Undefined 0 (2022) 1–0
IOS Press

Sustainable Semantic Similarity Assessment

Jorge Martinez-Gil [a,*] and Jose Manuel Chaves-Gonzalez [b]
[a] *Software Competence Center Hagenberg*
Softwarepark 32a, 4232 Hagenberg, Austria
E-mail: jorge.martinez-gil@scch.at
[b] *University of Extremadura - Department of Computer Systems Engineering*
Centro Univ. Mérida, Mérida, Spain
E-mail: jm@unex.es

Abstract. The automatic semantic similarity assessment field has attracted much attention due to its impact on multiple areas of study. In addition, it is also relevant that recent advances in neural computation have taken the solutions to a higher stage. However, some inherent problems persist. For example, large amounts of data are still needed to train solutions, the interpretability of the trained models is not the most suitable one, and the energy consumption required to create the models seems out of control. Therefore, we propose a novel method to achieve significant results for a sustainable semantic similarity assessment, where accuracy, interpretability, and energy efficiency are equally important. We rely on a method based on multi-objective symbolic regression to generate a Pareto front of compromise solutions. After analyzing the output generated and comparing other relevant works published, our approach's results seem to be promising.

Keywords: Knowledge Engineering, Sustainable Computing, Semantic Similarity Assessment

1. Introduction

Determining semantic similarity between pieces of text is a significant challenge for the scientific community since the results achieved in this domain can impact a wide range of disciplines, including retrieving information of a textual nature. As a result, numerous solutions have been proposed to address the problem. Among them, recent advances in the abstract representation of words and sentences achieved by BERT [9] and ELMo [43] stand out. These approaches and their different variants have achieved remarkable results in several competitions. However, some inherent drawbacks are often overlooked.

For example, these approaches need vast data to be adequately trained. Furthermore, while it is usually not difficult for general-purpose solutions to find large amounts of data, it is often a problem in much more restricted domains. In addition, these solutions are hardly interpretable. This means that a human operator can provide input and obtain an output. Nevertheless, it is impossible to explain how the model has produced this output. Although it may not seem so, this leads to problems of a legal and ethical nature and various practical issues. Since people are not likely to trust systems that they cannot fully understand. Therefore, this is a very limiting factor of this kind of solution.

Last but not least, the latest advances in neural computation require significant investments in hardware with very high associated energy consumption to deploy the models. Despite the efforts to date, no sufficiently good alternatives to the problem have been found. Therefore, motivated by the reasons mentioned above, our research focuses on developing a more sustainable solution.

Remark 1. Research Gap. Though semantic similarity assessment through computers is currently attracting attention, the community has remained indifferent towards the interpretability of the resulting models and the energy efficiency required to build and exploit the resulting solutions. In this context, deep neural networks have been effectively used to produce

*Corresponding author

highly accurate assessment methods, but there are several issues inherent to the neural nature of these solutions. The challenge of establishing the ideal model structure, the demand for extensive training datasets, and the time-consuming nature of numerical learning methods are all issues with these deep neural network models. The state-of-art misses the study of alternative approaches to overcome these problems.

Remark 2. Motivation. The motivation of this research work is to build semantic similarity assessment methods that are accurate, interpretable, and energy-efficient at the same time. There have been significant attempts to address these issues, but always in a separate form. While it is true that existing solutions achieve very high accuracy, this is often at the expense of vast amounts of data for training. In addition, some work is already beginning to highlight the need to improve the interpretability of neural solutions [37]. However, more efforts are still needed in this direction. Last but not least, there are ways to reduce energy consumption, for example, by designing hardware (e.g., processors, memory, disks) so that running software consumes less energy. Other branch tries using mechanisms for scheduling instructions to make them more efficient. However, alternative strategies for creating energy-efficient code from scratch remain largely unexplored in this domain.

Remark 3. Contributions. Given Remarks 1 and 2, it can be established that the two most significant scientific and technical contributions to the state-of-the-art of the current approach could be summarized as follows:

1. **C1.** We propose a novel strategy for semantic similarity that consists of developing algorithms designed to be accurate, interpretable, and energy-efficient from the beginning. We rely on symbolic regression and multi-objective optimization (MOO). Through symbolic regression, we create models capable of evolving towards accuracy, interpretability, and energy efficiency smartly, and using MOO techniques, we force our approach to be guided by optimizing each of these orthogonal objectives simultaneously.
2. **C2.** We evaluate this novel strategy for program synthesis using widely accepted benchmark datasets used for the assessment of semantic similarity and compare the results obtained to the state-of-the-art solutions. Since, as far as we are concerned, this is the first attempt to use such kind of strategy to meet the three objectives mentioned above simultaneously, we aim to segment the different comparisons with the other well-known approaches to determine if the strategy presented here is competitive in each of the addressed aspects separately.

This manuscript is structured as follows: Section 2 describes the state-of-the-art concerning methods and tools for automatic semantic similarity assessment using computers. Section 3 presents the foundations that explain our sustainable semantic similarity measurement approach. Section 4 reports the findings extracted from several experiments, including using the most popular benchmark datasets in this context, and we compare these results with those obtained by other approaches. Finally, we remark on the strengths and flaws of our proposal and discuss the future work in Section 5.

2. State-of-the-art

It is widely assumed that automatically evaluating the semantic similarity between pieces of textual information is a complex research problem that requires a multidisciplinary approach to address it. Nevertheless, because of its importance to industry and academia, this challenge has attracted much attention recently [2,6,46]. The rationale for this is that models that can accurately identify the semantic similarity between two pieces of text could open up new ways to impact such diverse sectors as basic research or the business world.

The scientific community has long aspired to automatically determine the semantic similarity of textual fragments reflecting the same real-world thing or idea, even if their lexicography differs. For many years, semantic similarity methods have been used in many computer-related fields [6]. Even today, a substantial and expanding corpus of academic study based on a variety of techniques exists [14,27,28,42,50]. In recent years, new neural embedding approaches have gained much traction [39]. To such an extent that today, these seminal works have inspired state-of-the-art solutions such as BERT [9] or ELMo [43]. However, three main issues persist:

1. The first issue is that these techniques rely on large amounts of data to train models. Whether developed or detected using pre-trained deep models, the input features may be impacted by noise inherent in raw data, making them impre-

cise. In addition, the mappings between data features and objective variables must be robust to data noise and other factors such as outliers and the adoption of a non-optimal model structure. We have discussed the use of semantic similarity controllers in the past as a way to solve these challenges [36]. These controllers are artifacts that may be developed automatically to avoid the issues we discussed before. However, understanding fuzzy code still needs a certain degree of mastery.

2. The second issue is the lack of interpretability, i.e., the inability of a human operator to understand the model. This is a characteristic since understanding a model with many interconnected nodes is widely considered rather difficult. The reason is that a human operator can specify which outputs correlate to which inputs and the deep neural network will automatically design a mapping function. However, the human operator will not know what happens inside the model. As a result, these models are often described as black-boxes because they do not show their operation insights to the users. Although in recent times much research is indeed being carried out to mitigate this problem [45], the solutions are not yet entirely satisfactory.
3. The third issue is that energy saving is one of the major concerns in today's societies. It is relevant to note here that data centers worldwide consume more than 320 terawatt-hours of electricity currently, which is more than 3% of the world's total electricity consumption[1]. Data centers assume that facility expenses have become significant cost factors. Moreover, highest energy impact of deep learning models is not just because of their training but rather because of their deployment in the cloud, being continuously online, making computations in real-time for thousands or millions of parameters. These reasons explain that the engineers in charge of maintaining these centers strongly warn that if energy consumption continues to grow, the expenses of the model's life cycle may exceed the cost related to the hardware by a wide margin, not to mention its environmental impact (e.g., carbon footprint). Some works have tried to improve our understanding of the consumption patterns of a program for writing sustainable, energy-efficient, and green code [1,32]. Now, we go a step further since one of the advantages of symbolic regression is that it can reduce software energy consumption by optimizing the source code.

[1] https://ukcop26.org/

Therefore, to date, very little attention has been paid to the sustainability of the models [13]. The novelty of this work lies in that we present a strategy for building more sustainable semantic similarity assessment systems for the first time. We emphasize the importance of facing a threefold goal: to make the models accurate, interpretable, and energy-efficient at the same time. Furthermore, we will rely on symbolic regression and MOO techniques to build our system. This approach is innovative in that it:

The challenge of finding a symbolic expression to identify the relationship between defined inputs and output variables has already been studied by the community. The key idea is that the expressions generated should be flexible enough without being restricted to a particular structure [47]. This technique is constrained by the choice of operations that are permitted in the sought equations [20]. Nevertheless, the resulting model is an equation that can be executed and interpreted in the context of the situation [37].

Concerning multi-objective symbolic regression, several works have appeared in recent times that address the problem [24,30]. Such techniques, as [16], aim to model problems involving conflicting objectives in the classical ways. Although there are some improvements, e.g., using semantic genetic programming [5], its applications have been little explored to date. In this work's context, we focus on combining symbolic regression with MOO, allowing us to shape our model in the way we need, for the first time.

In summary, neither the interpretability of semantic similarity measurement nor generating energy-efficient models have yet been explored in depth. Therefore, in the remainder of this work, we will focus on developing sustainable methods for automatically assessing semantic similarity between pieces of textual information. To do that, we aim to reach sustainability using a MOO similarity learning problem whereby three orthogonal objectives (accuracy, interpretability, and energy efficiency) are to be pursued simultaneously.

3. Sustainable Semantic Similarity Assessment

To face the challenge of designing more sustainable methods for semantic similarity assessment, we aim to combine symbolic regression, performed via genetic programming (GP), with MOO techniques. Symbolic regression explores the space search of all computer programs to find the one that best solves a given problem. No particular assumption is made as a starting point. This characteristic gives us enormous advantages in designing highly efficient approaches from the beginning. This approach takes nothing for granted, so the method is free to evolve to forms with high accuracy, interpretability, and low energy consumption. This is whereby MOO comes in since this approach involves more than one goal to be simultaneously met. MOO is helpful in scenarios where decisions need to be taken regarding two or more orthogonal objectives.

We can define our problem formally as follows: $K^O : R^p \rightarrow R$ that best fits a given training dataset $T = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ of n input and output pairs with $x_i \in R^p, y_i \in R$ defined as

$$K^O, \theta^O \leftarrow \underset{K \in G; \theta \in R^m}{arg\,min}\ f(K(x_i, \theta), y_i) \tag{1}$$

where G is the solution or solution space defined by the primitive set P of functions and terminals, f is the fitness function which is based on the difference between the model output $K(x_i, \theta)$. The desired output y_i, and θ is a particular parametrization of the mathematical expression K, assuming m are real-valued.

Furthermore, research problems with two or more orthogonal objectives have received much attention in recent decades. In this way, meta-heuristics have successfully addressed MOO problems, resulting in numerous techniques that offer an accurate approximation to the Pareto front of the solved issue. The decision-maker must next determine which alternatives are the most appropriate based on a set of criteria or preferences. We will also study which kind of meta-heuristic strategy best fits the problem.

3.1. Importance of Symbolic Regression

Symbolic regression is a computational approach that explores extensively across the space in which all equations are specified to discover the formula that best matches a specific dataset. Symbolic regression has already been applied previously in [15]. The authors proposed a method to solve specific problems associated with the identification and learning functions in that work. This application is possible thanks to Abstract Syntax Trees (ASTs), which allow identifying any function from past solves cases. This makes it easier for a human operator to grasp it and apply it to other issues of a similar kind [37].

Our aim is for the AST to grow to the point where it can find an expression that matches the input and output pairs supplied as training data and then validates that expression in a different situation. In each iteration, all candidates in the population are evaluated for how well they satisfy the objective. Those with higher scores are more likely to pass on to the next iteration. By introducing sources of random variation in each candidate solution, new candidates are generated, each with a possibility of being closer to the actual target solution.

To do that, we seek to aggregate modern similarity techniques strategically. The possible mistakes that a method could make lose importance on an ensemble of techniques that generally blur any of these mistakes [35]. In this way, only if all methods produce the same error does the aggregation lose its usefulness. Popular operations in this field are the arithmetical mean or the median. However, their strategy is widely considered to be short-sighted and does not usually lead to optimal results in this context [33].

Therefore, we combine methods with mathematical operators and numerical constants to get the goal function. As mentioned in [25], this model may evolve owing to an evolutionary algorithm. The result is obtained by assessing the nodes and then applying the parent operation to the children [37]. Our research leads to three interesting facts: first, the symbolic regression can find an expression that can adequately consider each of the measures of semantic similarity; second, in contrast to other neural network-based models, the generated expression can be recognized and understood by a human being without requiring any special training; and third, the energy consumption of the expression, although we can assume it is not very important for a single run, could be optimized to see a positive effect after millions of repeated executions.

In our strategy, the automatic generation of the AST must be guided by the three aforementioned main objectives. The problem is that, in the realm of MOO, there is not a single solution that can meet all of the goals at the same time. As a result, solutions that cannot improve the objectives without harming the remainder must be prioritized. In this way, the collection

of Pareto optimum points from the search space leads to a Pareto front, representing the best feasible compromise between the examined orthogonal objectives [38]. In our specific case, the three orthogonal objectives that define our learning phase are the following:

- Concerning *accuracy*, we try to maximize the Pearson or the Spearman Rank correlation coefficients since the research community usually measures semantic similarity as the difference in the correlation of human judgment (or ground truth) and artificially generated solutions. Therefore, the unit of measurement will be the degree of correlation with human judgment.
- Concerning *interpretability*, we try to minimize the size and the complexity of the final mathematical equation generated by the evolutionary strategy. To do so, we will work with individuals of a specific maximum size to reduce them as much as possible. Therefore, the unit of measurement will be the length of the generated mathematical equation, taking into account that the size is measured in the number of nodes.
- Concerning *energy efficiency*, we try to minimize the energy consumption. To do that, we use the pyRAPL[2] model to calculate the energy required by the execution of the symbolic program at runtime. This model is valid only on Intel CPUs (it remains future work to study other processors). The unit of measurement will be the Joule since it represents the energy dissipated as heat in the CPU.

Figure 1 shows a clear example of the solutions we are looking for. We want a three-dimensional solution Pareto front, where a human operator can choose the solution that best fits his specific needs. Since the three parameters are challenging to optimize simultaneously, choosing a solution that optimizes two will almost always be possible—for example, accuracy and interpretability, accuracy and energy efficiency, or interpretability and energy efficiency.

From now on, we will explain the details of the empirical study for the sustainable calculation of semantic similarity. In addition, we will provide an analysis of the applicability of this novel strategy to production environments.

[2]https://github.com/powerapi-ng/pyRAPL

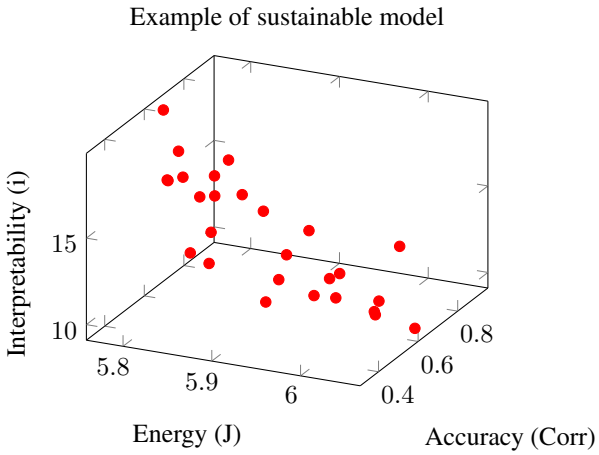


Fig. 1. Three-dimensional visualization of the front-end Pareto generated when learning an equation that optimizes accuracy, interpretability and energy efficiency at the same time. The X-axis represents the energy consumption in Joules. The Y-axis represents the interpretability in AST items, or if preferred, the size of the resulting model or mathematical equation. The Z-axis expresses the degree of correlation with respect to human judgment.

4. Results

The outcomes we got in our experiments are presented here. To accomplish so, we outline our experimental strategy setup, including the benchmark datasets we have used, the different objectives to be reached, and the base configuration of the approaches under consideration. We thoroughly examine the various MOO techniques explored and the empirical results. Furthermore, we provide a comparison with previous studies, including those that emphasize accuracy. We also offer a temporal analysis of the different training phases for the approaches under consideration and discuss the outcomes obtained.

4.1. Experimental setup

First, we will go through the benchmark datasets that we have utilized, the objective functions our strategy should aim towards, and finally, the base setup we used in the test we performed, ensuring that the experiments are repeatable.

4.1.1. Datasets

Our research uses a dataset that has become standard for working with general-purpose solutions. The dataset is known as the Miller & Charles [40] dataset. This benchmark compares textual information from various general-purpose contexts, i.e., terms we can

find in many traditional settings. We use the version with 30-word pairs (MC30), while many researchers utilize shorter versions (e.g., 28-word pairs) due to problems concerning dictionary coverage.

4.1.2. Goals

The fitness function guides the learning process through the Spearman Rank correlation and Pearson Correlation Coefficient. This last one is calculated between two vectors and aims to assess the linear relationship between both vectors. On the other hand, if the Spearman Rank correlation is used, a coefficient to measure how equals the vectors generated by the human and the machine. The correlation defined by Spearman is a reasonable goal when results need to be compared on an ordinal base. The distinction between these two correlations is that Pearson's correlation is better for situations with an absolute scale, while Spearman's correlation works better with relative scales.

4.1.3. Parameter setup

A standard grid search strategy has made us choose the following parameter settings:

- Set of functions $\{+, -, \cdot, exp, /, max, min\}$ (where division is protected)
- Individual sizes [0 - 50]: **22**
- Highest number of constants permitted [0 - 5]: **3**
- Maximum allowed depth [2 - 5]: **4**
- Population length [10 - 100]: **25**
- Mutation percentage [0.00 - 0.5]: **0.15**
- Crossover percentage [0.3 - 0.95]: **0.70**

The optimization of values guides the learning process in the training phase. Because the approaches we use are of stochastic nature, this procedure has been repeated 30 times so that it can be possible to achieve more robust results.

4.2. Analysis of strategies

MOO aims to learn a function that meets several orthogonal objectives simultaneously. There is no single optimal solution for problems of this kind. Therefore, according to the state-of-the-art in the area, our proposed approach is compared concerning five of the most representative strategies within the MOO domain. Furthermore, we have made such a selection based on the explanations of [11]. In our scenario, we want to reduce the energy used and the size of the generated equation while also increasing the correlation to human judgment (accuracy). In addition, it is helpful to emphasize one crucial point: our technique serves as a guide for achieving the best feasible outcome on a training dataset. However, the findings that we describe were obtained using a blind data set. This is appropriate to ensure that the final model has learned a correct setup for generalizing the results.

Concerning the different MOO strategies, we rely on the framework MOEA[3]. The different strategies, in alphabetical order, are: CellDE [10], CMAES [19], DBEA [22], GDE3 [26], MOEA/D[48], MSOPS [18], NSGA-II [7], NSGA-III [8], PAES [23], and SMPSO [41]. After a preliminary study, we show the five most promising (in terms of quantity and quality of the solutions) ones below: CellDE [10], CMAES [19], GDE3 [26], MOEA/D [48], and NSGA-II[7].

4.2.1. CellDE

CellDE [10] is a popular approach in the field of MOO. It obtains outstanding results for several reasons: it relies on efficient differential evolution and takes the idea of storing non-dominated solutions from other MOO approaches. These two design features give the strategy outstanding results in scenarios involving more than two orthogonal objectives.

We can see that it is feasible to gain more accuracy by using more sophisticated models, as demonstrated in Figure 2. It can also be shown that more energy is necessary to make the generated equation more interpretable (i.e., less complex).

4.2.2. CMAES

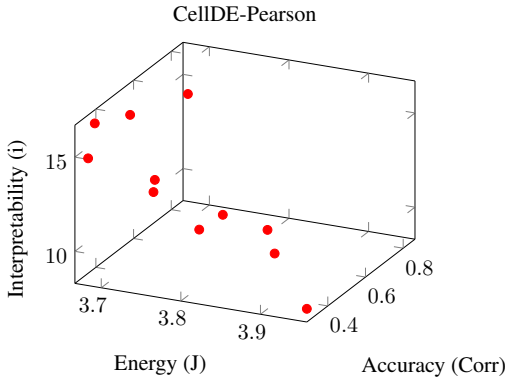
CMA-ES [19] is a stochastic approach that is usually involved in the optimization of non-convex continuous problems. Its design is supported by two ideas: the core notion of maximum likelihood and the analysis of evolution records, which are used to monitor the correlation between consecutive iterations.

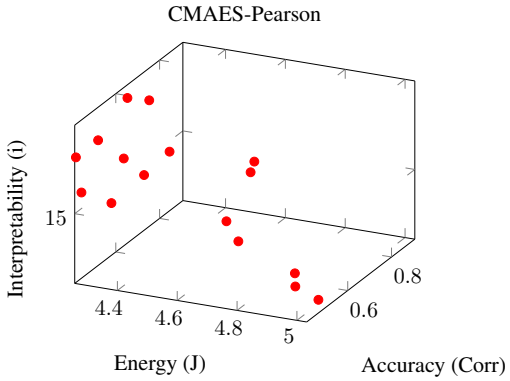
A solution front is shown in Figure 3, where a more straightforward way to comprehend the model is acquired at the expense of accuracy and vice versa. Furthermore, there is a substantial divergence between the Pearson correlation coefficient and the Spearman rank correlation.

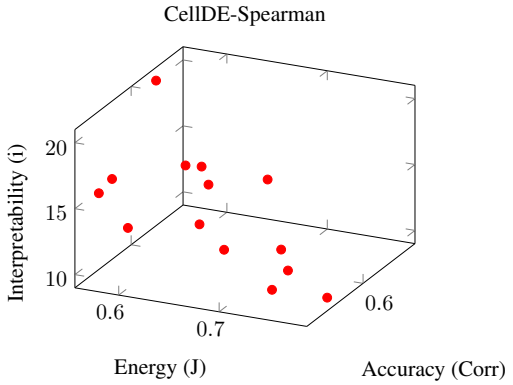
4.2.3. GDE3

GDE3 [26] employs differential evolution to optimize a problem by keeping a population of candidates and merging current individuals using a simple formula to create new candidate solutions. Differential

[3] http://moeaframework.org/







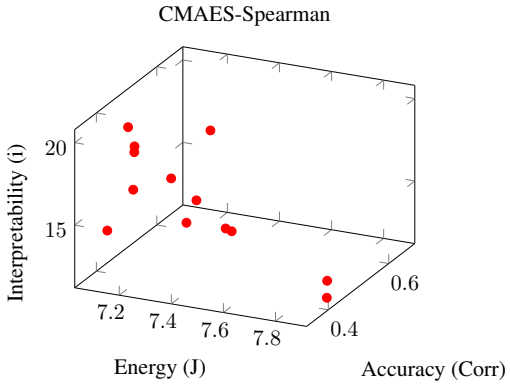


Fig. 2. CellDE was used to create a Pareto front of non-dominated points. The Pearson Correlation Coefficient is depicted in the first plot, while the Spearman Rank Correlation is depicted in the second.

Fig. 3. CMAES was used to create a Pareto front of non-dominated points. The Pearson Correlation Coefficient is depicted in the first plot, while the Spearman Rank Correlation is depicted in the second.

evolution methods are simple, efficient, and frequently produce good results in different MOO settings.

When we look at the experiments, we can see that GDE3 is one of the more successful ways to proceed. The results that GDE can produce for the two correlation coefficients of interest are shown in Figure 4.

4.2.4. MOEA/D

MOEA/D [48] is an evolutionary approach that relies on the concept of dividing the scene into several single-objective problems. MOEA/D usually performs better with MOO problems involving more than three conflicting objectives.

We have got a solution front for the two scenarios under study, as shown in Figure 5. The orthogonality explains the relationship between accuracy, the complexity of the equation required for the technique, and energy usage. The Spearman values are somewhat higher than the Pearson values in terms of accuracy.

4.2.5. NSGA-II

When the goals number is modest, NSGA-II [7] is a popular way to implement a MOO strategy. The algorithm employs the critical notion of dominance to provide better results than others since it enhances a single target without degrading others.

In general, NSGA-II-based methods provide good results. In both circumstances, this method comes out on top. This is because this method has been demonstrated to function well when several objectives are being pursued simultaneously. The outcomes of employing the NSGA-II approach are shown in Figure 6.

4.3. Comparison with other approaches

The best solutions found utilizing the various MOO techniques are compared here. It is not possible to directly compare our technique to any other existing proposal since it is the first to investigate the trade-off between accuracy, interpretability, and energy efficiency when measuring semantic similarity. Consequently, we will evaluate the findings for each factor studied about the wide range of available methods for automatically calculating semantic similarity.

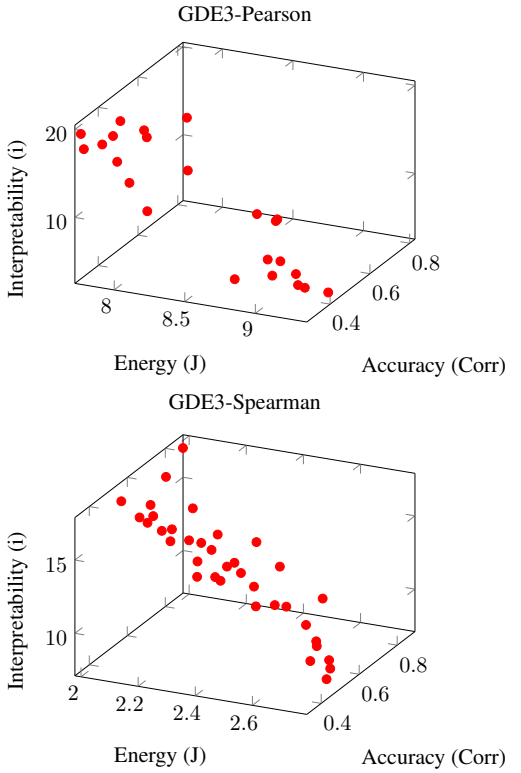


Fig. 4. GDE3 was used to create a Pareto front of non-dominated points. The Pearson Correlation Coefficient is depicted in the first plot, while the Spearman Rank Correlation is depicted in the second.

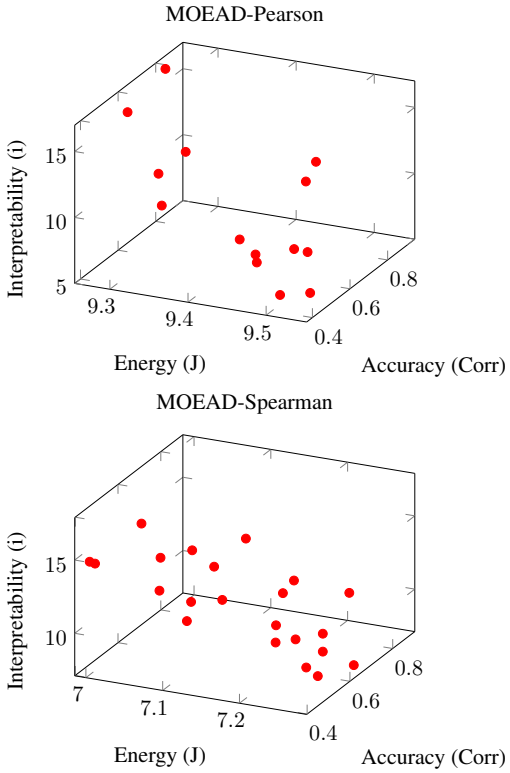


Fig. 5. MOEA/D was used to create a Pareto front of non-dominated points. The Pearson Correlation Coefficient is depicted in the first plot, while the Spearman Rank Correlation is depicted in the second.

In Table 2 the values for the Pearson Correlation are compared when solving the MC30 instance. Please note that in that table are shown the best findings from our empirical analysis. Even though the outcomes are contingent on how the model is trained, we can see that some configurations can produce superior results than those achieved using traditional approaches. Once again, we choose the average value because we deal with non-deterministic methods.

The best results achieved while solving the MC30 dataset with the Spearman Correlation coefficient are listed in Table 3. We provide the state-of-the-art and the top outcomes obtained using our method. As can be shown, some combinations can produce superior results than those produced using traditional approaches. There is, however, greater variety than in the prior situation. Furthermore, the complexity of the equation and the amount of energy consumed are also considered.

We can observe that our approach can place different configurations among the best ones for the MC30 benchmark dataset, which represents a good result if we also consider that the model's interpretability and its energy consumption, have been considered to achieve this score. This confirms our hypothesis that such a strategy can make sense when solving challenges such as the one studied here.

Algorithm	**Score (p-value)**	Interpretable?	Energy-Efficient?
Huang et al. [17]	0.659 ($7.5 \cdot 10^{-5}$)	Yes	No
Resnik [44]	0.780 ($1.9 \cdot 10^{-7}$)	Yes	No
Leacock & Chodorow [29]	0.807 ($4.0 \cdot 10^{-8}$)	Yes	No
Lin [31]	0.810 ($3.0 \cdot 10^{-8}$)	Yes	No
Faruqui & Dyer [12]	0.817 ($2.2 \cdot 10^{-8}$)	No	No
Mikolov et al. [39]	0.820 ($2.0 \cdot 10^{-8}$)	No	No
GDE3	0.831 ($1.4 \cdot 10^{-8}$)	**Yes**	**Yes**
CoTO [34]	0.850 ($1.0 \cdot 10^{-8}$)	Yes	No
MOEAD	0.851 ($1.0 \cdot 10^{-8}$)	**Yes**	**Yes**
FLC [36]	0.855 ($1.0 \cdot 10^{-8}$)	Yes	No
CellDE	0.862 ($8.5 \cdot 10^{-9}$)	**Yes**	**Yes**
CMAES	0.906 ($3.2 \cdot 10^{-9}$)	**Yes**	**Yes**
NSGA-II	0.914 ($2.5 \cdot 10^{-9}$)	**Yes**	**Yes**

Table 1
Correlation according to the Pearson for the existing approaches when the MC30 dataset is tested

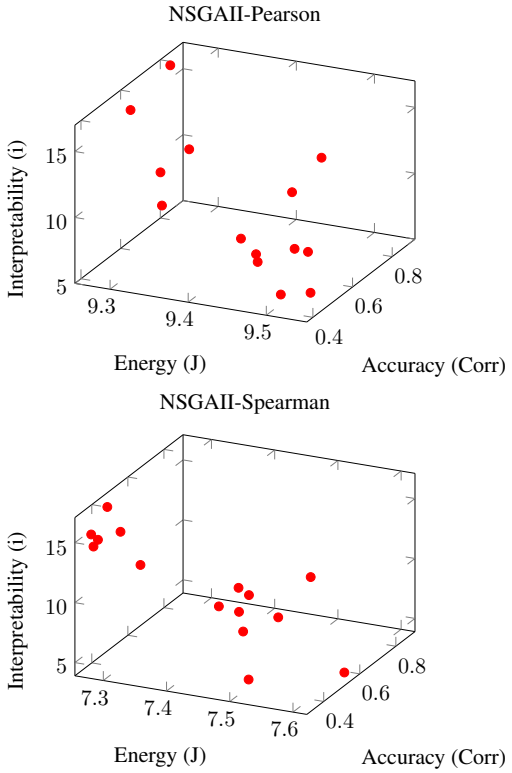


Fig. 6. NSGA-II was used to create a Pareto front of non-dominated points. The Pearson Correlation Coefficient is depicted in the first plot, while the Spearman Rank Correlation is depicted in the second.

Algorithm	**Score(p-value)**	Interpretable?	Energy-Efficient?
Lin [31]	0.619 ($1.6 \cdot 10^{-4}$)	Yes	No
Aouicha et al. [3]	0.640 ($8.0 \cdot 10^{-5}$)	Yes	No
CMAES	0.686 ($2.1 \cdot 10^{-5}$)	**Yes**	**Yes**
Resnik [44]	0.757 ($5.3 \cdot 10^{-7}$)	Yes	No
Mikolov et al. [39]	0.770 ($2.6 \cdot 10^{-7}$)	No	No
CellDE	0.779($1.1 \cdot 10^{-7}$)	**Yes**	**Yes**
Leacock & Chodorow [29]	0.789 ($8.1 \cdot 10^{-8}$)	Yes	No
Bojanowski et al. [4]	0.846 ($6.3 \cdot 10^{-9}$)	Yes	No
Zhao et al. [49]	0.857 ($1.4 \cdot 10^{-9}$)	Yes	No
FLC [36]	0.891 ($8.3 \cdot 10^{-12}$)	Yes	No
GDE3	0.893 ($1.1 \cdot 10^{-14}$)	**Yes**	**Yes**
MOEAD	0.903 ($1.9 \cdot 10^{-14}$)	**Yes**	**Yes**
NSGA-II	0.906 ($2.1 \cdot 10^{-14}$)	**Yes**	**Yes**

Table 2

Correlation according to the Spearman Rank for the existing approaches when the MC30 dataset is tested

4.4. Performance

We also analyze the performance for the training phase. In Figure 7, we show the average time for each of the MOO strategies considered. These times represent the average time (in milliseconds) resulting from 30 independent runs.

On the one hand, the MOEA/D method is the fastest approach. However, the this approach does not rank among the top. On the other hand, techniques that produce superior results, such as NSGA-II, need higher operation times to achieve the Pareto fronts.

4.5. Discussion

Experiments prove that it is feasible to find a solution whose accuracy, interpretability, and energy-efficient values cannot be improved except at the expense of the others. However, by studying the results separately, one can observe that it is possible to optimize two values at the cost of the other. There may be accurate and interpretable solutions, but they will require more CPU time to execute (think, for example, the max operator that requires much computation underneath even though it only occupies one item of the AST). There may be accurate and energy-efficient solutions but not interpretable (since they need large equations). Finally, there may be interpretable and energy-efficient solutions but will not be accurate (they will be executed quickly and cheaply but will not achieve the best results). These results are novel in that, to date, the community has not been very concerned about the development of solutions that give rise to sustainable models.

Regarding accuracy, it is necessary to remark, that additional experiments show us that performance decreases as the size of the input data increases. But this is true for all existing models that work with semantic similarity [28]. The reason is that it is necessary to learn more general models that deal with much more volume and diversity in the data to be processed.

Regarding a concrete example of interpretability, the AST[4] *MAX (MAX (ssm3 · ssm2, 2 · ssm1 + ssm2), ssm4/ssm3)* achieves an accuracy of 0.87 when solving the MC30. This AST requires 13 nodes (and 4 free variables). At the same time, the *AST MAX (MAX (ssm3 · ssm2, 2 · ssm1 + ssm2), 1)* achieves an accuracy of 0.83 with 11 nodes (and 3 free variables). It is therefore up to the human operator to choose the model to be exploited: More accuracy with more complex equations (larger size and more free variables) or less accuracy with simpler equations (smaller size and fewer free variables).

Finally, it also seems clear that the increasingly high speeds that microprocessors can reach pose a compro-

[4]Please note ssm1=[21], ssm2=[29], ssm3=[31], and ssm4=[44]

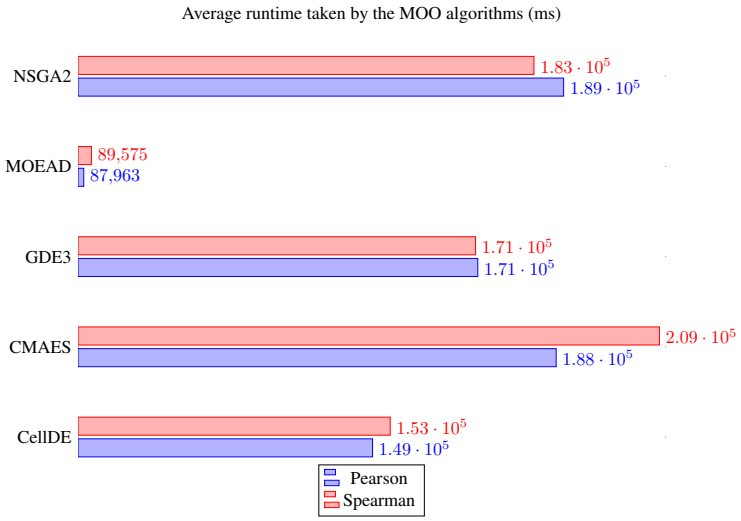


Fig. 7. Average execution time of the MOO strategies considered. The figures are given in milliseconds

mise to their energy consumption (as well as their reliability and lifetime). Therefore, minimizing their energy consumption is a significant challenge. However, many data centers are usually quite optimized since operators have strong incentives to reduce energy consumption. Thus, the less energy a data center uses, the greater the economic benefit of the organization.

However, using energy-efficient algorithms methods is still largely unexplored to date. It is assumed that as the customer base grows, the energy needed to provide the service grows as well. In this way, both hardware-based solutions and instruction planning are well thought out. However, our strategy can solve this problem by creating assessment strategies offered in a low-energy version of the source code to be executed. When working with large data environments, its impact on energy consumption will be significant. Besides, as the operator is likely to save money, it might encourage its users to use this kind of strategy.

Overall, we find it interesting to remark that we will have little room for improvement if we optimize traditional data science models because they involve many assumptions regarding the parameters they operate with. With symbolic regression and MOO, computer libraries can implement analogous functions (classification, regression, clustering, and optimization) with higher interpretability and lower energy consumption. The reason is that the models generated have to assume those requirements from the beginning. This means, for example, that accuracy has to be as crucial as interpretability and low energy consumption by design. Alternatively, the human operator should be offered a front of orthogonal solutions to decide which configuration best fits the needs of its specific case.

5. Conclusions and Future Work

People should be able to trust the data-driven technologies they utilize in their everyday operations as they become more important in many daily situations. Unfortunately, several technological domains have been immersed in a rush to enhance accuracy in recent years. As a result, novel solutions have paid insufficient attention to other critical factors, such as the long-term viability of the models they work with.

To overcome this situation, we have shown how to design a strategy that considers three fundamental objectives to achieve a sustainable assessment: accuracy, interpretability, and energy efficiency. Our study shows that even if it is impossible to get optimal solutions for all the three objectives, it is feasible to obtain a model that allows finding compromise solutions, leaving the decision-maker to choose the most suitable model for the scenario in which it has to operate. This represents a novelty for the community because it focuses, for the first time on sustainable models.

For future work, it would be good to consider that the limited number of libraries for calculating the energy consumption has been a limiting factor of this study. Our experimental setup has been performed only on Intel processors. However, an in-depth analysis of the implications of deploying our approach on processors from other manufacturers (AMD, ARM,

etc.) would be desirable. We want to point out that, even if the execution of a single case does not produce significant results, the cloud environment in which these solutions are used, often involving millions of executions, can have considerable savings associated.

Acknowledgments

The authors thank the anonymous reviewers for their comments and suggestions to improve the work. This research has been funded by the project NEFUSI (NGI Zero Discovery) by the NL Foundation and the EU Commission. Project number: 2021-04-069.

References

[1] Acar, H., Alptekin, G. I., Gelas, J., and Ghodous, P. (2016). The impact of source code in software on power consumption. *Int. J. Electron. Bus. Manag.*, 14.

[2] Afzal, N., Wang, Y., and 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*, pages 674–679.

[3] Aouicha, M. B., Taieb, M. A. H., and Hamadou, A. B. (2016). LWCR: multi-layered wikipedia representation for computing word relatedness. *Neurocomputing*, 216:816–843.

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

[5] Casadei, F., Martins, J. F. B. S., and Pappa, G. L. (2019). A multi-objective approach for symbolic regression with semantic genetic programming. In *8th Brazilian Conference on Intelligent Systems, BRACIS 2019, Salvador, Brazil, October 15-18, 2019*, pages 66–71. IEEE.

[6] Chandrasekaran, D. and Mago, V. (2021). Evolution of semantic similarity—a survey. *ACM Computing Surveys (CSUR)*, 54(2):1–37.

[7] Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.*, 6(2):182–197.

[8] Deb, K. and Jain, H. (2014). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. *IEEE Trans. Evol. Comput.*, 18(4):577–601.

[9] Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2019). BERT: pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

[10] Durillo, J. J., Nebro, A. J., Luna, F., and Alba, E. (2008). Solving three-objective optimization problems using a new hybrid cellular genetic algorithm. In Rudolph, G., Jansen, T., Lucas, S. M., Poloni, C., and Beume, N., editors, *Parallel Problem Solving from Nature - PPSN X, 10th International Conference Dortmund, Germany, September 13-17, 2008, Proceedings*, volume 5199 of *Lecture Notes in Computer Science*, pages 661–670. Springer.

[11] Emmerich, M. T. M. and Deutz, A. H. (2018). A tutorial on multiobjective optimization: fundamentals and evolutionary methods. *Nat. Comput.*, 17(3):585–609.

[12] Faruqui, M. and 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*, pages 462–471.

[13] Gómez-Galán, J., Vázquez-Cano, E., Luque de la Rosa, A., and López-Meneses, E. (2020). Socio-educational impact of augmented reality (ar) in sustainable learning ecologies: A semantic modeling approach. *Sustainability*, 12(21):9116.

[14] Han, L., Kashyap, A. L., Finin, T., Mayfield, J., and Weese, J. (2013). Umbc_ebiquity-core: Semantic textual similarity systems. In Diab, M. T., Baldwin, T., and Baroni, M., editors, *Proceedings of the Second Joint Conference on Lexical and Computational Semantics, *SEM 2013, June 13-14, 2013, Atlanta, Georgia, USA*, pages 44–52. Association for Computational Linguistics.

[15] Haslam, E., Xue, B., and Zhang, M. (2016). Further investigation on genetic programming with transfer learning for symbolic regression. In *IEEE Congress on Evolutionary Computation, CEC 2016, Vancouver, BC, Canada, July 24-29, 2016*, pages 3598–3605. IEEE.

[16] Hinde, C. J., Chakravorti, N., and West, A. A. (2016). Multi objective symbolic regression. In Angelov, P., Gegov, A. E., Jayne, C., and Shen, Q., editors, *Advances in Computational Intelligence Systems - Contributions Presented at the 16th UK Workshop on Computational Intelligence, September 7-9, 2016, Lancaster, UK*, volume 513 of *Advances in Intelligent Systems and Computing*, pages 481–494. Springer.

[17] Huang, E. H., Socher, R., Manning, C. D., and 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*, pages 873–882.

[18] Hughes, E. J. (2003). Multiple single objective pareto sampling. In *The 2003 Congress on Evolutionary Computation, 2003. CEC'03.*, volume 4, pages 2678–2684. IEEE.

[19] Igel, C., Hansen, N., and Roth, S. (2007). Covariance matrix adaptation for multi-objective optimization. *Evol. Comput.*, 15(1):1–28.

[20] Iqbal, M., Xue, B., Al-Sahaf, H., and Zhang, M. (2017). Cross-domain reuse of extracted knowledge in genetic programming for image classification. *IEEE Trans. Evol. Comput.*, 21(4):569–587.

[21] Jiang, J. J. and Conrath, D. W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. In Chen, K., Huang, C., and Sproat, R., editors, *Proceedings of the 10th Research on Computational Linguistics International Conference, ROCLING 1997, Taipei, Taiwan, August 1997*, pages 19–33. The Association for Computational Linguistics and Chinese Language Processing (ACLCLP).

[22] Jiang, S. and Yang, S. (2016). Convergence versus diversity in multiobjective optimization. In Handl, J., Hart, E., Lewis, P. R., López-Ibáñez, M., Ochoa, G., and Paechter, B., editors, *Parallel*

Problem Solving from Nature - PPSN XIV - 14th International Conference, Edinburgh, UK, September 17-21, 2016, Proceedings, volume 9921 of *Lecture Notes in Computer Science*, pages 984–993. Springer.

[23] Knowles, J. D. and Corne, D. (2000). Approximating the nondominated front using the pareto archived evolution strategy. *Evol. Comput.*, 8(2):149–172.

[24] Kommenda, M., Beham, A., Affenzeller, M., and Kronberger, G. (2015). Complexity measures for multi-objective symbolic regression. In Moreno-Díaz, R., Pichler, F., and Quesada-Arencibia, A., editors, *Computer Aided Systems Theory - EUROCAST 2015 - 15th International Conference, Las Palmas de Gran Canaria, Spain, February 8-13, 2015, Revised Selected Papers*, volume 9520 of *Lecture Notes in Computer Science*, pages 409–416. Springer.

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

[26] Kukkonen, S. and Lampinen, J. (2005). Gde3: The third evolution step of generalized differential evolution. In *2005 IEEE congress on evolutionary computation*, volume 1, pages 443–450. IEEE.

[27] Lastra-Díaz, J. J., García-Serrano, A., Batet, M., Fernández, M., and 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.

[28] Lastra-Díaz, J. J., Goikoetxea, J., Taieb, M. A. H., García-Serrano, A., Aouicha, M. B., and 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. Artif. Intell.*, 85:645–665.

[29] Leacock, C., Chodorow, M., and Miller, G. A. (1998). Using corpus statistics and wordnet relations for sense identification. *Comput. Linguistics*, 24(1):147–165.

[30] Lensen, A., Xue, B., and Zhang, M. (2020). Genetic programming for evolving similarity functions for clustering: Representations and analysis. *Evol. Comput.*, 28(4):531–561.

[31] 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*, pages 296–304.

[32] Mackowski, M. and Niezabitowski, M. (2015). Power consumption analysis of microprocessor unit based on software realization. In *20th International Conference on Control Systems and Computer Science, CSCS 2015, Bucharest, Romania, May 27-29, 2015*, pages 493–498. IEEE.

[33] Martinez-Gil, J. (2016a). Accurate semantic similarity measurement of biomedical nomenclature by means of fuzzy logic. *Int. J. Uncertain. Fuzziness Knowl. Based Syst.*, 24(2):291–306.

[34] Martinez-Gil, J. (2016b). CoTO: A novel approach for fuzzy aggregation of semantic similarity measures. *Cogn. Syst. Res.*, 40:8–17.

[35] 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(2):361–380.

[36] Martinez-Gil, J. and Chaves-Gonzalez, J. M. (2019). Automatic design of semantic similarity controllers based on fuzzy logics. *Expert Syst. Appl.*, 131:45–59.

[37] Martinez-Gil, J. and Chaves-Gonzalez, J. M. (2020). A novel method based on symbolic regression for interpretable semantic similarity measurement. *Expert Syst. Appl.*, 160:113663.

[38] Martinez-Gil, J. and Chaves-Gonzalez, J. M. (2021). Semantic similarity controllers: On the trade-off between accuracy and interpretability. *Knowledge-Based Systems*, page 107609.

[39] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and 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.*, pages 3111–3119.

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

[41] Nebro, A. J., Durillo, J. J., García-Nieto, J., Coello, C. A. C., Luna, F., and Alba, E. (2009). SMPSO: A new pso-based metaheuristic for multi-objective optimization. In *2009 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making, MCDM 2009, Nashville, TN, USA, March 30 - April 2, 2009*, pages 66–73. IEEE.

[42] Pesquita, C., Faria, D., Bastos, H., Ferreira, A. E., Falcão, A. O., and Couto, F. M. (2008). Metrics for go based protein semantic similarity: a systematic evaluation. In *BMC bioinformatics*, volume 9, pages 1–16. BioMed Central.

[43] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237.

[44] 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.

[45] Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "why should I trust you?": Explaining the predictions of any classifier. In Krishnapuram, B., Shah, M., Smola, A. J., Aggarwal, C. C., Shen, D., and Rastogi, R., editors, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144. ACM.

[46] Solovyev, V. and Loukachevitch, N. (2020). Semantic similarity of words in ruwordnet thesaurus and in psychosemantic experiment. In *International Conference on Cognitive Sciences*, pages 395–402. Springer.

[47] Trujillo, L., Muñoz, L., López, U., and Hernández, D. E. (2018). Untapped potential of genetic programming: Transfer learning and outlier removal. In Banzhaf, W., Spector, L., and Sheneman, L., editors, *Genetic Programming Theory and Practice XVI, [GPTP 2018, University of Michigan, Ann Arbor, USA, May 17-20, 2018]*, Genetic and Evolutionary Computation, pages 193–207. Springer.

[48] Zhang, Q. and Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comput.*, 11(6):712–731.

[49] Zhao, Z., Liu, T., Li, S., Li, B., and Du, X. (2017). Ngram2vec: Learning improved word representations from ngram co-occurrence statistics. In Palmer, M., Hwa, R., and Riedel, S., editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 244–253. Association for Computational Linguistics.

[50] Zhu, G. and Iglesias, C. A. (2017). Computing semantic similarity of concepts in knowledge graphs. *IEEE Trans. Knowl. Data Eng.*, 29(1):72–85.