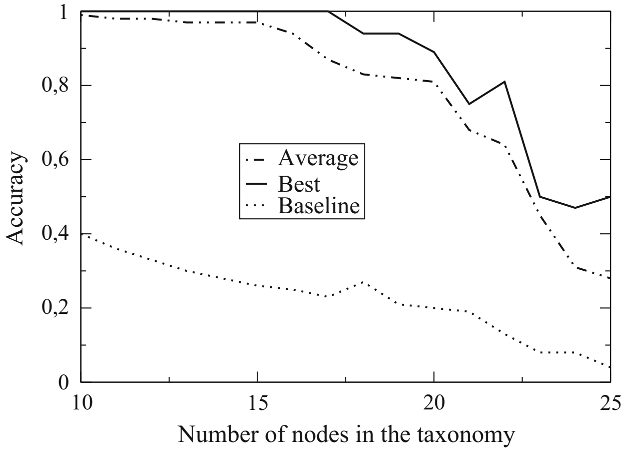


Fig. 9 Accuracy obtained using function *F3* from Table 4 for a person taxonomy with different number of nodes using the GA parameters shown in Table 2

uate the results using the DBpedia taxonomy as reference, it can also be applied to other kind of linked web pages. We also consider to explore other kind of relationships between Wikipedia concepts. In the current system, the features that have been included as variables in the BNF grammar—cosine, AWD(average weight of distinct terms), etc.—are specifically designed to capture the subclass relationship. However, other features could be included for detecting more specific semantic relationships. For example, the semantic relationship IS-PART-OF can be found in the Wikipedia page for *car*, which says “These controls include a steering wheel,...,” where steering wheel is a link to the corresponding page. This relationship could be discovered by including new features related to the presence of some particular expressions referring to “to be part of.”

Acknowledgements This work has been partially supported by the Spanish Ministry of Science and Innovation within the projects EXTRECM (TIN2013-46616-C2-2-R) and PROSA-MED (TIN2016-77820-C3-2-R), as well as by the Universidad Nacional de Educación a Distancia (UNED) through the FPI-UNED 2013 Grant. The authors would like to thank the referees for their valuable comments which led to improvements in the paper.

Compliance with ethical standards

Conflict of interest Lourdes Araujo declares that she has no conflict of interest. Juan Martinez-Romo declares that he has no conflict of interest. Andres Duque declares that he has no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

Ali E, Raghavan V (2015) Extending skos: A wikipedia-based unified annotation model for creating interoperable domain ontologies. In: Esposito F, Pivert O, Hacid MS, Rás ZW, Ferilli S (eds) Proceedings of the 22nd international symposium on foundations of intelligent systems. Springer, pp 364–370

Araujo L, Martinez-Romo J, Duque A (2015) Grammatical evolution for identifying wikipedia taxonomies. In: Genetic and evolutionary computation conference, GECCO 2015, Madrid, Spain, July 11–15, 2015, companion material proceedings, pp 1345–1346

Bartoli A, De Lorenzo A, Medvet E, Tarlao F (2016) Syntactical similarity learning by means of grammatical evolution. In: Handl J, Hart E, Lewis PR, López-Ibáñez M, Ochoa G, Paechter B (eds) Proceedings of parallel problem solving from nature—PPSN XIV. Springer, pp 260–269

Ben Aouicha M, Hadj Taieb MA, Ezzeddine M (2016) Derivation of “is” taxonomy from wikipedia category graph. Eng Appl Artif Intell 50(C):265–286. doi:10.1016/j.engappai.2016.01.033

Bhogal J, Macfarlane A, Smith P (2007) A review of ontology based query expansion. Inf Process Manag 43(4):866–886

Camous F, Blott S, Smeaton A (2007) Ontology-based medline document classification. In: Hochreiter S, Wagner R (eds) Bioinformatics research and development. Lecture notes in computer science, vol 4414. Springer, Berlin, pp 439–452. doi:10.1007/978-3-540-71233-6_34

Cerri R, Barros RC, Freitas AA, de Carvalho AC (2014) Evolving relational hierarchical classification rules for predicting gene ontology-based protein functions. In: Proceedings of the 2014 conference companion on genetic and evolutionary computation companion, GECCO Comp ’14. ACM, New York, pp 1279–1286

Chernov S, Iofciu T, Nejdl W, Zhou X (2006) Extracting semantics relationships between wikipedia categories. In: Völkel M, Schaffert S (eds) Proceedings of the first workshop on semantic wikis-from wiki to semantics, ESWC2006. Workshop on semantic wikis

Clarke LE (1958) On Cayley’s formula for counting trees. J Lond Math Soci 33(4):471–474 [4]

Dempsey I, O’Neill M, Brabazon A (2007) Constant creation in grammatical evolution. Int J Innov Comput Appl 1(1):23–38

Forsati R, Shamsfard M (2016) Symbiosis of evolutionary and combinatorial ontology mapping approaches. Inf Sci 342(C):53–80

Galitsky BA (2013) Transfer learning of syntactic structures for building taxonomies for search engines. Eng Appl Artif Intell 26(10):2504–2515

Geem ZW, Kim JH, Loganathan G (2001) A new heuristic optimization algorithm: harmony search. Simulation 76(2):60–68

He P, Deng Z, Gao C, Wang X, Li J (2016) Model approach to grammatical evolution: deep-structured analyzing of model and representation. Soft Comput 1–11 [5]

Herbelot A, Copestake A (2006) Acquiring ontological relationships from wikipedia using rmrs. In: Proceedings of the ISWC 2006 workshop on web content mining with human language technologies

Hovy E (1998) Combining and standardizing large-scale, practical ontologies for machine translation and other uses. In: Language resource and evaluation conference. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.66.8225

Isele R, Bizer C (2013) Active learning of expressive linkage rules using genetic programming. Web Semant Sci Serv Agents World Wide Web 23(0) [6]

Khalatbari S, Mirroshandel SA (2015) Automatic construction of domain ontology using wikipedia and enhancing it by google search engine. J Inf Syst Telecommun 3:248–258

Koza JR (1992) Genetic programming: on the programming of computers by means of natural selection. MIT Press, cambridge

Lehmann J, Isele R, Jakob M, Jentzsch A, Kontokostas D, Mendes PN, Hellmann S, Morsey M, van Kleef P, Auer S, Bizer C (2014) DBpedia–a large-scale, multilingual knowledge base extracted from wikipedia. Semant Web J 6:167–195 [7]

Manning CD, Raghavan P, Schütze H (2008) Introduction to information retrieval. Cambridge University Press, New York

Mao Y (2001) A semantic-based genetic algorithm for sub-ontology evolution. Inf Technol J 9(4):609–620

Medelyan O, Milne D, Legg C, Witten IH (2009) Mining meaning from wikipedia. Int J Hum Comput Stud 67(9):716–754

Miles A, Bechhofer S (2008) SKOS simple knowledge organization system reference. Working draft, W3C. http://www.w3.org/TR/skos-reference

Morales LP, Esteban AD, Gervás P (2008) Concept-graph based biomedical automatic summarization using ontologies. In: Proceedings of the 3rd textgraphs workshop on graph-based algorithms for natural language processing. Association for Computational Linguistics, Stroudsburg, pp 53–56

Nakayama K, Hara T, Nishio S (2007) A thesaurus construction method from large scale web dictionaries. In: Proceedings of the 21st IEEE international conference on advanced information networking and applications, AINA07. IEEE Computer Society, pp 932–939

Navigli R, Velardi P, Gangemi A (2003) Ontology learning and its application to automated terminology translation. Intell Syst IEEE 18(1):22–31

Nguyen DPT, Matsuo Y, Ishizuka M (2007) Exploiting syntactic and semantic information for relation extraction from Wikipedia. In: IJCAI workshop on text-mining and link-analysis (TextLink 2007)

O'Neill M, Ryan C (2001) Grammatical evolution. IEEE Trans Evol Comput 5(4):349–358

Otero FEB, Freitas AA, Johnson CG (2009) A hierarchical classification ant colony algorithm for predicting gene ontology terms. In: Pizzuti C, Ritchie MD, Giacobini M (eds) EvoBIO. Lecture notes in computer science, vol 5483. Springer, pp 68–79

Othman RM, Deris S, Illias RM, Alashwal HT, Hassan R, Farhan M (2007) Incorporating semantic similarity measure in genetic algorithm: an approach for searching the gene ontology terms. Int J Comput Intell 1(12):325–334

Ponzetto SP, Strube M (2007) Deriving a large scale taxonomy from wikipedia. In: AAAI'07, Proceedings of the 22nd national conference on artificial intelligence, vol 2. AAAI Press, pp 1440–1445

Prokofyev R, Demartini G, Boyarsky A, Ruchayskiy O, Cudr-Mauroux P (2013) Ontology-based word sense disambiguation for scientific literature. In: Serdyukov P, Braslavski P, Kuznetsov S, Kamps J, Rger S, Agichtein E, Segalovich I, Yilmaz E (eds) Advances in information retrieval. Lecture notes in computer science, vol 7814. Springer, Berlin, pp 594–605

Ruiz-Casado M, Alfonseca E, Castells P (2005) Automatic extraction of semantic relationships for wordnet by means of pattern learning from wikipedia. In: NLDB, pp 67–79

Salton G, Wong A, Yang CS (1975) A vector space model for automatic indexing. Commun ACM 18(11):613–620

Schlegel DR, Crowner C, Elkin PL (2015) Automatically expanding the synonym set of SNOMED CT using wikipedia. In: MEDINFO 2015: eHealth-enabled Health—Proceedings of the 15th world congress on health and biomedical informatics, São Paulo, Brazil, 19–23 August 2015, pp 619–623

Suchanek FM, Ifrim G, Weikum G (2006) Combining linguistic and statistical analysis to extract relations from web documents. In: KDD '06, Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, New York, pp 712–717

Suchanek FM, Kasneci G, Weikum G (2007) Yago: A core of semantic knowledge. In: WWW '07, Proceedings of the 16th international conference on world wide web. ACM, New York, pp 697–706

Vicient C, Sánchez D, Moreno A (2013) An automatic approach for ontology-based feature extraction from heterogeneous textualresources. Eng Appl Artif Intell 26(3):1092–1106

Weber N, Buitelaar P (2006) Web-based ontology learning with isolde. In: Proceedings of the workshop on web content mining with human language at the international semantic web conference

Wu F, Weld DS (2007) Autonomously semantifying wikipedia. In: CIKM '07, Proceedings of the sixteenth ACM conference on conference on information and knowledge management. ACM, New York, USA, pp 41–50