Multiscale Parameter Tuning of a Semantic Relatedness Algorithm

Conference Paper · January 2014

DOI: 10.4230/OASIcs.SLATE.2014.201

2 authors:

Teresa Costa
University of Porto
6 PUBLICATIONS **6** CITATIONS
SEE PROFILE

José Paulo Leal
University of Porto
97 PUBLICATIONS **288** CITATIONS
SEE PROFILE

Multiscale Parameter Tuning of a Semantic Relatedness Algorithm

José Paulo Leal[1] and Teresa Costa[2]

1 **CRACS & INESC-Porto LA, Faculty of Sciences, University of Porto**
Porto, Portugal
zp@dcc.fc.up.pt

2 **CRACS & INESC-Porto LA, Faculty of Sciences, University of Porto**
Porto, Portugal
teresa.costa@dcc.fc.up.pt

Abstract

The research presented in this paper builds on previous work that lead to the definition of a family of semantic relatedness algorithms that compute a proximity given as input a pair of concept labels. The algorithms depends on a semantic graph, provided as RDF data, and on a particular set of weights assigned to the properties of RDF statements (types of arcs in the RDF graph). The current research objective is to automatically tune the weights for a given graph in order to increase the proximity quality. The quality of a semantic relatedness method is usually measured against a benchmark data set. The results produced by the method are compared with those on the benchmark using the Spearman's rank coefficient. This methodology works the other way round and uses this coefficient to tune the proximity weights. The tuning process is controlled by a genetic algorithm using the Spearman's rank coefficient as the fitness function. The genetic algorithm has its own set of parameters which also need to be tuned. Bootstrapping is based on a statistical method for generating samples that is used in this methodology to enable a large number of repetitions of the genetic algorithm, exploring the results of alternative parameter settings. This approach raises several technical challenges due to its computational complexity. This paper provides details on the techniques used to speedup this process. The proposed approach was validated with the WordNet 2.0 and the WordSim-353 data set. Several ranges of parameters values were tested and the obtained results are better than the state of the art methods for computing semantic relatedness using the WordNet 2.0, with the advantage of not requiring any domain knowledge of the ontological graph.

1998 ACM Subject Classification E.1 Graphs and networks, G.2.2 Graph theory, Path and circuit problems, H.3.1 Content Analysis and Indexing, I.2.4 Knowledge Representation Formalisms and Methods, Semantic networks, I.2.8 Problem Solving, Control Methods, and Search, Graph and tree search strategies

Keywords and phrases semantic similarity, linked data, genetic algorithms, bootstrapping, WordNet

Digital Object Identifier 10.4230/OASIcs.SLATE.2014.201

1 Introduction

Consider a magazine, a pencil and a notepad. Of these three items which is the most related pair? Is it magazine and pencil, pencil and notepad, or newspaper and notepad? People living in more individualistic societies tend to find the magazine and the notepad more related, since they are both made of sheets of paper; while people living in more collectivist societies tend to find the pencil and the notepad more related, since they complement each

3rd Symposium on Languages, Applications and Technologies (SLATE'14).
Editors: Maria João Varanda Pereira, José Paulo Leal, and Alberto Simões; pp. 201–213

OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

other (a pencil writes on notepad) [7]. The differences are even more striking when people are asked to assign a value to the relatedness [6]. These experiments reveal the lack of a standard definition of relatedness and the difficulty to measure the relatedness of two concepts.

A standard approach to measure relatedness is to use an ontology [13]. An ontology is a formal and explicit specification of the relationships between concepts. For instance, a thesaurus is a kind of ontology specifying the relationships between words: synonymy and antonymy, as well as hyponymy (words whose semantic field encloses other words, e.g., mammal has as hyponym horse) and hypernymy (words whose semantic field is enclosed in other words, e.g, horse has a hypernym mammal).

This paper presents ongoing work aiming at the development of a new methodology to determine the semantic relatedness between two concepts. This methodology is ontology based, can be applied to an ontological graph, and does not require any knowledge of the ontological domain. It uses a family of semantic relatedness algorithms based on the notion of proximity [10]. An algorithm of this family is parametrized by a semantic graph and a set of weights. The semantic graph is provided as RDF data, where the resources are the graph nodes and the properties are the arcs. Each type of arc has a specific weight value. Tuning these weights in order to improve the quality of the semantic relatedness is the current objective of this research.

Other methods available in the literature [1, 11, 13] measure the quality of their algorithms using as benchmark a standard data set [6]. The reference similarity of concept pairs is the average similarity assigned by a group of persons. The relatedness computed with an algorithm is compared against those of the benchmark using the Spearman's rank order correlation. The quality of an algorithm is as high as the value of this correlation.

A measure of quality is essential for using a genetic algorithm to tune weight values. In this tuning approach, an assignment of values to weights is encoded as a set of genes of a chromosome. New chromosomes are obtained by crossover and mutation of the chromosomes from the previous generation and the best are selected using a fitness function plus randomness. The fitness function receives as input a weight assignment and returns the Spearman's rank order correlation for a subset of benchmark data.

The genetic algorithm has in turn its own set of parameters that need to be tuned. Bootstrapping is done through a statistical procedure that produces a large number of samples that is used to explore the most promising settings of the genetic algorithm. The best settings are finally used to run the genetic algorithm a large number of times with the complete data set.

This methodology was validated with the ontology of WordNet 2.0 [5], using as benchmark the WordSim-353 [6] data set. The obtained results were better than the best results available on the literature for the same ontology and benchmark [13, 9].

Due to its computational complexity this tuning methodology raises several technical challenges. Firstly, the semantic algorithms must collect a large number of paths connecting each pair of labels in the graph. Secondly, in order to compute the Spearman's rank order, the semantic relatedness algorithm must be executed with several hundreds of pairs of concept labels. Thirdly, the genetic algorithm must compute a correlation for each chromosome (a set of weight assignments) in the genetic pool for hundreds of generations. And finally, the evolution process of the genetic algorithm has to be repeated hundreds of times as part of the bootstrapping method. This paper presents also approaches used to speedup the tuning process.

The rest of the paper is organized as follows. The next section present the state of the art on semantic relatedness. Section 3 describes the tuning methodology and Section 4 details

its implementation. The experimental results and their analysis can be found in Section 5. Finally, Section 6 summarizes this work and identifies opportunities for further research.

2 Related Work

The problem of computing semantic relatedness can be approached in several ways. Most approaches fall in one of two types: path methods, based on the topology of the relationships between concepts; and content methods, based on the frequency of word occurrence in corpora. In many cases the paths relating the concepts traverse an ontology. The research described in this paper follows the ontological approach.

Some of the ontological methods use only the underlying taxonomy, for instance, the taxonomy created by the *is-a* relationships, or the hypernymy and hyponymy relationships of a thesaurus. An example of this kind of approach is the work of Mazuel and Sobouret [11]. Their approach measures the relatedness based on the taxonomical part of the ontology of the WordNet and discards paths that are not "semantically correct" working only with a subset of "semantically correct" paths. To measure the semantic distance this methodology selects the best one from the subset.

Other ontological methods explore the full range of relationships in an ontology. An example of this approach is the work of Hirst and St-Onge [8] that used the WordNet as a knowledge source to create a lexical *chainer* (SIC). A lexical chain is a chain where words are included if they have a cohesive relationship with another word already in the chain. In this work they defined three types of relations: extra-strong, strong and medium-strong. The weight of a relation is higher as stronger is the relationship between the words.

Some path approaches use also statistical concepts. J. Garcia and E. Mena [2] developed a method that uses the Web as knowledge source, based on the Normalized Google Distance. This approach uses the frequencies of concepts provided by search engines to define a new semantic relatedness measure among ontology terms.

The work of Michael Strube and Simone Paolo Ponzetto [13] analyses several path and content approaches to choose the best one. The approaches they analysed were assigned to three categories: path, content and text overlap. The approaches in this last category compute an overlap score by using stemming to explore related words.

Several of the mentioned approaches use the WordNet. The WordNet [5] is a large lexical knowledge base of English words. It groups nouns, verbs, adjectives and adverbs into *synsets* (sets of cognitive synonyms) that express distinct concepts. *Synsets* are interlinked by lexical and conceptual-semantic relationships. This knowledge base is well-known and widely used but lacks some specialized vocabularies and named entities, such as Diego Maradona or Freddie Mercury. On the other hand, it is a comparatively small knowledge base and thus it is ideal for the initial tests of a tuning methodology.

3 Multiscale Weight Tuning

This section is to describe an approach for tuning weights in a family of semantic relatedness algorithms. The proposed approach for tuning weights operates at different scales. Although each scale has its own distinctive features, there are self-similar patterns common to all scales.

Consider a fractal as a metaphor. At each scale a fractal exhibits features that are found also on other fractal dimensions. That is, if we zoom in (or zoom out) on a fractal we observe a identical pattern. Mathematical fractals are exact and infinite repetitions of the same

pattern. Nonetheless, fractals observed in nature, such as shells, leaves or coastlines, exhibit self-similar patterns that are neither infinite nor an exact repetition of other scales.

In this tuning approach, the common pattern is the concept of function, with input and output values, and a set of parameters that can be tuned. At the lowest scale this function is the semantic relatedness algorithm. It takes as input a pair of strings and produces a value in the interval [0,1]. This function takes as parameters a semantic graph and a set of weights that must be tuned.

Zooming out to the next scale there is a genetic algorithm. A genetic algorithm can be seen as a function taking another function as input a fitness function and producing a result. It has also its own set of parameters that need to be tuned: the number of generations, the mutation rate, etc.. In this case the fitness function takes as input a set of weights and computes the correlation between the relatedness obtained by the algorithm and the standard benchmark. Hence, each application of a genetic algorithm (seen as a function) aggregates thousands of applications of functions from the previous scale – the semantic relatedness algorithm.

Continuing to zoom out to the next and final scale there is a statistical method, the bootstrapping method. This method measures the accuracy of the results obtained by the genetic algorithm with different parameter sets. It can also be seen as a function taking as input genetic algorithms, the candidates for producing a weight tuning, and producing an estimate of which is the best. Again, each application of the bootstrapping method (seen as a function) aggregates thousands of applications of the of functions from the previous scale – the genetic algorithm – since each candidate configuration set is repeated hundreds of times.

The following subsections detail each of these "fractal scales". Each scale describes the function that is used as input for its upper scale, identifying its parameters. At least metaphorically, these functions can be seen as a self-similar pattern that is present in the three different layers in which the proposed approach operates.

3.1 Proximity Measure Layer

The core of the methodology for calculate proximity between concepts is an algorithm to compute semantic relatedness using ontological information in RDF graphs. It uses the notion of proximity, rather than distance, as the underlying concept for computing semantic relatedness between two nodes.

Concepts in ontological graphs are represented by nodes. Take for instance the music domain. Singers, bands, music genres, instruments or virtually any concept related to music is represented as nodes in an ontology. These nodes are related by properties, such as `has genre` connecting singers to genres, and thus form a graph. This graph can be retrieved in RDF format using the SPARQL endpoint of a knowledge base, such as DBpedia or Freebase.

The core idea of the research presented in this paper is to use the RDF graph to compute the relatedness between nodes. Actually, the goal is the relatedness between terms, but concept nodes of this graph typically have a label – a string representation or stringification – that can be seen as a term.

At first sight relatedness may seem to be the inverse of the distance between nodes. Two nodes far apart are unrelated and every node is totally (infinitely) related to itself. Interpreting relatedness as a function of distance has an obvious advantage: computing distances between nodes in a graph is a well studied problem with several known algorithms. After assigning a weight to each arc one can compute the distance as the minimum length of all the paths connecting the two nodes.

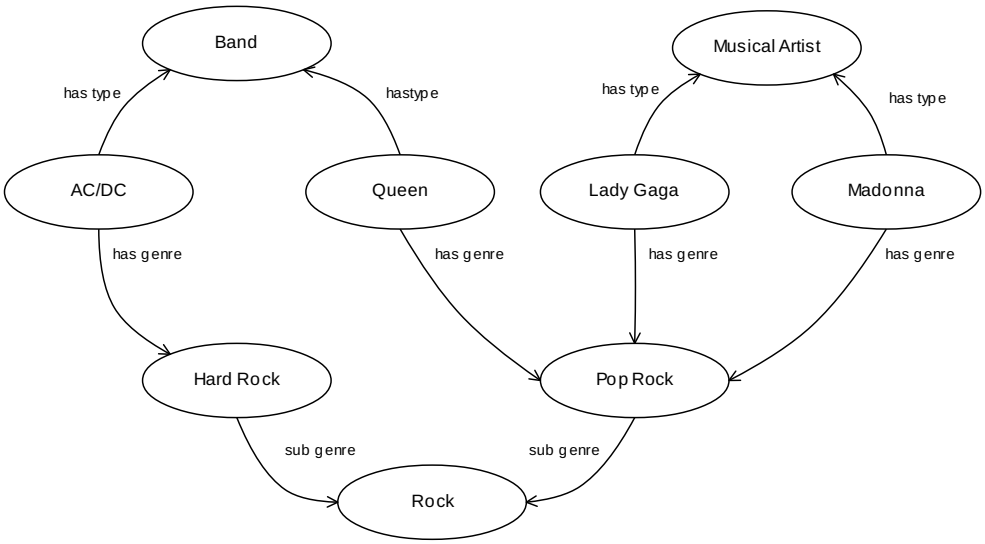

Figure 1 RDF graph for concepts in music domain.

On a closer inspection this interpretation of relatedness as the inverse of distance reveals some problems. Consider the graph in Figure 1. Depending on the weight assigned to the arcs formed by the properties `has type` and `has genre`, the distances between Lady Gaga, Madonna and Queen are the same. If the `has genre` has less weight than `has type`, this would mean that the band Queen is as related to Lady Gaga as Madonna, which obviously should not be the case. On the other hand, if `has type` has less weight than `has genre` then Queen is more related to AC/DC than to Lady Gaga or Madonna simply because they are both bands, which also should not be the case.

In the semantic relatedness methodology proposed, we consider *proximity* rather than distance as a measure of relatedness among nodes. By definition[1], proximity is closeness; the state of being near as in space, time, or relationship. Rather than focusing solely on minimum path length, proximity balances also the number of existing paths between nodes. As an example consider the proximity between two persons. More than resulting from a single common interest, however strong, it results from a collection of common interests.

With this notion of proximity, Lady Gaga and Madonna are more related to each other than with Queen since they have two different paths connecting each other, one through `Musical Artist` and another `Pop Rock`. By the same token the band Queen is more related to them than to the band AC/DC.

An algorithm to compute proximity must take into account the several paths connecting two nodes and their weights. However, paths are made of several edges, and the weight of an edge should contribute less to proximity as it is further away in the path. In fact, there must be a limit in number of edges in a path, as RDF graphs are usually connected graphs.

The main issue with this definition of proximity[2] is how to determine the weights of

[1] `https://en.wiktionary.org/wiki/proximity`
[2] See [10] for a detailed description of the algorithm.

transitions. The first attempt was to define these weights using domain knowledge. For instance, when comparing musical performers one may consider that being associated with a band or with another artist is more important than their musical genre, and that genre is more important than their stylistics influences and even more important than instruments they play.

This naïve approach to weight setting has several problems. Firstly, this kind of "informed opinion" frequently has no evidence to support it, and sometimes is plainly wrong. How sure can one be that stylistics influences should weight more than the genre in musical proximity? Even if it is true sometimes, how can one be sure it is true in most cases? Secondly, this approach is difficult to apply to a large ontology encompassing a broad range of domains. Is a specialist required for every domain? How should an ontology be structured in domains? What domain should be considered for concepts that fall in multiple domains? To be of practical use, the weights of a proximity based semantic relatedness algorithm must be automatically tuned.

3.2 Genetic Algorithm Layer

Genetic algorithms are a family of computational models that mimic the process of natural selection in the evolution of the species. These algorithms use the concepts of *variation*, *differential reproduction* and *heredity* to guide the co-evolution of a set of problem solutions. This type of algorithm is frequently used to improve solutions of optimization problems [14].

There are two necessary conditions for using a genetic algorithm. Firstly, the different candidate solutions must be representable as individuals (*variation*). This encoding of an individual solution is sometimes called a *chromosome* which are a collection of *genes* that characterize the solution. Secondly, it must be possible to compare a set of individuals, decide which are the fittest and allow them to pass their genetic information to the next generation (*differential reproduction*). Also, the representation of solutions as individuals must allow their recombination with other solutions (*heredity*) so that favorable traits are preferred over unfavorable ones as the population of solutions evolves.

A simple approach in the case of weight tuning is to consider as *individual* a vector of weight values. This representation contrasts with the binary representations typically used in genetic algorithms [4]. However it is closer to the domain and it can be processed more efficiently with large number of weights.

Genetic algorithms introduce variance also by *mutation*. The are a number of mutation operators, such as swap, scramble, insertion, that can be used on binary representations [4]. However, the approach taken to represent individuals in this methodology makes these kind of mutations less interesting. Since weights are independent from each other, swapping values among them is as likely to improve the solution as selecting a new random values. Hence, the genetic algorithm created for tuning weights has a single kind of mutation: randomly selecting a new value for a given "gene".

The fitness function plays a decisive role in selecting the new generation of individuals, created by crossover and mutation of their parents. The usual method for estimating the quality of a semantic relatedness function is to compare it with a benchmark data set. The benchmark data set contains pairs of words and their relatedness.

The Spearman's rank order coefficient is commonly used to compare the relatedness values in the benchmark data set with those produced by a semantic relatedness algorithm. Rather than the simple correlation between the two data series, the Spearman's rank order sorts those data series and correlates their rank.

The genetic algorithm of this weight tuning methodology uses as fitness function the

Spearman's rank order coefficient on benchmark data, using as input a vector of weight values assigned to each arc type.

3.3 Bootstrap Layer

The genetic algorithm itself has a number of parameters that must be tuned. Generic parameters of a genetic algorithm include the number of generations and the mutation rate. In this particular case the range of values that may be assigned to weights must also be considered.

Several approaches to tuning parameters of genetic algorithms have been proposed and compared [12]. Although with different approaches, these methods highlight the advantage of using automated parameter tuning over tuning based on expert "informed opinions". In many cases the best solution contradicts the expert best intuitions.

The proposed methodology relies on a single benchmark data set to compare alternative weight attributions. To repeat a large number of experiments using the genetic algorithm to co-evolve a set solutions one needs a larger test sample. Bootstrapping [3] is a statistical method for assigning measures of accuracy to data samples, using simple techniques known as *resampling*.

Resampling is applied to the original data set to build a collection of sample data sets. Each sample data set has the same size as the original data set and is build from the same elements. If the original data set has size n then n elements from that set are randomly chosen to create the sample set. When an element is selected it is not removed from the original data set. Hence, a particular element may occur repeatedly on the sample data set while other may not occur at all.

The bootstrapping method is used for comparing different approaches. Each approach is repeated a large number of times, typically 200, each time with a different sample set. Each approach is summarized by a statistics, such as the mean or the third quartile. In the end, these statistics are compared to select the most effective approach. Since the objective is to select the approach that may lead to the highest Spearman's coefficient, the third quartile is specially relevant since it is a lower bound of the largest solutions.

In this tuning methodology, each approach corresponds to a particular setting of the genetic algorithm. Candidate settings include values for parameters such as the number of generations or the mutation rate. Another important parameter that is specific to this methodology is the range of values that are used as possible values for weights. As these values have to be enumerated, this methodology considers only integer values bellow a certain threshold.

As the result of the bootstrapping method a particular setting of the genetic algorithm's parameters is selected. The final stage is to run the genetic algorithm with these settings, using the full benchmark data set in the fitness function. The selected genetic algorithm is repeated an even larger number of times, typically 1000, and the best result is selected as weights for the relatedness algorithm.

4 Implementation

The methodology presented for parameter tuning has a high computational complexity. At its core it has to find all paths connecting two concepts to compute a single proximity. To test the quality of a vector of weights, the proximity has to be computed for each pair of concepts in a benchmark data set. Bootstrapping repeats 200 times the genetic algorithm for each setting, and this process is repeated for a large number of settings.

The strategy used for improving the efficiency of the methodology has three main components: graph pre-processing (described in the Subsection 4.1), factorization of the proximity algorithm and concurrent evaluation of the bootstrapping method (described in the Subsection 4.2). The remainder of this section details each of these components.

4.1 Graph Pre-processing

The computation of the semantic proximity between two concepts depends on a data graph search that finds all the paths that connect both concepts. The data graph search is implemented in two different ways, supporting queries of remote and local data. The main differences are the methods that retrieve the nodes with a specific label and the methods used to retrieve the transitions from a node used by the semantic relatedness algorithm.

Remote data is usually retrieved from SPARQL endpoints. A SPARQL endpoint is addressed by a URI to which SPARQL queries can be sent and which returns RDF as a response. Paths are built from data collected from two SPARQL queries. The following query retrieves the list of all nodes that have a given string as label. This query is executed twice, one for each label. For each node retrieved with the previous SPARQL query another SPARQL query is executed, as shown bellow, until the set of paths connecting both concepts are finished.

The SPARQL approach raises a number of issues. Firstly, the endpoint or network may be under maintenance or with performance problems. Secondly, some endpoints have configuration problems and do not support queries with some operators, such as `UNION`. And thirdly, the SPARQL queries can have performance issues, mainly when using operators such as `DISTINCT`, and having a large amount of queries per proximity search can cause a huge impact at the execution time.

In order to avoid those issues, this methodology also implements searches in local data. Knowledge bases often provide dumps of their data. Local data are preprocessed RDF graphs that are stored in the local file system, retrieved from those dumps. Graph pre-processing begins with parsing the RDF data. RDF data can be retrieved in several formats, such as Turtle, RDF/XML or N-Triples. To simplify this process, all RDF data is converted to N-Triples, since this is the simplest RDF serialization.

This process takes some time to execute but it is only necessary to execute it once. Also, the most used data is cached in memory which has a significant impact on performance.

The proximity algorithm is based on a previous definition [10]. This algorithm takes two strings as labels and builds a set with all the paths that connect both concepts. In this current implementation, there is a stemming process with labels aiming to increase the meaning scope of each word.

The computation of the proximity of a single pair of concepts using the WordNet 2.0 SPARQL endpoint[3] takes about 20 minutes. With the pre-processed graph[4] that is executed once and takes 30 minutes, the same computation takes about 6 seconds.

4.2 Other Optimizations

Traversing the graph searching for paths connecting two labels is the most frequently executed part of this semantic relatedness methodology. Nevertheless, this procedure is almost the same for each pair of concepts, varying only on the weights that are used for each arc type.

[3] `http://wordnet.rkbexplorer.com/sparql/`
[4] The tests were executed in a 8 core machine at 3.5 GHz and 16Gb of RAM

This computation is repeated many times since the exact same pair of concepts is used each time that the genetic algorithm is run.

The solution found was to alter the proximity algorithm to compute the set of coefficients that are multiplied to each weight. These coefficients are organized in a vector, using the same order of the weight vector used in the genetic algorithm. Thus, computing the proximity of a pair of concepts given a different weight vector is just the inner product of the weight vector and the coefficient vector.

With this modification a single run of the genetic algorithm with 200 generations takes less than a 1 minute and computing the coefficients for all the pairs takes about 30 minutes.

The final optimization was concurrent evaluation of the bootstrapping method. Each of the settings can be processed independently, hence they could be assigned to a different processor of a multi-core machine. Each run of the bootstrapping method takes about 200 minutes. It run 120 configurations that sequentially would take more than 16.5 days in about 2 days.

5 Validation

The validation of the proposed tuning approach consisted of tuning the weights of the relatedness algorithm for WordNet 2.0. The tuning was performed in two rounds. In the first round a large number of settings was explored to determine which were the most relevant. A second round was then performed to explore new settings on those parameters that have more impact on performance.

The tuning process uses as benchmark the WordSimilarity-353 data set [6]. It has 353 pairs of concepts with the mean of the relatedness values given by humans. Since WordNet 2.0 does not have all the words listed in this data set, the pairs with missing elements were removed, creating a new data set with the non-missing pairs. In total 7 pairs were removed.

The bootstrapping process tests three parameters: weight values, mutation rate, and number of generations. The weight values were divided in positive and mixed (positive and negative) values; the positive values ranged in $[0, n]$ with $n \in \mathbb{N}^+$ and $n \leq 10$. The mixed values ranged in $[-n, n]$ for the same values of n. The mutation rate took values in the set $\{0.3, 0.4, 0.5\}$ values and the number of generations in $\{100, 200\}$. Permutating these values, 120 different sets of parameters were tested. Each set of parameters was executed 200 times in the bootstrapping process. The results of those tests can be seen in the following graphs. These graphs show the statistics of the correlation as function of a single variable: number of weight values, mutation rate and number of generations.

Figure 2 shows how the correlation evolves with different amounts and ranges of distinct weights. The correlation obtained when there are only positive values in the weight set is much lower than when positive and negative weights are used. The positive values also appear to reach a maximum value. However, the sets with positive and negative values do not show that stabilization, becoming relevant more tests with a larger range of values.

The graph on the left of Figure 3 shows the impact of changing the mutation rate. Despite the large overall variation, the mean and third quartile values are similar, showing that variations in this parameter have a small impact on the tuning process. Still, the variation of the maximums indicate the relevance of also testing lower mutation rates in the future.

The graph on the right of Figure 3 presents the variation of the number of generations. As it occurs with the mutation rate, changes in the number of generations have no significant impact in the correlation values.

After the first round, changes in range of weight values appear to have a higher impact

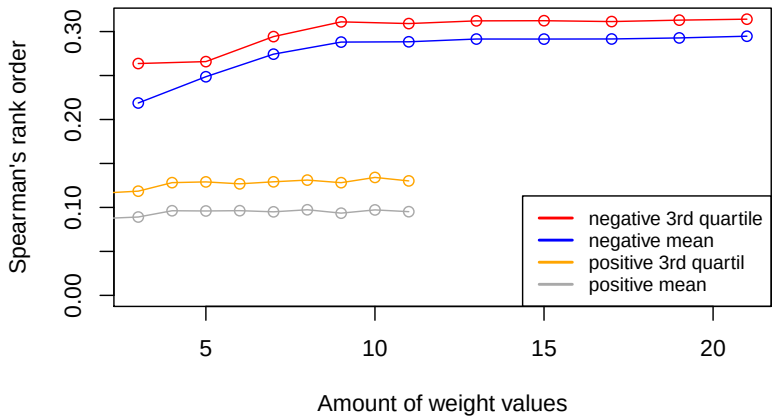

Figure 2 Graph of weights distribution.

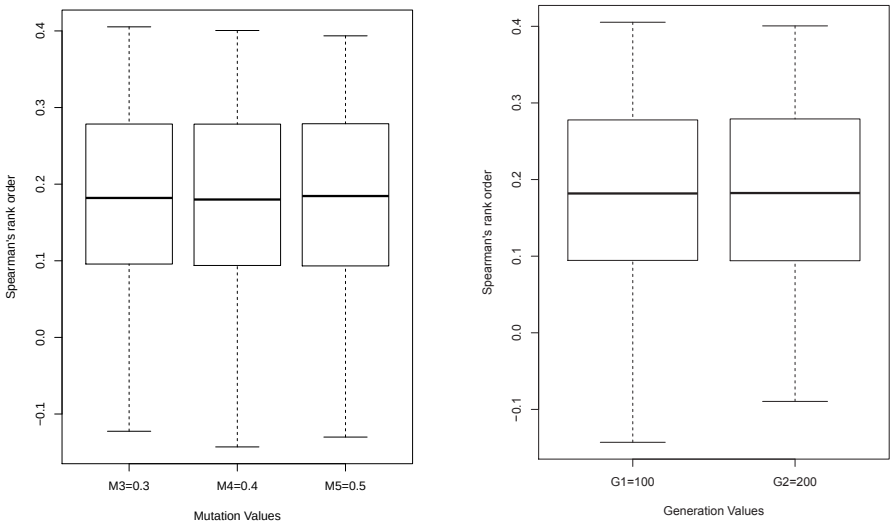

Figure 3 Distribution of different mutation rates (left) and number of generations (right).

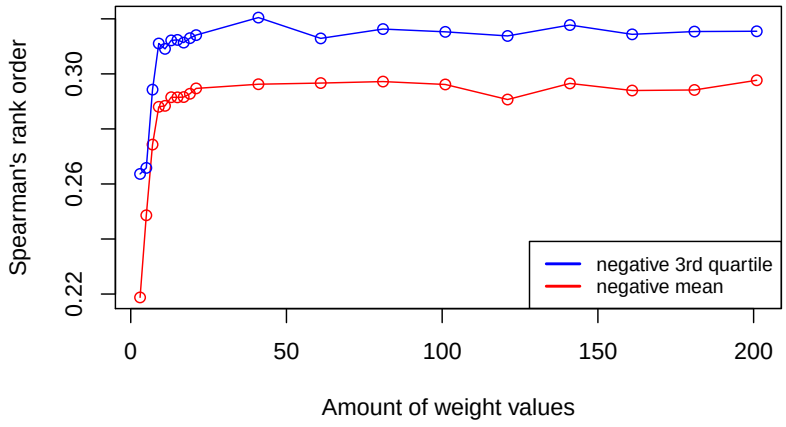

Figure 4 Graph of weights distribution.

Table 1 Weight values obtained after tuning process.

Edge type	Weight	Edge type	Weight
null	1	wn:classifiedByUsage	18
wn:memberMeronymOf	−2	wn:tagCount	1
wn:participleOf	−6	wn:sameVerbGroupAs	−8
wn:antonymOf	−5	wn:derivationallyRelated	0
wn:classifiedByTopic	19	wn:attribute	−15
wn:partMeronymOf	19	wn:synsetId	18
wn:word	12	wn:seeAlso	6
wn:gloss	19	rdfs:type	−2
wn:similarTo	−19	entails	−1
wn:containsWordSense	4	wn:classifiedByRegion	−9
wn:causes	−17	wn:adverbPertainsTo	12
wn:frame	7	wn:hyponymOf	9
wn:adjectivePertainsTo	3	wn:substanceMeronymOf	18

Table 2 Previous work with WordNet and WordSim-353.

Method	Spearman's rank order
Jarmasz (2003)	0.33 - 0.35
Strube and Ponzetto (2006)	0.36
Proposed method	**0.41**

in the correlation values, specially if they allow negative values, increasing the correlation as the range size increases. New tests were needed to investigate for how long the correlation continues to increase, if it converges to an asymptote, or if the correlation degrades after a certain threshold.

A new round of tests was made to investigate these hypothesis. This time only positive and negative values were used, with fixed values of mutation rate and number of generations. These new configurations uses ranges from $[-10 \times n, 10 \times n]$ with $n \in \mathbb{N}^+$ and $n \leq 10$. The mutation rate value was fixed at 0.4 and the number of generations was fixed at 200. The results are displayed in Figure 4. The values obtained by increasing the range of values show a maximum value at the range [-20,20]. Ranges with higher values seem to never exceed the Spearman's rank order obtained at that point, indicating that performance degrades after this threshold.

Using the best configuration obtained by the bootstrap process the genetic algorithm was executed 1000 times aiming to obtain the best correlation value and the related configuration.

The best Spearman's rank order value obtained was 0.409 and the corresponding weight set is listed in the Table 1. The edges with the prefix `wn` correspond to the WordNet 2.0 URI[5] and the prefix `rdfs` to the RDF Schema URI[6]. The edge type `null` is the custom edge created in the stemming process.

Table 2 compares the results obtained by tuning the edge weights without domain knowledge with other methodologies.

[5] `http://www.w3.org/2006/03/wn/wn20/schema/`
[6] `http://www.w3.org/1999/02/22-rdf-syntax-ns#type`

6 Conclusion

The major contribution of the research presented in this paper is a method for tuning a relatedness algorithm to a particular ontological graph, without requiring any domain knowledge on the graph itself. The results obtained with this approach for WordNet 2.0 are better than the state of the art for the same graph. A number of solutions to speedup graph processing and the evaluation of fitness functions are also relevant contributions.

The proposed tuning approach performs a multiscale parameter tuning of an ontology based semantic relatedness algorithm. The main feature of the base algorithm is the fact that it considers all paths in an ontological graph that connect two labels and computes the contribution of each path as a function of its length and of the type of its arcs (properties). The main issue of this algorithm is the selection of parameters (weight values for each type of arc) that maximize the quality of the relatedness algorithm.

The quality of semantic relatedness algorithms is usually measured against a benchmark data set. This data set consists of the relatedness of a set of words, defined as the mean of the relatedness attributed by a group a persons. The quality of the algorithm is computed as the Spearman's rank correlation coefficient between the relatedness produced by the algorithm and the relatedness given by the data set. By defining this correlation as a function of weight assignments it is possible to frame the problem of maximizing the quality of the relatedness algorithm as finding the maximum of a function.

Evolutionary algorithms in general, and genetic algorithm in particular, are popular choices for improving the quality of solutions. Using a genetic algorithm it is possible to use variation and selection to improve the Spearman's coefficient. The proposed genetic algorithm uses as chromosome a set of weights attributions. The range of values used in attributions, as well as the number of generations and the mutation rate are in turn parameters that must also be tuned.

The statistical method of bootstrapping was used to measure the accuracy of different parameter settings. This method generates diversity by producing many sample data sets from the original data set. Bootstrapping is used to compare the results of the genetic algorithm with different settings. After selecting the best candidate parameters for the genetic algorithm, this is rerun with the complete benchmark data set.

The proposed approach for tuning the parameters of the semantic relatedness algorithm was validated with Wordnet 2.0. The tuning procedure was actually executed twice. In the first run several parameters of the genetic algorithm were tested to conclude that the range of weight values is the decisive parameter, in particular if it is allowed to contain negative values. The variation of some of the parameters, such as mutation rate and number of generations, had no impact on the quality. Based on these findings a second run of the tuning procedure explored a wider range of values. It showed that quality improves with the width of range values but also that a small degradation occurs after a certain threshold. The genetic algorithm was finally repeated a large number of times with the settings selected by this approach and the maximum Spearman's correlation obtained is significantly higher than the best result reported on the literature for the same graph.

The Wordnet 2.0 graph used for the evaluation is comparatively small. It has just 26 different types of properties and 464.795 nodes. The next step is to investigate how this approach works with Wordnet 3.0 and with even larger graphs, such as the DBPedia or Freebase. Apart from the challenges of dealing with such large graphs, it will be interesting to compare the semantic relatedness potential of different graphs and try to combine them to improve the accuracy of the semantic relatedness algorithm.

Acknowledgements. This work is financed by the ERDF – European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through the FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project FCOMP-01-0124-FEDER-037281. Project "NORTE-07-0124-FEDER-000059" is financed by the North Portugal Regional Operational Programme (ON.2 – O Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF), and by national funds, through the Portuguese funding agency, Fundação para a Ciência e a Tecnologia (FCT). The authors would also like to thank Luis Torgo for his help in the use of statistical methods.

References

1 Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 19–27. Association for Computational Linguistics, 2009.

2 Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. Measuring semantic similarity between words using Web search engines. *Proceedings of the 16th international conference on World Wide Web*, 7:757–766, 2007.

3 Bradley Efron and Robert J. Tibshirani. *An introduction to the bootstrap*, volume 57. CRC press, 1994.

4 Agoston E. Eiben and James E. Smith. *Introduction to evolutionary computing*. Springer, 2003.

5 Christiane Fellbaum. *WordNet*. Wiley Online Library, 1999.

6 Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. Placing search in context: The concept revisited. In *Proceedings of the 10th international conference on World Wide Web*, pages 406–414. ACM, 2001.

7 Yuriy Gorodnichenko and Gerard Roland. Understanding the individualism-collectivism cleavage and its effects: Lessons from cultural psychology. *Institutions and Comparative Economic Development*, 150:213, 2012.

8 Graeme Hirst and David St-Onge. Lexical chains as representations of context for the detection and correction of malapropisms. *WordNet: An electronic lexical database*, 305:305–332, 1998.

9 Mario Jarmasz. Roget's thesaurus as a lexical resource for natural language processing. *CoRR*, abs/1204.0140, 2012.

10 José Paulo Leal. Using proximity to compute semantic relatedness in rdf graphs. *Comput. Sci. Inf. Syst.*, 10(4), 2013.

11 Laurent Mazuel and Nicolas Sabouret. Semantic relatedness measure using object properties in an ontology. In *The Semantic Web-ISWC 2008*, pages 681–694. Springer, 2008.

12 Selmar K. Smit and Agoston E. Eiben. Comparing parameter tuning methods for evolutionary algorithms. In *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*, pages 399–406. IEEE, 2009.

13 Michael Strube and Simone Paolo Ponzetto. Wikirelate! computing semantic relatedness using wikipedia. In *AAAI*, volume 6, pages 1419–1424, 2006.

14 Darrell Whitley. A genetic algorithm tutorial. *Statistics and computing*, 4(2):65–85, 1994.