A novel method based on symbolic regression for interpretable semantic similarity measurement

Jorge Martinez-Gil

Software Competence Center Hagenberg GmbH
Softwarepark 21, 4232 Hagenberg, Austria
jorge.martinez-gil@scch.at

Jose Manuel Chaves-Gonzalez

University of Extremadura - Department of Computer Systems Engineering
Avda. de la Universidad s/n Caceres, Spain
jm@unex.es

Abstract

The problem of automatically measuring the degree of semantic similarity between textual expressions is a challenge that consists of calculating the degree of likeness between two text fragments that have none or few features in common according to human judgment. In recent times, several machine learning methods have been able to establish a new state-of-the-art regarding the accuracy, but none or little attention has been paid to their interpretability, i.e. the extent to which an end-user could be able to understand the cause of the output from these approaches. Although such solutions based on symbolic regression already exist in the field of clustering [34], we propose here a new approach which is being able to reach high levels of interpretability without sacrificing accuracy in the context of semantic textual similarity. After a complete empirical evaluation using several benchmark datasets, it is shown that our approach yields promising results in a wide range of scenarios.

Keywords: Knowledge Engineering, Symbolic Regression, Similarity Learning, Semantic Similarity Measurement

1. Introduction

Handling human language can seem a very simple and intuitive activity since people do it every day, but it is a difficult challenge for machines. The reason is that human language is full of phenomena that make comprehension very complex: polysemy, irony, sarcasm, double meanings, paraphrases, etc. But among all these phenomena, there is one that due to its vast amount of applications in modern computer science, stands out from the rest: semantic resemblance or similarity. Being able to automatically calculate the semantic similarity of two textual expressions has applications in an extensive number of fields belonging to computer science, from data integration to automatic translation, through fields such as semantic search, natural language modeling, query expansion, document classification, or question answering.

The fact is that when speaking about human language, and more specifically about written human language or text, we observe that pieces of textual information can be similar in many different ways: they can have a similar structure, discuss similar topics, or even express similar ideas. But if we stick to purely objective criteria, we have that textual information can be similar in just three ways: lexically, syntactically or semantically. Two pieces of textual information are lexically similar if the text that represents the information contains similar sequences of characters. The same pieces can be syntactically similar if they present the same structure. And last but not least, those pieces can be semantically similar if they are referring to the same concepts, regardless of whether they share the same string sequences or structure.

It is widely assumed that assessing the lexical and syntactical similarity over textual expressions are problems that have been already studied in-depth, and they have subsequently been the base for developing a wide range of solutions in the context of information retrieval and text classification. Existing methods are quite accurate and produce good results when it comes to working with a wide variety of use cases. However, these methods are not ideal for measuring semantic similarity since neither the representation nor the structure usually gives clues about the meaning of textual

expressions.

As a consequence, finding semantic similarity between pairs of textual expressions has raised a great deal of research interest for a long time [35]. What makes the problem difficult is the fact that the meaning of a textual expression is not only dependent on the words in it, but also on the way they are perceived by the people using them. To overcome this difficult problem, many methods, resources, and standardization techniques have been proposed (e.g. WordNet, Wikipedia and ontologies such as SUMO, and so on). Even the most recent approaches are based on powerful deep learning models that have been able to achieve good performance [52]. However, to date, researchers have paid very little attention to interpretability, i.e. to the degree to which the output from the resulting models could be understood by a person. In this context, the use of methods based on symbolic regression to improve the interpretability of existing solutions is noteworthy [34]. In this work, we aim to explore the use of such techniques in the field of semantic textual similarity.

As a result, a plethora of methods have addressed the challenge of assessing semantic similarity but without taking into account that the solution must not only be accurate, but capable of being understood by people using them. Therefore, the development of computational solutions whose results can be interpreted by humans poses a fundamental challenge. For this reason, we present here our work to boost interpretability in the field of semantic textual similarity. The major highlights of this work are:

- *Contribution 1:* The proposal of a novel method that relies on symbolic regression to specifically raise higher levels of interpretability when solving semantic similarity problems.
  - *Contribution 1.1.* This proposal is capable of properly handling a trade-off between accuracy and interpretability.
  - *Contribution 1.2.* The results obtained from the proposed approach can be put into production after their automatic exportation to a wide variety of real programming languages: C, C++, Java, Python, etc.

- *Contribution 2:* The evaluation of our method over the benchmark datasets that are tested in several domains, from very general to extremely specific, and its comparison with state-of-the-art methods.
  - *Contribution 2.1.* Our proposal produces very high quality results that are comparable with the results obtained by the best existing methods.
  - *Contribution 2.2.* The results obtained by our proposal are the most interpretable which have been presented to date in the field of semantic similarity.

The rest of this work is organized as follows: Section 2 overviews the state-of-the-art semantic similarity measurement over textual information. Section 3 presents our contribution to boosting interpretability without harming accuracy. Section 4 presents the evaluation of our symbolic regression approach, a comparison with existing methods from the literature, and a thorough analysis of the results achieved. Finally, we remark the major conclusions of this work and possible future lines of research in Section 5.

2. State-of-the-art

The problem of automatically measuring semantic similarity is widely assumed to be a research challenge that tries to address one of the aspects of artificial intelligence that will allow computers to communicate in a natural way with people: the evaluation of resemblance. In fact, given two textual fragments, knowing whether both have a similar meaning or not is vital for facilitating the information exchange. Moreover, it is widely assumed that detecting semantic textual similarity is not considered a final task, but an intermediate process for solving more specific tasks. Obtaining the semantic similarity between two texts not only provides a notion of the resemblance that exists between them, but it also provides information to make decisions following the result obtained.

Therefore, automatically determining the semantic similarity between two textual expressions is an important research challenge that consists of analyzing them to calculate their likeness [43]. However,

the resemblance is not always judged on a binary basis, but it has a whole variety of middle tones, which makes the task especially complex. For this reason, existing semantic similarity methods aims to automatically assess the similarity between textual fragments in a numeric range (e.g. [0, 1], {0, 1, 2, 3, 4}, etc.) where each result represents in an easily understandable way the differences that make two text fragments equivalent or not [30].

In the context of computer science, some methods for semantic similarity have had very important applications for many years in the fields of natural language understanding and text retrieval [39, 42]. Even today, there is quite a large and growing body of academic research based on a number of different approaches: statistical co-occurrence [8], bag-of-words [11], TF-IDFs [3], WordNet [57], PLSI [21], NMF [33] and even the always right, but very expensive manual work done by linguists and domain experts for the creation of synonym lists, taxonomies, hierarchies, knowledge bases or ontologies [54]. More recently, many tasks related to semantic similarity measurement has started relying on using word embeddings [46] as the underlying representation for solving their problem, and this research branch has taken very little time to stand out since the first results obtained had been really good.

With this regard, the community has established a classification on which the different semantic similarity measures rely to calculate the semantic similarity. The classification is based on word similarity, since finding semantic similarity between words is the first and fundamental step for the detection of many kinds of semantic textual similarity in more complex scenarios such as sentences, paragraphs or complete documents. This classification is made up of two differentiated models: the geometric model and the Tversky model.

2.1. The geometric model

The geometric model [14] works with entities in a metric space $< X, \delta >$ where each entity is represented by $x \in X$, and $\delta(a, b)$ is the distance between an entity a and other entity b. This distance is a function over X that returns a non-negative number, so that the less is this number, the greater is the semantic similarity. In a geometric space, for all points $a, b, c \in X$, the three axioms of a metric

are always fulfilled: minimality, symmetry, and triangular inequality:

- $Minimality : \delta(a, b) \geq \delta(a, a) = 0$
- $Symmetry : \delta(a, b) = \delta(b, a)$
- $Triangular\ inequality : \delta(a, b) + \delta(b, c) \geq \delta(a, c)$

2.2. Tversky Model

[60] proposes to use the theory of sets to calculate the similarity of two entities. This similarity model is based on the features of the entities which are being compared. The similarity $sim(a, b)$ is calculated using a function that compares the features that share both a and b and taking into account the features that are only in a and only in b, so that:

$$sim(a, b) = f(a \cap b, a - b, b - a)$$

Thus, those elements that make it possible to distinguish the two entities are the ones that are considered.

Traditionally, the geometric method has been more popular. Although, in recent times, Tversky's method has gained a lot of popularity because it is better suited to learning about similarity. The reason is explained because the underlying problem is that resemblance between two text fragments is often regarded as subjective. For this reason, most existing solutions try to learn to adapt to the tastes of the users who are going to put the systems into operation. The definition of similarity measures and their induction through automatic learning is known as similarity learning, and it consists of looking for a model that improves similarity by means of learning through a set of data (past solved cases or background truth) so that the model learned satisfies similarity constraints and improves outcomes by evaluating certain criteria, and then arrive at a single result of semantic similarity. In the next subsections, we overview the existing methods for semantic similarity aggregation.

2.3. Similarity aggregation methods

There are already several works that explain the use of semantic similarity models from a data analysis perspective. Most of these works, get the results from different semantic similarity measures over the data to be analyzed as an input and then apply a learning process with the purpose of defining a semantic similarity function that improves the results of each of the aforementioned input measures. The most outstanding methods to date are regression analysis, ensemble learning, deep learning, and fuzzy learning.

2.3.1. Regression analysis

Regression analysis is a direct method being able to calculate functional relationships between different semantic similarity measures. The relationship is expressed as a mathematical expression that connects the final result and several input semantic similarity measures [19]. This means that regression analysis tries to predict the final semantic similarity score based on the known values of other semantic similarity measures. In this context, there are different types under which regression analysis linear regression [9], N-Gram regression [38], or Support Vector regression [12]. In general, the results obtained through regression analysis are usually quite easy to interpret by a human, and therefore they are commonly used as interpretable models. However, their accuracy is quite far from the current state-of-art.

2.3.2. Ensemble learning

Concerning ensemble learning, the usual aim is to use multiple learning algorithms to achieve better performance than could be obtained from any of the similarity learning algorithms alone [55]. In contrast to traditional methods that try to learn one hypothesis from training data, ensemble methods try to build a set of hypotheses and smartly combine them. The success of these approaches relies on the idea that the learning mechanism can boost weak methods for semantic similarity measurement which are slightly better than a random guess to build a powerful semantic similarity predictor [52].

2.3.3. Deep learning

Concerning deep learning, the goal is to build artificial neural networks that contain many neuron layers to greatly improve the learning capability of the existing methods [2]. In recent times, there has been rapid progress in natural language processing research, especially on the learning of abstract representations of textual information [28]. In this context, deep learning technology has been successfully applied to solve a number of problems concerning computer vision and computational perception but also, and in a very special way, research in this field has shown the effectiveness of deep neuronal networks, combined with word vector representations, to predict the semantic similarity of words [17], multilingual similarity [7], and short sentences [49]. Moreover, these already solutions can be improved by using linear combinations as shown by Lastra-Díaz et al. [31]. However, and despite the good results, the lack of interpretability has always been a limiting factor in such situations that require proper explanations of the features involved in the model that is put into exploitation.

2.3.4. Fuzzy learning

To date, fuzzy learning has been one of the few attempts to reach high levels of interpretability. In our previous work, we have demonstrated that semantic similarity aggregation can be defined as a problem in which each of the individual results can be satisfactorily aggregated using a fuzzy controller. For instance, CoTO [41] can dramatically improve both accuracy and interpretability if atomic methods might be aggregated, but without reflecting antagonistic suggestions in case of a consensus is possible, or by calculating an optimal trade-off between them in case that such consensus cannot be reached. Or the latest semantic similarity controllers being able to process similarity values in order to calculate an accurate score through a complex, yet human-understandable, fuzzy aggregation approach [44]. The key that guides all research in this branch is based on the idea that good levels of interpretability may be reached if the resulting model can be explained using some kind of rules that are understandable to people.

2.4. The problem of the interpretability

We have seen that perhaps the easiest way to achieve interpretability is to use an aggregation strategy being able to build interpretable models, e.g. regression analysis. However, results achieved with this strategy are far from the state-of-the-art methods. The methods being able to achieve the best results currently rely on huge node networks to get word embeddings. For example, this kind of semantic similarity is often computed by first embedding large text corpora to derive word vectors associated with each word, and then calculating the cosine similarity between them [40]. However, this can be difficult to be understood by a person, since word vectors are designed to be processed by a neural network, not by a person [37].

It is widely assumed that the higher the interpretability of a resulting model, the easier it should be for a person to understand the results achieved. From this, it can be concluded that a resulting model is more interpretable than a different model if the results are easier for a person to understand that results from the different models. In our previous work, we were able to show that fuzzy rule-based models can achieve good interpretability levels without penalizing accuracy. But still, many authors claim that fuzzy rule-based models can just easily be interpreted by fuzzy experts. Many of these authors claim that to equate fuzzy with linguistic and linguistic with interpretable is too simplistic.

For all these reasons, it is worth to look and reflect on the idea that human interpretability has three different approaches: application level, human level, and function level [15]. And it is possible that fuzzy rule-based learning does not satisfy those three interpretability levels. Therefore, complete proof of interpretability would require a successful evaluation with regards to the following three levels.

- *Application level interpretability* is the level at which a domain expert is needed to fully understand how the approach works. Therefore, it requires an appropriate experimental setting and a good understanding of how to measure the quality in that context. Therefore, a proper baseline could consist of determining how good an expert could explain the same decision.

- *Human level intepretability* is a simplified application-level evaluation because it does not require

any expert, but a normal user. This facilitates to perform experiments since it should be easy to find more people to test the approach. A possible example of this level would be to show different explanations to the users, so those users could identify the most suitable one.

- *Function level interpretability* does not involve human intervention. This kind of interpretability works much better when the resulting model used is assumed to be highly interpretable in advance, e.g. it might be assumed that people are able to understand decision trees. In this case, a good baseline to reach this level could be to focus on the size of that decision tree, so that smaller trees are more likely to be associated with a better interpretability score.

Although fuzzy learning can offer good interpretability at the application level, our goal is to go one step further. Therefore, in this work, we pursue to reach high levels of interpretability with regards to the application level, human level, and function level at the same time. The rest of the paper explains our proposal to achieve that goal.

2.5. Novelty of our contribution

Symbolic regression is an application of genetic programming that has the same goal as traditional regression methods. But unlike these traditional methods, it can operate with fewer limitations [26]. Symbolic regression is well-known for being one of the most used methods for function identification and learning. It consists of finding a mathematical expression in a symbolic form describing the mathematical between a dependent variable and one or more independent variables with the highest possible accuracy. The idea is to automatically build resulting models from a set of input values, are able to describe the past solved cases or the background truth [16].

The novelty of our contribution is that, to the best of our knowledge, it is the first time that an approach for boosting interpretability by learning similarity functions through a symbolic regression strategy is proposed. Symbolic regression has already been used for the resolution of specific problems related to function identification. However, our approach is novel with regard to the interpretability

	Regression Analysis	**Ensemble Learning**	**Deep Learning**	**Fuzzy Learning**	**Symbolic Regression**
Knowledge about model structured required	Yes	No	No	No	No
Parametric or Non-Parametric	Parametric	Non-Parametric	Parametric	Parametric	Non-Parametric
Model complexity depends on the amount of samples	No	Yes	No	No	No
Resulting models provide insight into the problem, and increase system understanding	Yes	Hardly	Hardly	Yes	Yes
Complexity control possible	Yes	Yes	Yes	Yes	Yes
Risk of over-fitting the data without complexity control	No	Limited	High	Limited	No
Risk of having insignificant variables in the resulting model	Present	Present	Present	Present	Present
Risk of not having significant variables in the resulting model	Not Present	Present	Present	Present	Present

Table 1: Summary of the main characteristics of the semantic similarity learning methods

in the process of semantic similarity measurement. This is due to the notion of Abstract Syntax Tree (AST), which is a tree representation of the abstract syntactic structure of a function whereby each node of the tree denotes a construct occurring in that function.

The AST provides a mathematically sound model of the meaning behind the semantic similarity aggregation strategy. The final score is obtained by evaluating the nodes' children and then performing the operation stated by the parent on the corresponding child values. In this way, a whole program population is represented by a set of ASTs whereby the inner nodes represent the functions, steps or methods that are used, while the leaves of the symbolic tree represent the semantic similarity measures or the numeric values that are considered as arguments. This allows any mathematical function to make use of several input variables to be identified, correctly represented, and put into exploitation.

Table 1 extends the analysis of [48] to show a summary of the principal characteristics of the different semantic learning methods reviewed. In this table, it is possible to see how the symbolic regression differs from existing methods.

As it can be seen, symbolic regression presents many positive characteristics concerning the other methods, although it has special relevance for our study the possibility that the final models can provide insight into the problem, and increase system understanding. Therefore, in this work, we rely on symbolic regression methods for modeling nonlinear functions on the basis of semantic similarity measures being able to operate in the same way that the rest of models (regression analysis, ensemble learning, deep learning, and fuzzy learning) in order to discover hidden, non-linear relationships in data. Moreover, the great advantage of symbolic regression is the capability of being much better from the interpretability of the results obtained.

There is recent work on the use of symbolic regression to obtain measures of semantic similarity, e.g.[34]. In that paper, the authors introduce an idea similar to ours, since they propose to improve the interpretability of the clustering methods. They use a genetic strategy which automatically selects a small subset of features and then combines them using a variety of operators However, there are some relevant differences with our study. For example, while such work tries to overcome the rigidity of similarity measures such as the Euclidean distance which cannot be easily tailored to the properties of a particular dataset, our research is oriented to serve as an alternative to deep learning techniques based on black box models. This is because [34] apply similarity measures in the geometric space while our research focuses on the study of semantic similarity measures on textual information. This means that although the design of similarity measures has already been explored using symbolic regression, our work is the first attempt to bring such automatic design into the field of semantic textual similarity.

3. Symbolic regression for interpretable semantic similarity measurement

Symbolic regression is a kind of regression analysis that operates in the space of all mathematical expressions to find a resulting model that best fits a given dataset, by considering accuracy and simplicity as major goals. No specific constrains are provided as a starting point to the algorithm [1]. Unlike other types of regression, symbolic regression does not just fit parameters to a pre-existing

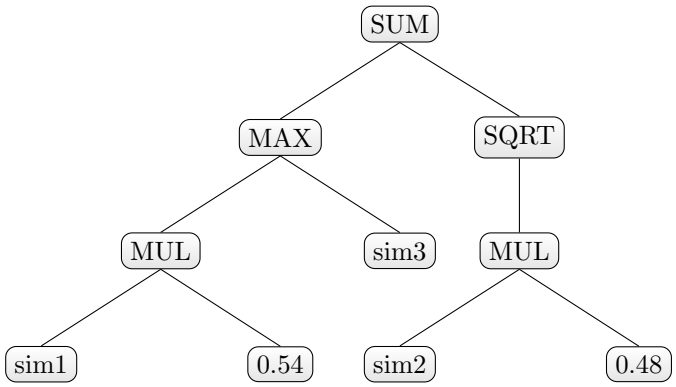

Figure 1: Example of AST for representing the mathematical expression f(sim1, sim2, sim3) = max((sim1*0.54), sim3) + sqrt (sim2 * 0.48))

structure, but it tries to determine the most appropriate structure as well as the optimal parameters for the given input dataset.

On the other hand, genetic programming tries to emulate the general principle of natural selection to solve problems by inducing optimized strategies coded as individuals. In the case of symbolic regression, these evolutionary strategies are directed to automatically generate computer programs, where a program can be a mathematical function, a set of steps (a.k.a. algorithm), or even a set of grammatical rules that given a set of input values, it is able to define a data model that describes their interrelationships. Generally, the program population is represented by ASTs, where the inner nodes of the tree represent the functions, steps or methods that are used, while the outer nodes represent the arguments of the functions. This allows any program, algorithm, or mathematical expression to be represented. Fig. 1 shows an example of AST for the aggregation of three semantic similarity measures (sim1, sim2, and sim3) using several pre-existing operators.

Therefore, the core of this approach consists of using an evolutionary approach for the generation of programs that solve in the best possible way a certain task. In our case, that task is an intelligent aggregation of semantic similarity measures to obtain highly interpretable results. We work here with the Koza approach [27], which means that the programs to do are represented by ASTs through which

it is possible to express complex mathematical functions which are depending on a set of symbols that has to be chosen in advance.

Algorithm 1 Evolutionary strategy to obtain high-quality ASTs

```
1:  procedure CALCULATION OF THE SYMBOLIC EXPRESSION
2:      population ← generationRandomASTs (population)
3:      calculateFitness (population)
4:      while (stop condition not reached) do
5:          parents ← selectionOfIndividuals (population)
6:          offspring ← UXOver (parents)
7:          offspring ← rndMutation (offspring)
8:          calculateFitness (offspring)
9:          population ← updatePopulation (offspring)
10:     endwhile
11:     return optimizedAST (population)
```

Algorithm 1 briefly explains in pseudo-code how the whole process is performed by adapting the original structure of the evolutionary algorithm [22]. The first step consists of the random generation of the chromosome population (i.e. the AST population), and the evaluation of these first solutions (lines 2 and 3). After that, this AST initial population is improved by creating new offspring generations by the application of genetic operators (lines 4 to 10). These operators are the random selection of the parents used to generate a new, and evolved, AST population (line 5); the combination of the information taken from the parents previously selected to create new descendants through the use of uniform crossover (line 6); and the application of a random change (mutation) in a specific part of the solution (line 7). All these changes over the current population produce a new one which is evaluated using a fitness function that will be explained in the next paragraphs (line 8). Then, the individuals of this new population update the solutions of the previous one (line 9). At the end of the process, when the stop condition is reached, the final population will contain the optimized ASTs, which will

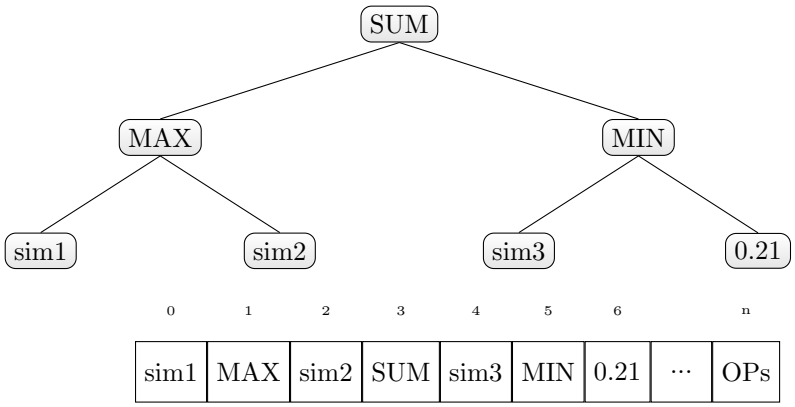

Figure 2: Example of the individual: max (sim1, sim2) + min (sim3, 0.21), the rest of the individual is allocated to identify operator precedence

be returned (line 11).

Figure 2 shows an example of an individual with a length of seven units for the symbolic expression, and the rest of the individual is intended to optimize the operator precedence. We do not work with binary trees in this work since we have developed the evaluation engine and we can decide the number of operands that each function can accept. Besides, the representation of the AST can be done in the array after traversing it in-order.

During the process, the evolutionary strategy is guided towards the goal of maximizing the quality of the results achieved using the different ASTs. This is done by comparing the degree of correlation between the background truth (i.e. the set of past cases solved by the experts in the specific field and which are assumed to be accurate) and the results issued by the AST. In order to be able for the solution to generalize the results, we train and test the model by considering cross-validation when calculating the fitness. In this sense, it is also worth mentioning that we want to keep the size of the ASTs low because it is well known that although larger trees can give very good results on the training, these results are usually due to over-fitting. This means that the validation score which is the most important to assess a solution, would not be all the good we want. At the same time, it is also very important that the training dataset should be well-balanced [61].

As for the fitness function, we have two ways to guide our evolutionary process. The Pearson Correlation Coefficient and the Spearman Rank correlation. The Pearson Correlation Coefficient is calculated between two numerical arrays. It can be formally defined using the following formula, whereby x is the array representing the results from the human expert, and y the array from those cases solved by the best generated AST we have obtained during the evolutionary process:

$$r_{xy} = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\ \sqrt{n\sum y_i^2 - (\sum y_i)^2}}$$

The other alternative is to use the Spearman Rank correlation which is a coefficient that is used to measure the degree of association between the human-generated vector and the machine-generated solution. The Spearman Rank correlation test is the appropriate analysis when the results have to be compared on an ordinal scale. It is calculated as follows:

$$\rho = 1 - \frac{6\sum d_i^2}{n(n^2-1)}$$

Being $d_i = rg(X_i) - rg(Y_i)$ the difference between the two ranks of each array, and n is the size of both arrays.

The major difference between the two ways to measure correlation is that the first is most appropriate in scenarios involving an interval scale, while the second is more suitable for scenarios involving ordinal scales. This means that when we want to produce absolute values of semantic similarity, then we should train the system with Pearson. On the contrary, if the important issue is the relative ordering of the semantic similarity values, then we should train the evolutionary strategy concerning Spearman Rank.

The rest of this section describes in greater detail the AST calculation process. A symbolic regression process for solving a specific problem consists of five main steps: a) Problem Statement, b) Data collection, c) Execution of the symbolic regression program, d) Validation of the model, and e) Use of the resulting model to solve the problem.

3.1. Problem Statement

The initial population in genetic programming consists of creating the individuals of the population, in the form of ASTs, as valid solutions to the problem. An important parameter in the process of initialization of the population is to indicate the maximum size with which the ASTs should be created. It is also important to define the mathematical operators that are going to be allowed, if some real or integer constants are going to be used and the operator precedences. All these parameters have an impact on the search performed through the evolutionary process that will result in the generation of a high-quality solution to a certain problem. Please note that in this case, we work with a supervised type of symbolic regression since the method is expected to produce a certainly known output given certain inputs. This idea is opposed to unsupervised regression in which it is expected a certain behavior of a system but does not explicitly have the precise outputs that must be produced.

3.2. Data collection

The data collections consist of gathering the observations on several elements of the problem to be solved, where each of these elements contains the values for each of the attributes that will be taken into account. Generally, the data are obtained from real environments. In fact, in our specific case, the input data are just the outputs of some existing semantic similarity measurements. Our goal is to add them intelligently thanks to the symbolic regression process. It is important to note that the inputs must all come from methods with high interpretability, such as, for example, those that use dictionaries such as WordNet. Otherwise, maybe the results could be better, but we cannot trace the whole process from beginning to end.

3.3. Execution of the symbolic regression program

The symbolic regression approach is put into execution according to the specific definition of parameters. The values to be defined in this step are those related to the execution time and complexity of the solution. The algorithm starts with an initial population of random ASTs and through an

evolutionary process are assumed to evolve towards the obtaining of better and better ASTs. As is the case with most evolutionary strategies, the modifications made to the population allow information to be mixed from parents to be passed on to descendants or to introduce mutations within the population, in order to find solutions of better quality from generation to generation.

3.4. Validation of the model

The symbolic regression process is considered as an iterative process, in which the outputs generated by the model are used to validate the resolution of the problem. Therefore, it is considered that the model obtained produces a satisfactory output, i.e. reasonably consistent with the data. Then the model obtained is passed to the last phase which consists of the assessment of the score over a dataset that has not been seen before (validation set). Once a model obtains a certain level of validation, it is selected, and the iterative process of symbolic regression is finished. Then this model is used to give a solution to the problem.

3.5. Use of the resulting model to solve the problem

Once the execution of the evolutionary process is over, a valid solution to the problem is taken from the final population generated. Generally, this solution is presented as the best individual (maximum adjustment at the moment of finishing the execution of the program). As the problem-solving process was treated with ASTs, it is convenient to transform this representation into the corresponding form. For example, transform the final AST into a mathematical model in case of working with mathematical functions, or transform the tree into a computer program in case of being managing lines of code.

3.6. The importance of operator precedence

When working with symbolic regression, it is important to note that when two operators have the same operand, the operator having the higher precedence has to go first. For example, in most programming languages, the multiplication operation takes precedence over the addition operation.

In this way, in the expression, $2 + 3 \cdot 5$ the result is 15. Parentheses or brackets can be used to avoid confusion, but they are not mandatory as a general rule. Therefore, optimizing operator precedence has two main effects. On the one hand, it is capable of obtaining better results, since the evolutionary process can guide the solution to obtain the best possible list of priorities. On the other hand, it reduces interpretability and makes it difficult to export directly to programming languages that have already defined in advance the priority of each operator. For this reason, optimizing the precedence of operators allows having a trade-off. If more accuracy is required, we optimize the list. If more interpretability is needed, we do not optimize it. This is line with one of the hypotheses we mentioned at the beginning, and which is that the symbolic regression allows adjusting the trade-off between accuracy and interpretability at the user's convenience. This will be shown much more clearly in the experiments and results analysis we explain in the next section.

3.7. Example

In this subsection, we show how our approach works with a practical example. Let us assume that we have a dataset of semantic similarity whereby the human (or background truth) indicates the degree of similarity between textual expressions. Table 2 shows us the real results showcased by several existing semantic similarity measures. The last column is the result of the transformation of the semantic similarity measures by means of the automatically obtained mathematical formula:

$$MAX(MAX(ssm4 \cdot ssm3, 2 \cdot ssm2 + ssm3), ssm1/ssm4)$$

As we can see: a) the symbolic regression is able to find a mathematical expression being able of appropriately consider each of the semantic similarity measures, and b) unlike other models based on neural networks or ensembles, the resulting expression can be understood by a person.

Human	ssm1	ssm2	ssm3	ssm4	→	**Machine**
1.00	1.00	1.00	1.00	0.49	→	1.00
0.97	1.00	1.00	1.00	1.00	→	1.00
0.97	0.16	0.80	0.69	0.47	→	0.76
0.95	0.23	0.80	0.82	0.60	→	0.80
0.94	0.64	0.80	0.97	0.69	→	0.85
0.92	0.66	0.80	0.97	0.87	→	0.85
0.89	1.00	1.00	1.00	0.75	→	1.00
0.87	1.00	1.00	1.00	0.82	→	1.00
0.79	0.06	0.39	0.23	0.18	→	0.34
0.78	0.08	0.39	0.11	0.05	→	0.54
0.77	0.15	0.80	0.69	0.46	→	0.76
0.75	0.13	0.59	0.65	0.46	→	0.61
0.75	0.54	0.80	0.93	0.46	→	0.84
0.71	0.29	0.80	0.89	0.81	→	0.83
0.42	0.07	0.53	0.27	0.19	→	0.44
0.42	0.08	0.53	0.39	0.26	→	0.48
0.29	0.07	0.23	0.00	0.00	→	0.16
0.28	0.05	0.39	0.23	0.19	→	0.34
0.24	0.04	0.33	0.06	0.05	→	0.30
0.22	0.06	0.23	0.08	0.05	→	0.38
0.22	0.14	0.53	0.68	0.49	→	0.58
0.21	0.05	0.33	0.06	0.05	→	0.30
0.16	0.05	0.48	0.12	0.09	→	0.36
0.14	0.06	0.53	0.24	0.19	→	0.43
0.10	0.05	0.43	0.12	0.09	→	0.33
0.10	0.06	0.53	0.26	0.19	→	0.44
0.03	0.06	0.30	0.28	0.22	→	0.30
0.02	0.05	0.39	0.12	0.09	→	0.30
0.02	0.04	0.13	0.00	0.00	→	0.16
0.02	0.05	0.28	0.00	0.00	→	0.16
Pearson	0.70	0.82	0.82	0.77		0.87

Table 2: Example of semantic similarity aggregation by means of the automatically obtained expression $MAX(MAX(ssm4 \cdot ssm3, 2 \cdot ssm2 + ssm3), ssm1/ssm4)$. Please note that column Machine is normalized

4. Results

Semantic similarity measurement is usually evaluated under several standard methods that allow a fair comparison between different approaches. There are several alternatives for evaluating the quality of novel methods for semantic similarity assessment. These methods are usually based on an intrinsic evaluation that measures the quality of a novel semantic similarity strategy compared to manually tagged information. The comparison between the scores from our symbolic regression strategy and the human judgments are evaluated as the correlation between two arrays of real numbers of the same size, whereby each specific array index indicates the corresponding index within the benchmark datasets.

When working with correlation coefficients, the results that can be obtained will always be in the real interval [-1, 1], being the lower bound of the interval being the least preferred and the upper bound the result we are looking for. Our challenge here is double. On the one hand, to get a result fully interpretable by a non-expert user, which means a milestone in the state-of-the-art. On the other hand, to obtain a final result as close as possible to 1, which means that our approach could properly emulate the judgment of the domain experts. In addition, due to we work with stochastic methods, it is also necessary to perform a significance test to determine the statistical validity of the experiments performed. We estimate here a threshold for the p-value parameter as the usual $5.0 \cdot 10^{-2}$ which means that the correlation coefficient did not come by coincidence. This is not usually a problem because the datasets were already designed beforehand with enough size to make it difficult to obtain strong correlation degrees just by chance.

4.1. Experimental setup

When a novel method for measuring semantic similarity is used to solve instances from benchmark datasets, it is necessary to inform about the best configuration that could be found to be able to face the specific problem. For this reason, the different parameters of both the symbolic regression and the evolutionary processes have been carefully adjusted to obtain more competitive results.

The parameters for the Symbolic Regression are:

- Function set $\{+, -, \cdot, /, exp, max, min\}$
- Size of the individuals [0 - 50]: **15**
- Size of the individual when optimizing operator precedence: **22** (15+7 since 7 different operators are managed)
- Maximum number of constants allowed [0 - 5]: **3** (Moreover the constants are restricted to the real interval [-1, 1])

Concerning the evolutionary process:

- Representation of genes (binary, real): **real**
- Population size [10 - 100]: **25**
- Crossover probability [0.3 - 0.95]: **0.70**
- Mutation probability [0.00 - 0.5]: **0.15**
- Stop condition: Iterate over (1,000 - 100,000): **20,000** generations

Therefore, after having performed a parametric study consisting of experiments from 1,000 to 100,000 iterations, we have realized that the optimal results are obtained on average after 20,000 iterations. Moreover, it is necessary to mention that the results reported in the next subsections are the result of a cross-validation process with one part of the instances for training and other part for the test. This ratio is very common when modeling non-linear functions, and it has been chosen in a wide range of experiments [25]. On the other hand, all the experiments have been performed on a regular PC holding Microsoft Windows 10 64-bit over a processor Intel Core i7-8700 at 3.20Ghz and 32 GB of main memory. As an example, we just wish to mention that the automatic build of a symbolic tree with 22 items took around 55 minutes of processor time in the system described. It is also necessary

to bear in mind that the results shown below are the result of 10 independent executions for each use case, and since our method starts with random numbers, the solution reached will not always be the same.

4.2. General purpose semantic similarity

The first round of experiments is based on a benchmark dataset that is general-purpose oriented. It is called the Miller & Charles [47] dataset. This benchmark dataset is usually used to measure the degree of similarity between terms belonging to a number of general-purpose scenarios, i.e. terms that can be found in numerous situations, so it is formed by a set of words that in principle one could expect to find in almost any file, database or document. It is important to note that this work, we have considered the 30-pair version of the dataset, i.e. while many algorithms cannot work with word pairs that are not present in dictionaries and therefore have to exclude such cases from the evaluation. Our method can be built on the top of any type of algorithm, including those that support the calculation of uncovered words.

The different methods that are going to be considered by our strategy are Jiang & Conrath [24], Leacock & Chodorow [32], Lin [36], and Resnik [54]. These atomic methods are based on different algorithmic strategies over dictionaries such as Wordnet. This makes it possible to achieve a high degree of interpretability [62]. Although it would be possible to use state-of-the-art methods based on word embeddings, for example, whose results would be slightly better, those methods act like black boxes, so it would not be possible to trace the resulting model from beginning to end.

As it is possible to see in Figure 3, the median score is a little higher when the operator precedence is optimized being SyRe our approach for Symbolic Regression and SyReOP the approach for Symbolic Regression with optimization of the Operator Precedence. It is also noteworthy that the results obtained (both maximum and median values) are better than the other existing solutions (CoTO: Consensus or Trade-Off, and FLC: Fuzzy Logic Controller) that also pointed to interpretability as a major goal.

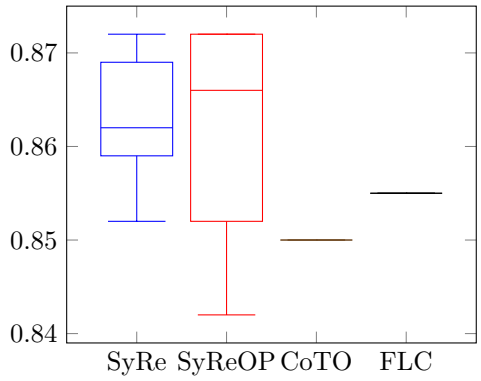

Figure 3: Results of the four interpretable approaches for the calculation of **Pearson Correlation Coefficient** over the Miller & Charles dataset. SyRe = Symbolic Regression, SyReOP = Symbolic Regression with optimization of the Operator Precedence, CoTO = Consensus or Trade-Off, and FLC = Fuzzy Logic Controller

Figure 4 shows the boxplots for the Spearman Rank Correlation, whereby SyReOP was able to achieve a maximum score of 0.931, whereas the median values are 0.810 and 0.928 respectively. It is important to note also the results for the different approaches (e.g. FLC and CoTO) used to solve the benchmark dataset by paying attention to the interpretability of the resulting model which is a key factor of our study.

4.3. Geospatial semantic similarity

This benchmark dataset is strongly related to the geospatial field. In fact, the GeReSiD benchmark dataset [5] addresses a complete set of unique geographic terms that have been grouped and rated in 50 different pairs. Therefore, this geospatial dataset contains words and short textual expressions that an average user could easily find in a number of computational resources, including but not limited to geographic information systems, maps or guides, and travel documents.

Our experiments show that, for the GeReSiD dataset, we have achieved a very similar maximum score for both SyRe and SyReOP (score of 0.670 vs 0.676). Figure 5 shows the results that we have obtained using a boxplot chart as a result of ten independent executions. The atomic methods being used by our symbolic regression approach have been the classical LSA and LSAII [13], and also the

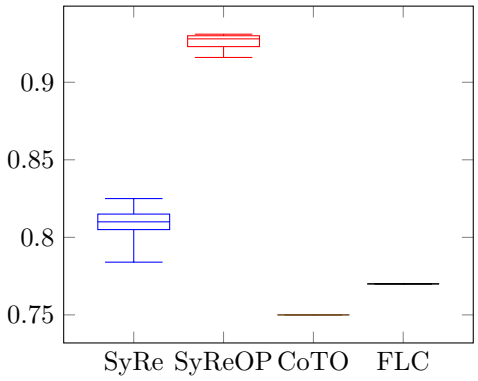

Figure 4: Results of the four interpretable approaches for the calculation of **Spearman Rank Correlation** over the Miller & Charles dataset. SyRe = Symbolic Regression, SyReOP = Symbolic Regression with optimization of the Operator Precedence, CoTO = Consensus or Trade-Off, and FLC = Fuzzy Logic Controller

popular similarity measures UMBC and UMBC-STS [20]. It is important to highlight once again that all of them are also highly interpretable (as they operate over existing textual corpora) and therefore, it is possible to trace the whole process from the beginning to the end so a user might be able to understand the result in a clear way.

The results obtained for the Spearman Rank Correlation is 0.807 for both SyRe and SyReOP (that applies an optimization of the operator precedence). The median values are 0.768 and 0.777 respectively. In Figure 6, we can see a comparison with the other interpretable approaches. Although in this case, it has not been possible to obtain the best results, some of the experiments do indicate that our results are among the best existing methods.

4.4. Biomedical semantic similarity

Our last group of experiments revolves around the MeSH dataset that was proposed in [50]. Evaluating this benchmark dataset is one of the classical experiments for assessing the semantic similarity of biomedical vocabularies. In fact, this dataset, which is formed by 36 use cases, collects domain-specific terms that belong to the field of medicine and biomedicine. Therefore, it is useful to measure the effectiveness of recently proposed methods when working with scientific and medical

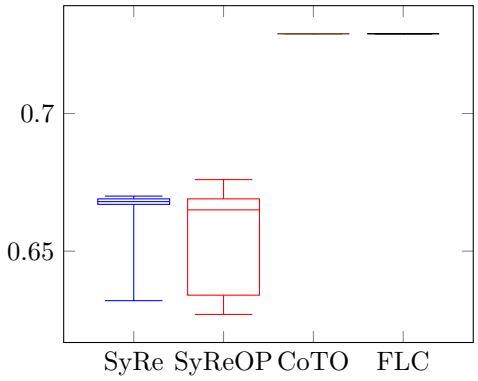

Figure 5: Results of the 4 interpretable approaches for the calculation of **Pearson Correlation Coefficient** over the GeReSiD dataset. SyRe = Symbolic Regression, SyReOP = Symbolic Regression with optimization of the Operator Precedence, CoTO = Consensus or Trade-Off, and FLC = Fuzzy Logic Controller

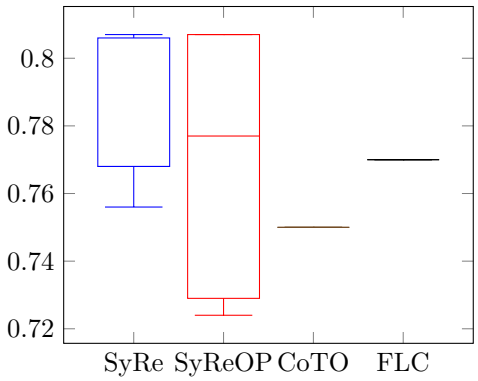

Figure 6: Results of the four interpretable approaches for the calculation of **Spearman Rank Correlation** over the GeReSiD dataset. SyRe = Symbolic Regression, SyReOP = Symbolic Regression with optimization of the Operator Precedence, CoTO = Consensus or Trade-Off, and FLC = Fuzzy Logic Controller

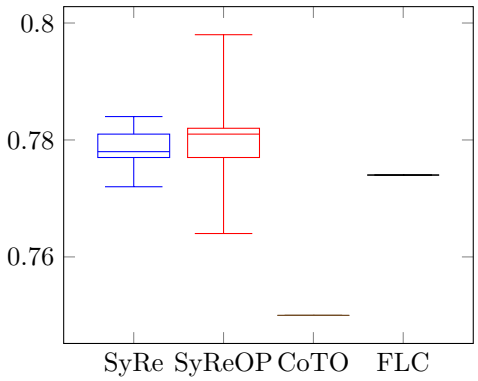

Figure 7: Results of the four interpretable approaches for the calculation of the **Pearson Correlation Coefficient** over the MeSH dataset. SyRe = Symbolic Regression, SyReOP = Symbolic Regression with optimization of the Operator Precedence, CoTO = Consensus or Trade-Off, and FLC = Fuzzy Logic Controller

literature, biological databases, clinical histories of patients who receive some treatment, and so on.

After performing an empirical evaluation once again, we conclude that we have been able to achieve a maximum Pearson Correlation coefficient of 0.784 and 0.798 for SyRe and SyReOP respectively. Although the median values are very similar (0.778 and 0.781 respectively). Figure 7 shows how the results for each sample are distributed using boxplots. The atomic methods for being aggregated in an optimal AST are Adapted Lesk [6], Leacock [32], Path-based [53], and Resnik [54], which are highly interpretable.

And concerning the Spearman Rank Correlation, Figure 8 shows that the median values are 0.885 and 0.923 for SyRe and SyReOp respectively, with scores of 0.928 and 0.932 for the maximum values. In all cases, it is also possible to see that using symbolic regression is able to outperform the existing interpretable fuzzy models.

To better understand how the fitness of each solution evolves, we offer a study about the computational costs of evaluating the solutions that have the objective of interpretability. In this study, we detail the behavior of the four solutions in relation to time. Although it is true that, in really minimal cases, it can take up to one hour to complete the calculation of the symbolic regression function, most

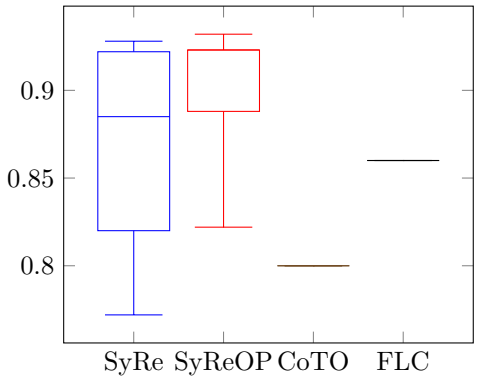

Figure 8: Results of the four interpretable approaches for the calculation of **Spearman Rank Correlation** over the MeSH dataset. SyRe = Symbolic Regression, SyReOP = Symbolic Regression with optimization of the Operator Precedence, CoTO = Consensus or Trade-Off, and FLC = Fuzzy Logic Controller

of the times a reasonably good result is obtained in the first few minutes.

Stagnation of our proposal, and consequently important time penalizations, is avoided because once the evolutionary strategy reaches what seems to be a maximum, the search space is explored in different regions through the use of mutations performed to the current solutions.

Figure 9 shows the behavior of the different solutions for the first minutes of the calculation of the symbolic regression function over the Miller and Charles dataset. We show the evolution of the test results over time. After that, the function usually stabilizes with small improvements through mutations. As you can see, methods based on symbolic regression tend to converge faster for both Pearson and Spearman. This is because functions based on fuzzy logic need to learn much more complex fuzzy models.

Figure 10 shows again the evolution of the test results for both Pearson and Spearman over time, but this time on the GeReSiD dataset. Again, we can see that the methods based on symbolic regression are lighter and therefore they can learn better solutions in less time.

Finally, Figure 11 shows again the evolution of the test results over time. But this time on the MeSH dataset. The plot on the left represents the results for the Pearson Correlation Coefficient and

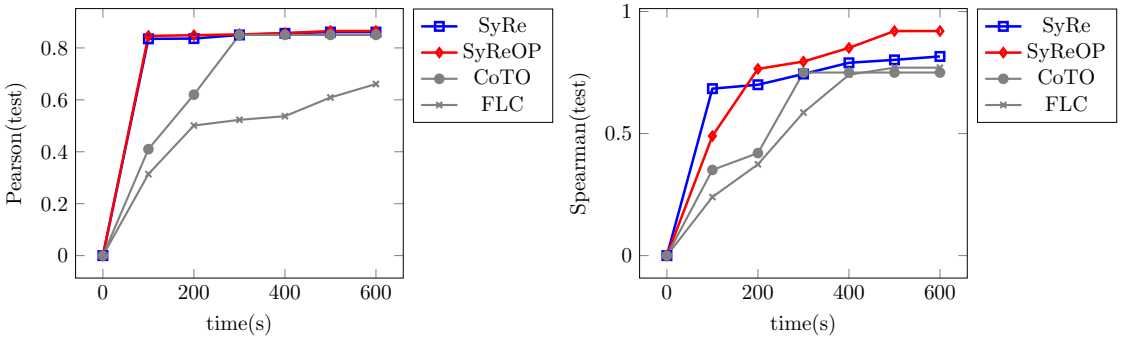

Figure 9: Fitness evolution for the four interpretable approaches over the **Miller and Charles dataset**. Methods based on symbolic regression tend to converge more quickly since the model to be learned is simpler and more interpretable.

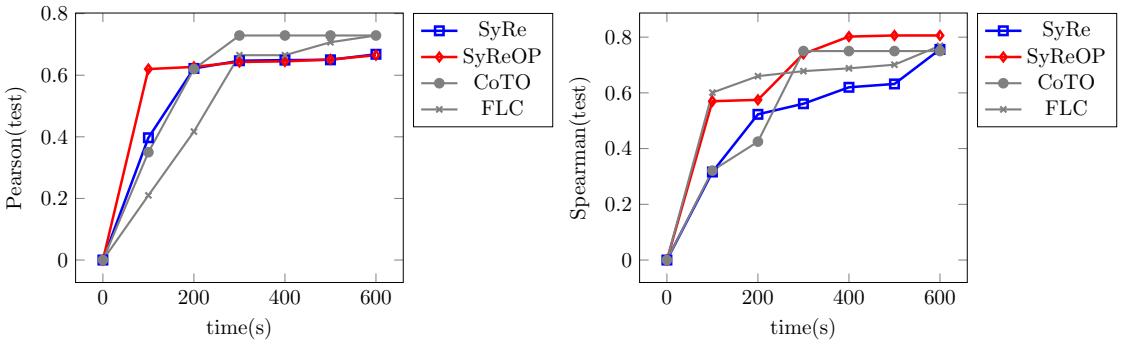

Figure 10: Fitness evolution for the four interpretable approaches over the **GeReSiD dataset**. Once again, the methods based on symbolic regression are able to find better solutions in less time due to the simplicity of the model

the plot on the right represents the results for the Spearman Rank Correlation.

4.5. Comparison With Existing Works

It seems reasonable to think that accuracy and interpretability are antagonistic goals. In fact, it could be thought that reaching high levels of interpretability at the expense of poor accuracy does not seem to have a real practical impact. And the truth is that achieving good results in terms of interpretability is not useful if they are not accompanied by good results in terms of accuracy. For that reason, we analyze here the performance of our proposal in relation to some of the most outstanding semantic similarity measures.

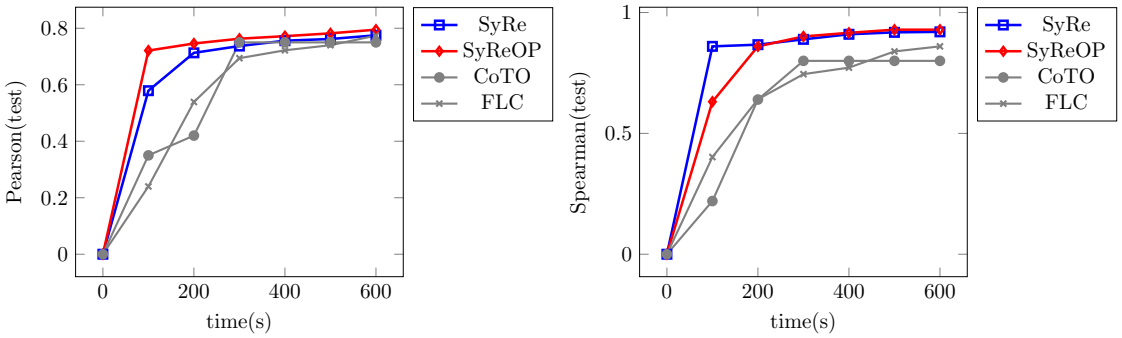

Figure 11: Fitness evolution for the four interpretable approaches on the **MeSH dataset**. Once again, it can be seen that quite good semantic similarity measures can be obtained in a matter of several minutes

The first comparison is related to the measurement of semantic similarity in general-purpose settings. To date, the most methods are those calculated by creating a vectorized representation of the words (a. k. a. word embeddings) by means of deep neuronal networks trained with huge text collections [7, 17, 46]. Table 3 shows the results of different approaches when solving the Miller & Charles dataset using the Pearson Correlation Coefficient. As it is possible to see, the different variants of our proposal gets the best results in this case. Although we recommend taking into account the median results since the maximum result only occurs one out of every ten times.

Table 4 shows the results of different approaches when solving the Miller & Charles benchmark dataset using the Spearman Rank Correlation. As can be seen, our approaches (in particular the one being able to optimize the operator precedence) are able to achieve the best results in this particular case. Again, we want to note that the maximum result is only obtained in one out of every ten times.

Table 5 shows the results achieved by a number of existing methods when solving the GeReSiD dataset with the goal of achieving a good Pearson Correlation Coefficient. The problem with GeReSiD is that it is still a very recent and domain-specific dataset, and that means that there are just a few works that cover it. As we can see, although our approach is not capable of leading the ranking, at least it is able to overcome a good amount of existing methods.

Table 6 shows the results of a number of existing methods when solving the GeReSiD dataset using

Algorithm	Score	p-value
Google distance [10]	0.470	$8.8 \cdot 10^{-3}$
Huang et al. [23]	0.659	$7.5 \cdot 10^{-5}$
Jiang & Conrath [24]	0.669	$5.3 \cdot 10^{-5}$
Resnik [54]	0.780	$1.9 \cdot 10^{-7}$
Leacock & Chodorow [32]	0.807	$4.0 \cdot 10^{-8}$
Lin [36]	0.810	$3.0 \cdot 10^{-8}$
Faruqui & Dyer [17]	0.817	$2.0 \cdot 10^{-8}$
Mikolov et al. [46]	0.820	$2.2 \cdot 10^{-8}$
CoTO [41]	0.850	$1.0 \cdot 10^{-8}$
FLC [44]	0.855	$1.0 \cdot 10^{-8}$
Our approach (median)	0.862	$9.4 \cdot 10^{-9}$
Our approach + OP (median)	0.866	$6.4 \cdot 10^{-9}$
Our approach (maximum)	0.872	$3.5 \cdot 10^{-9}$
Our approach + OP (maximum)	0.872	$3.5 \cdot 10^{-9}$

Table 3: Results for the **Pearson Correlation Coefficient** achieved by existing methods over the **Miller & Charles** dataset

Algorithm	Score	p-value
Jiang & Conrath [24]	0.588	$4.0 \cdot 10^{-4}$
Lin [36]	0.619	$1.6 \cdot 10^{-4}$
Aouicha et al. [4]	0.640	$8.0 \cdot 10^{-5}$
Resnik [53]	0.757	$5.3 \cdot 10^{-7}$
Mikolov et al. [46]	0.770	$2.6 \cdot 10^{-7}$
Leacock & Chodorow [32]	0.789	$8.1 \cdot 10^{-8}$
Our approach (median)	0.810	$1.9 \cdot 10^{-8}$
Our approach (maximum)	0.825	$6.3 \cdot 10^{-9}$
Bojanowski et al. [7]	0.846	$1.1 \cdot 10^{-9}$
FLC [44]	0.891	$8.3 \cdot 10^{-12}$
Our approach + OP (median)	0.928	$2.1 \cdot 10^{-14}$
Our approach + OP (maximum)	0.931	$1.1 \cdot 10^{-14}$

Table 4: Results for the **Spearman Rank Correlation** achieved by existing methods over the **Miller & Charles** dataset

Algorithm	Score	p-value
Han et al. (UMBC) [20]	0.490	$3.0 \cdot 10^{-4}$
Deerwester et al. (LSA) [13]	0.540	$5.2 \cdot 10^{-5}$
Deerwester et al. (LSAII) [13]	0.594	$3.5 \cdot 10^{-7}$
Han et al. (UMBC-STS) [20]	0.630	$4.7 \cdot 10^{-7}$
Aouicha et al. [4]	0.640	$4.7 \cdot 10^{-7}$
Our approach + OP (median)	0.665	$2.0 \cdot 10^{-7}$
Our approach (median)	0.668	$1.1 \cdot 10^{-7}$
Our approach (maximum)	0.671	$9.7 \cdot 10^{-8}$
Our approach + OP (maximum)	0.676	$7.2 \cdot 10^{-8}$
FLC [44]	0.729	$1.9 \cdot 10^{-9}$

Table 5: Results for the **Pearson Correlation Coefficient** achieved by the different methods over the **GeReSiD** dataset

Algorithm	Score	p-value
Resnik [54]	0.260	$6.3 \cdot 10^{-2}$
J&C [24]	0.310	$2.5 \cdot 10^{-2}$
Lin [36]	0.390	$4.3 \cdot 10^{-3}$
Gabrilovich & Markovitch (ESA) [18]	0.680	$2.9 \cdot 10^{-8}$
Deerwester et al. (LSA) [13]	0.710	$3.8 \cdot 10^{-9}$
FLC [44]	0.749	$1.7 \cdot 10^{-10}$
Our approach (median)	0.768	$3.0 \cdot 10^{-11}$
Our approach + OP (median)	0.777	$1.3 \cdot 10^{-11}$
Our approach (maximum)	0.806	$5.6 \cdot 10^{-13}$
Our approach + OP (maximum)	0.807	$5.0 \cdot 10^{-13}$

Table 6: Results for the **Spearman Rank Correlation** achieved by the different methods over the **GeReSiD** dataset

the Spearman Rank Correlation. As can be seen, our approaches did perform as well as the fuzzy controllers and the classic LSA, and it is still capable of beating the most classic algorithms.

Table 7 shows the results achieved by already existing methods when solving the MeSH dataset. The biomedical domain has a long tradition in the field of biomedical similarity measures which makes there a good number of semantic similarity methods in this context that work fairly well. However, we can see that our symbolic regression approach is able to perform better in this particular scenario with the median values having the greatest chance of occurring in a production environment.

Finally, Table 8 shows the results of already published methods when solving MeSH dataset. Once again, our approach with the capability to optimize the operator precedence has achieved the best

Algorithm	Score	p-value
Adapted Lesk [6]	0.584	$9.2 \cdot 10^{-4}$
Path-based [53]	0.584	$9.2 \cdot 10^{-4}$
Li et al. [35]	0.707	$7.2 \cdot 10^{-7}$
J&C [24]	0.718	$4.1 \cdot 10^{-7}$
Lin [36]	0.718	$4.1 \cdot 10^{-7}$
Resnik [54]	0.721	$4.0 \cdot 10^{-7}$
Meng et al. [45]	0.731	$2.1 \cdot 10^{-7}$
Seco et al. [57]	0.732	$2.1 \cdot 10^{-7}$
Sanchez et al. [56]	0.735	$1.8 \cdot 10^{-7}$
Taieb et al. [58]	0.753	$6.0 \cdot 10^{-8}$
FLC [44]	0.774	$1.2 \cdot 10^{-8}$
Our approach (median)	0.778	$3.0 \cdot 10^{-8}$
Our approach + OP (median)	0.781	$1.9 \cdot 10^{-8}$
Our approach (maximum)	0.784	$1.6 \cdot 10^{-8}$
Our approach + OP (maximum)	0.798	$5.5 \cdot 10^{-9}$

Table 7: Results for the **Pearson Correlation Coefficient** achieved by the different methods over the **MeSH** dataset

results in this case.

From the experiments, it is possible to deduce that despite our approach aims to achieve resulting models which are highly interpretable, we have also been able to achieve results that are comparable with the results from some of the best methods proposed until date. Besides, it is necessary to remark that all the semantic similarity measures that have been chosen as inputs are classical methods, that although they do not currently yield the best results when operating separately, are easy for users to trace and understand. This makes the whole process highly interpretable. Therefore, we could always add other inputs based on the use of deep learning approaches that provide very good scores as a way to improve the results presented. However, this improvement would be achieved at the expense of interpretability. In contrast, the solution that we have presented here has the advantage that it makes it possible to trace the whole process from the beginning to the end. Moreover, if the operator precedence is not optimized, the results are easily exportable to any type of system to be immediately put into operation.

Algorithm	Score	p-value
Seco et al. [57]	0.624	$2.8 \cdot 10^{-5}$
Li et al. [35]	0.707	$7.0 \cdot 10^{-7}$
J&C [24]	0.710	$6.0 \cdot 10^{-7}$
Lin [36]	0.718	$4.0 \cdot 10^{-7}$
Resnik [54]	0.721	$3.3 \cdot 10^{-7}$
FaITH [51]	0.724	$2.8 \cdot 10^{-7}$
FLC [44]	0.859	$5.2 \cdot 10^{-12}$
Our approach (median)	0.885	$1.7 \cdot 10^{-13}$
Our approach + OP (median)	0.923	$1.7 \cdot 10^{-16}$
Our approach (maximum)	0.928	$5.2 \cdot 10^{-17}$
Our approach + OP (maximum)	0.932	$1.9 \cdot 10^{-17}$

Table 8: Results for the **Spearman Rank Correlation** achieved by the different methods over the **MeSH** dataset

4.6. Lessons Learned

Several lessons can be drawn from this work that can be useful for the research community that seeks to design novel semantic similarity aggregation methods that meet the needs of users in terms of accuracy but also interpretability. The most remarkable lessons that can be learned from our work are as follows:

- *Lesson 1.* Our proposal represents a novel method that relies on symbolic regression to specifically raise higher levels of interpretability when solving semantic similarity problems. Thus, this proposal is not only capable of obtaining good results concerning semantic similarity values, but it is also able to properly handling a trade-off between accuracy and interpretability, in such a way that as required one of the two can be increased at the expense of the other.

- *Lesson 2.* It is possible to improve the accuracy of the results obtained by symbolic regression by doing one of these three things: a) allowing larger ASTs, b) allowing state-of-the-art semantic similarity functions to be used as inputs even if they are based on black-box models, and c) optimizing operator precedence. However, applying each of these options makes the solution less interpretable by a human operator. It is up to the operator to decide what is most convenient at any given situation.

- *Lesson 3.* If the operator precedence is not optimized during the process, the resulting model can easily be exported to a wide range of programming languages: C, C++, Java, Python, etc... This is because the automatic conversion of the function represented by an AST into source code is trivial.

- *Lesson 4.* Our proposal gives results that are in line with the best methods available today. Therefore, in some cases, the maximum obtained by our symbolic regression method is the highest value in the literature for semantic similarity aggregation techniques, and the median values obtained are also among the best values studied. However, it is necessary to mention that the comparison between an aggregation method and an atomic method is usually not entirely appropriate.

- *Lesson 5.* The results provided in this study are the most interpretable which have been presented to date because symbolic regression strictly complies with most of the best practices regarding interpretability. And this is possible because the method of symbolic regression adapts very well to the three levels of interpretability considered in this work: application level (where only experts can understand the model), human level (where any end-user can understand the model) and functional level (which by definition is the level at which a model is described by symbolic regression).

Despite the lessons learned, there are still some open questions that will have to be faced as part of future work. The most relevant are those related to the use of background knowledge and to the large consumption of computational resources required to put this type of method into exploitation. Concerning the first, a step further beyond could be to improve the current method by using background knowledge in the sense of putting together the strengths of genetic programming with some kind of background knowledge that could be injected into the evolutionary process. However, one first limitation is that genetic programming does not offer an easy way to incorporate such domain knowledge besides the set of operators. Moreover, it is necessary to keep in mind that our method

based on symbolic regression results in a semantic similarity measure that reflects some of the particular characteristics of the training set. Therefore, it is not reasonable to think that the mathematical function obtained can be applied to broader scenarios or cases involving datasets of a different nature.

Respect to the second limitation, it is widely known that the use of genetic programming requires huge amounts of computational resources, even when the search space is properly limited. This is usually called the bloat phenomenon in the literature [29], and it usually affects genetic programming in two ways; on the one hand, it might cause the early termination of an evolutionary process due to the exhaustion of the available memory; on the other hand, it could increase the fitness computation cost. However, these issues will be treated as future work.

Finally, it could be an interesting idea to define a measure of comprehensibility and then compare other methods to the novel methods concerning that comprehensibility measure. It is possible to find several examples of formal comprehensibility measures in the idea mining community since ideas are identified in texts with mining methods that are comprehensible to the user [59].

5. Conclusions

In this work, we have introduced our research concerning the design of aggregated semantic similarity measures that are not only capable of achieving good performance but that are also understandable by human operators. To do this, we have worked in the framework of symbolic regression, which allows calculating the mathematical function that best fits a given set of input data derived from existing and interpretable semantic similarity measures. This is possible because the symbolic regression allows learning and optimizing the calculation of the best possible AST through an evolutionary learning process. This process can ensure that the resulting model reveals the data's underlying structure in a way that can be understandable from a human perspective.

We have shown that deep learning models can achieve high accuracy but at the expense of high abstraction. However, working under genetic programming, symbolic regression is responsible for

evolving populations of individuals, which represent equations, functions, or mathematical models and have been used notably to improve the interpretability of clustering solutions [34] in order to improve the similarity measures which cannot be easily tailored to the particular properties of a given dataset. In this work, we use a similar approach but oriented to the systematic aggregation of semantic similarity measures. The results obtained are quite promising since it is up to the human end-user to decide what kind of configuration can better fit their needs in terms of accuracy and simplicity. Besides, it is trivial to put the resulting models into production after their automatic exportation to a wide variety of real programming languages: C, C++, Java, Python, etc.

As it can be noted throughout this work, the question that plans in the background is that the election of a model is not only a question of gaining some precision through the design of models increasingly complex to understand. It is also important to assume that aspects such as the interpretability of the model are as important as accuracy in order novel approaches for semantic similarity might be adopted by people.

Competing interest

Authors have no competing interest to declare.

Acknowledgments

We would like to thank the anonymous reviewers for their help towards improving this work. This research work has been partially supported by the Austrian Ministry for Transport, Innovation and Technology, the Federal Ministry of Science, Research and Economy, and the Province of Upper Austria in the frame of the COMET center SCCH. It was also supported by 4IE+ project (0499_4IE_PLUS_4_E) funded by the Interreg V-A Spain-Portugal (POCTEP) 2014-2020 program and by RTI 2018-094591-B-I00 (MCIU/AEI/FEDER, UE) project.

References

[1] Affenzeller, M., Winkler, S. M., Kronberger, G., Kommenda, M., Burlacu, B., & Wagner, S. (2013). Gaining deeper insights in symbolic regression. In *Genetic Programming Theory and Practice XI [GPTP 2013, University of Michigan, Ann Arbor, USA, May 9-11, 2013].* (pp. 175–190). URL: https://doi.org/10.1007/978-1-4939-0375-7_10. doi:10.1007/978-1-4939-0375-7_10.

[2] Afzal, N., Wang, Y., & Liu, H. (2016). Mayonlp at semeval-2016 task 1: Semantic textual similarity based on lexical semantic net and deep learning semantic model. In *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016* (pp. 674–679). URL: https://www.aclweb.org/anthology/S16-1103/.

[3] Albitar, S., Fournier, S., & Espinasse, B. (2014). An effective tf/idf-based text-to-text semantic similarity measure for text classification. In *Web Information Systems Engineering - WISE 2014 - 15th International Conference, Thessaloniki, Greece, October 12-14, 2014, Proceedings, Part I* (pp. 105–114). URL: https://doi.org/10.1007/978-3-319-11749-2_8. doi:10.1007/978-3-319-11749-2_8.

[4] Aouicha, M. B., Taieb, M. A. H., & Hamadou, A. B. (2016). LWCR: multi-layered wikipedia representation for computing word relatedness. *Neurocomputing*, *216*, 816–843. URL: https://doi.org/10.1016/j.neucom.2016.08.045. doi:10.1016/j.neucom.2016.08.045.

[5] Ballatore, A., Bertolotto, M., & Wilson, D. C. (2014). An evaluative baseline for geo-semantic relatedness and similarity. *GeoInformatica*, *18*, 747–767. doi:10.1007/s10707-013-0197-8.

[6] Banerjee, S., & Pedersen, T. (2002). An adapted lesk algorithm for word sense disambiguation using wordnet. In *Computational Linguistics and Intelligent Text Processing, Third International*

Conference, CICLing 2002, Mexico City, Mexico, February 17-23, 2002, Proceedings (pp. 136–145). URL: https://doi.org/10.1007/3-540-45715-1_11. doi:10.1007/3-540-45715-1_11.

[7] Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *TACL*, *5*, 135–146.

[8] Bollegala, D., Matsuo, Y., & Ishizuka, M. (2011). A web search engine-based approach to measure semantic similarity between words. *IEEE Trans. Knowl. Data Eng.*, *23*, 977–990. doi:10.1109/TKDE.2010.172.

[9] Chaves-González, J. M., & Martinez-Gil, J. (2013). Evolutionary algorithm based on different semantic similarity functions for synonym recognition in the biomedical domain. *Knowl.-Based Syst.*, *37*, 62–69. doi:10.1016/j.knosys.2012.07.005.

[10] Cilibrasi, R., & Vitányi, P. M. B. (2007). The google similarity distance. *IEEE Trans. Knowl. Data Eng.*, *19*, 370–383. URL: https://doi.org/10.1109/TKDE.2007.48. doi:10.1109/TKDE.2007.48.

[11] Clinchant, S., & Perronnin, F. (2013). Textual similarity with a bag-of-embedded-words model. In *International Conference on the Theory of Information Retrieval, ICTIR '13, Copenhagen, Denmark, September 29 - October 02, 2013* (p. 25). URL: https://doi.org/10.1145/2499178.2499180. doi:10.1145/2499178.2499180.

[12] Croce, D., Annesi, P., Storch, V., & Basili, R. (2012). UNITOR: combining semantic text similarity functions through SV regression. In *Proceedings of the 6th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2012, Montréal, Canada, June 7-8, 2012* (pp. 597–602). URL: https://www.aclweb.org/anthology/S12-1088/.

[13] Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., & Harshman, R. A. (1990). Indexing by latent semantic analysis. *JASIS*, *41*, 391–407.

[14] Deza, M. M., & Deza, E. (2009). Encyclopedia of distances. In *Encyclopedia of distances* (pp. 1–583). Springer.

[15] Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, .

[16] Fagundes, R. A. A., de Souza, R. M. C. R., & de A. Cysneiros, F. J. (2013). Robust regression with application to symbolic interval data. *Eng. Appl. of AI*, *26*, 564–573. URL: https://doi.org/10.1016/j.engappai.2012.05.004. doi:10.1016/j.engappai.2012.05.004.

[17] Faruqui, M., & Dyer, C. (2014). Improving vector space word representations using multilingual correlation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014, April 26-30, 2014, Gothenburg, Sweden* (pp. 462–471).

[18] Gabrilovich, E., & Markovitch, S. (2009). Wikipedia-based semantic interpretation for natural language processing. *J. Artif. Intell. Res.*, *34*, 443–498. URL: https://doi.org/10.1613/jair.2669. doi:10.1613/jair.2669.

[19] Greiner, P., Proisl, T., Evert, S., & Kabashi, B. (2013). KLUE-CORE: A regression model of semantic textual similarity. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics, *SEM 2013, June 13-14, 2013, Atlanta, Georgia, USA.* (pp. 181–186). URL: https://www.aclweb.org/anthology/S13-1026/.

[20] Han, L., Finin, T., McNamee, P., Joshi, A., & Yesha, Y. (2013). Improving word similarity by augmenting PMI with estimates of word polysemy. *IEEE Trans. Knowl. Data Eng.*, *25*, 1307–1322. doi:10.1109/TKDE.2012.30.

[21] Hofmann, T. (1999). Probabilistic latent semantic indexing. In F. C. Gey, M. A. Hearst, & R. M. Tong (Eds.), *SIGIR '99: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 15-19, 1999, Berkeley, CA, USA*

(pp. 50–57). ACM. URL: https://doi.org/10.1145/312624.312649. doi:10.1145/312624.312649.

[22] Holland, J. H., & Reitman, J. S. (1977). Cognitive systems based on adaptive algorithms. *SIGART Newsletter*, *63*, 49. doi:10.1145/1045343.1045373.

[23] Huang, E. H., Socher, R., Manning, C. D., & Ng, A. Y. (2012). Improving word representations via global context and multiple word prototypes. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 1: Long Papers* (pp. 873–882).

[24] Jiang, J. J., & Conrath, D. W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the 10th Research on Computational Linguistics International Conference, ROCLING 1997, Taipei, Taiwan, August 1997* (pp. 19–33).

[25] Kedem, D., Tyree, S., Weinberger, K. Q., Sha, F., & Lanckriet, G. R. G. (2012). Non-linear metric learning. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.* (pp. 2582–2590).

[26] Kommenda, M., Affenzeller, M., Burlacu, B., Kronberger, G., & Winkler, S. M. (2014). Genetic programming with data migration for symbolic regression. In *Genetic and Evolutionary Computation Conference, GECCO '14, Vancouver, BC, Canada, July 12-16, 2014, Companion Material Proceedings* (pp. 1361–1366). URL: https://doi.org/10.1145/2598394.2609857. doi:10.1145/2598394.2609857.

[27] Koza, J. R. (1992). *Genetic programming: on the programming of computers by means of natural selection* volume 1. MIT press.

[28] Lan, W., & Xu, W. (2018). Neural network models for paraphrase identification, semantic textual similarity, natural language inference, and question answering. In *Proceedings of the 27th*

International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018 (pp. 3890–3902).

[29] Langdon, W. B. (2000). Quadratic bloat in genetic programming. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '00), Las Vegas, Nevada, USA, July 8-12, 2000* (pp. 451–458).

[30] Lastra-Díaz, J. J., García-Serrano, A., Batet, M., Fernández, M., & Chirigati, F. (2017). HESML: A scalable ontology-based semantic similarity measures library with a set of reproducible experiments and a replication dataset. *Inf. Syst.*, *66*, 97–118. URL: `https://doi.org/10.1016/j.is.2017.02.002`. doi:`10.1016/j.is.2017.02.002`.

[31] Lastra-Díaz, J. J., Goikoetxea, J., Taieb, M. A. H., García-Serrano, A., Aouicha, M. B., & Agirre, E. (2019). A reproducible survey on word embeddings and ontology-based methods for word similarity: Linear combinations outperform the state of the art. *Eng. Appl. of AI*, *85*, 645–665. URL: `https://doi.org/10.1016/j.engappai.2019.07.010`. doi:`10.1016/j.engappai.2019.07.010`.

[32] Leacock, C., & Chodorow, M. (1998). Combining local context and wordnet similarity for word sense identification. *WordNet: An electronic lexical database*, *49*, 265–283.

[33] Lee, D. D., & Seung, S. H. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, *401*, 788–791.

[34] Lensen, A., Xue, B., & Zhang, M. (2019). Genetic programming for evolving similarity functions for clustering: Representations and analysis. *Evolutionary computation*, (pp. 1–29).

[35] Li, Y., Bandar, Z., & McLean, D. (2003). An approach for measuring semantic similarity between words using multiple information sources. *IEEE Trans. Knowl. Data Eng.*, *15*, 871–882. doi:`10.1109/TKDE.2003.1209005`.

[36] Lin, D. (1998). An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning (ICML 1998), Madison, Wisconsin, USA, July 24-27, 1998* (pp. 296–304).

[37] Lipton, Z. C. (2018). The mythos of model interpretability. *Commun. ACM*, *61*, 36–43. URL: https://doi.org/10.1145/3233231. doi:10.1145/3233231.

[38] Malandrakis, N., Iosif, E., & Potamianos, A. (2012). Deeppurple: Estimating sentence semantic similarity using n-gram regression models and web snippets. In *Proceedings of the 6th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2012, Montréal, Canada, June 7-8, 2012* (pp. 565–570). URL: https://www.aclweb.org/anthology/S12-1082/.

[39] Martinez-Gil, J. (2014). An overview of textual semantic similarity measures based on web intelligence. *Artif. Intell. Rev.*, *42*, 935–943. doi:10.1007/s10462-012-9349-8.

[40] Martinez-Gil, J. (2015). Automated knowledge base management: A survey. *Comput. Sci. Rev.*, *18*, 1–9. URL: https://doi.org/10.1016/j.cosrev.2015.09.001. doi:10.1016/J.COSREV.2015.09.001.

[41] Martinez-Gil, J. (2016). Coto: A novel approach for fuzzy aggregation of semantic similarity measures. *Cognitive Systems Research*, *40*, 8–17. doi:10.1016/j.cogsys.2016.01.001.

[42] Martinez-Gil, J. (2019). Semantic similarity aggregators for very short textual expressions: a case study on landmarks and points of interest. *J. Intell. Inf. Syst.*, *53*, 361–380. doi:10.1007/s10844-019-00561-0.

[43] Martinez-Gil, J., & Aldana-Montes, J. F. (2013). Semantic similarity measurement using historical google search patterns. *Inf. Syst. Frontiers*, *15*, 399–410. URL: https://doi.org/10.1007/s10796-012-9404-7. doi:10.1007/S10796-012-9404-7.

[44] Martinez-Gil, J., & Chaves-González, J. M. (2019). Automatic design of semantic similarity

controllers based on fuzzy logics. *Expert Syst. Appl.*, *131*, 45–59. URL: https://doi.org/10.1016/j.eswa.2019.04.046. doi:10.1016/j.eswa.2019.04.046.

[45] Meng, L., Gu, J., & Zhou, Z. (2012). A new model of information content based on concept's topology for measuring semantic similarity in wordnet. *International Journal of Grid and Distributed Computing*, *5*, 81–94.

[46] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.* (pp. 3111–3119).

[47] Miller, G., & Charles, W. (1991). Contextual correlates of semantic similarity. *Language and Cognitive Processes*, *6*, 1–28.

[48] Minnebo, W., & Stijven, S. (2011). *Empowering knowledge computing with variable selection*. Ph.D. thesis Ph. D. dissertation, Dept. Comput. Sci. Math., Univ. at Antwerp, Antwerp.

[49] Nguyen, H. T., Duong, P. H., & Cambria, E. (2019). Learning short-text semantic similarity with word embeddings and external knowledge sources. *Knowl.-Based Syst.*, *182*. URL: https://doi.org/10.1016/j.knosys.2019.07.013. doi:10.1016/j.knosys.2019.07.013.

[50] Pedersen, T., Pakhomov, S. V. S., Patwardhan, S., & Chute, C. G. (2007). Measures of semantic similarity and relatedness in the biomedical domain. *Journal of Biomedical Informatics*, *40*, 288–299. doi:10.1016/j.jbi.2006.06.004.

[51] Pirrò, G., & Euzenat, J. (2010). A feature and information theoretic framework for semantic similarity and relatedness. In *The Semantic Web - ISWC 2010 - 9th International Semantic Web Conference, ISWC 2010, Shanghai, China, November 7-11, 2010, Revised Selected Papers,*

Part I (pp. 615–630). URL: https://doi.org/10.1007/978-3-642-17746-0_39. doi:10.1007/978-3-642-17746-0_39.

[52] Potash, P., Boag, W., Romanov, A., Ramanishka, V., & Rumshisky, A. (2016). Simihawk at semeval-2016 task 1: A deep ensemble system for semantic textual similarity. In *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016* (pp. 741–748). URL: https://www.aclweb.org/anthology/S16-1115/.

[53] Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, IJCAI 95, Montréal Québec, Canada, August 20-25 1995, 2 Volumes* (pp. 448–453). URL: http://ijcai.org/Proceedings/95-1/Papers/059.pdf.

[54] Resnik, P. (1999). Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *J. Artif. Intell. Res.*, *11*, 95–130. doi:10.1613/jair.514.

[55] Rychalska, B., Pakulska, K., Chodorowska, K., Walczak, W., & Andruszkiewicz, P. (2016). Samsung poland NLP team at semeval-2016 task 1: Necessity for diversity; combining recursive autoencoders, wordnet and ensemble methods to measure semantic similarity. In *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016* (pp. 602–608). URL: https://www.aclweb.org/anthology/S16-1091/.

[56] Sánchez, D., Batet, M., & Isern, D. (2011). Ontology-based information content computation. *Knowl.-Based Syst.*, *24*, 297–303. URL: https://doi.org/10.1016/j.knosys.2010.10.001. doi:10.1016/j.knosys.2010.10.001.

[57] Seco, N., Veale, T., & Hayes, J. (2004). An intrinsic information content metric for semantic

similarity in wordnet. In *Proceedings of the 16th Eureopean Conference on Artificial Intelligence, ECAI'2004, including Prestigious Applicants of Intelligent Systems, PAIS 2004, Valencia, Spain, August 22-27, 2004* (pp. 1089–1090).

[58] Taieb, M. A. H., Aouicha, M. B., & Hamadou, A. B. (2014). Ontology-based approach for measuring semantic similarity. *Eng. Appl. of AI*, *36*, 238–261. URL: https://doi.org/10.1016/j.engappai.2014.07.015. doi:10.1016/j.engappai.2014.07.015.

[59] Thorleuchter, D., den Poel, D. V., & Prinzie, A. (2010). Mining ideas from textual information. *Expert Syst. Appl.*, *37*, 7182–7188. URL: https://doi.org/10.1016/j.eswa.2010.04.013. doi:10.1016/j.eswa.2010.04.013.

[60] Tversky, A. (1977). Features of similarity. *Psychological review*, *84*, 327.

[61] Vladislavleva, E., Smits, G., & den Hertog, D. (2010). On the importance of data balancing for symbolic regression. *IEEE Trans. Evolutionary Computation*, *14*, 252–277. URL: https://doi.org/10.1109/TEVC.2009.2029697. doi:10.1109/TEVC.2009.2029697.

[62] Zhao, F., Fang, F., Yan, F., Jin, H., & Zhang, Q. (2012). Expanding approach to information retrieval using semantic similarity analysis based on wordnet and wikipedia. *International Journal of Software Engineering and Knowledge Engineering*, *22*, 305–322. URL: https://doi.org/10.1142/S0218194012500088. doi:10.1142/S0218194012500088.