Transfer Learning for Semantic Similarity Measures based on Symbolic Regression

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. Recently, transfer learning strategies have become ideal for reusing acquired knowledge through a training phase. The key idea is that reusing such knowledge brings advantages such as increased accuracy and considerable resource savings. In this work, we design a novel strategy for effective and efficient transfer learning in semantic similarity. Our approach is based on generating and transferring optimal models obtained through a symbolic regression process being able to stack evaluation scores from several fundamental techniques. After an exhaustive empirical study, the results lead to high accuracy in addition to significant savings in terms of training time consumed in most of the scenarios considered.

Keywords: Knowledge Engineering, Transfer Learning, Semantic Textual Similarity

1. Introduction

The research community has long recognized that transferring knowledge acquired through a learning phase is critical for intelligence. Furthermore, although there are many forms of learning, such as learning by analogy, case-based learning, and domain adaption, it seems clear that appropriately transferring such accumulated experience, irrespective of the acquisition method, is an efficient approach to obtaining favorable results in the real world.

For this reason, it seems reasonable to devote research efforts to designing novel Transfer Learning (TL) techniques that can benefit from the latest advances to accumulate and transfer the knowledge generated [40]. TL is about how computer systems can reuse the knowledge acquired to face future scenarios of a similar nature.

TL is a crucial area of application and practice nowadays since knowledge adaption is one of the most tangible forms of implementing intelligent solutions [34]. We can see this clearly in our daily lives since learning from small data explains how systems can leverage previous experience and knowledge to face new problems. In this sense, adaption is the fundamental building block that facilitates more innovative and thoughtful solutions.

Remark 1. Motivation. TL methods represent one of the most important concepts related to knowledge representation. These methods are based on the possibility of using a given model to export it to other scenarios of analogous nature. In other words, with TL methods, the knowledge representation learned to complete one task can be generalized to help complete other tasks. A suitable knowledge representation method must determine which factors and features will be exploited and thus reused in another task.

Remark 2. Research Gap. Most traditional machine learning approaches can build models capable of addressing various problems given enough data and

[*] Corresponding author

time. However, the amount of data and time available are frequently limited. For this reason, TL has received considerable attention from Deep Learning and Big Data communities. These communities' problems are very resource-intensive in the form of data and time. This fact explains why strategies of this kind are often seen as a way to alleviate such issues.

As a result, breakthroughs have been achieved primarily through the use of deep learning [41]. However, it is assumed that these methods require a large amount of data, significant processor time, and associated power consumption. Moreover, interpretability, or the degree to which a person can understand the output of the generated models, has received minimal attention.

In this context, research on TL techniques in symbolic regression could be a promising path to explore even though it is currently at a very early stage [32]. Going deeper in this field would be desirable since knowledge representation presents excellent advantages regarding the interpretability and simplicity of the resulting models.

Remark 3. Contributions. Furthermore, the problem of TL has yet to be studied in specific domains, such as semantic similarity, despite all the practical implications that some advances could have in many disciplines, such as information integration, question answering, or machine translation. Therefore, positive results in this regard can be very relevant. The following is a condensed version of the most important contributions that can be drawn from this body of work:

- **C1.** We introduce a novel technique for transfer learning of models trained on datasets of similar nature to the one to be solved. Our approach is based on symbolic regression and brings several advantages over traditional techniques of neural nature, such as improved accuracy, increased interpretability and significant savings in time and power consumption in the training phase.
- **C2.** We perform a complete empirical study that reliably shows the behavior of this approach concerning the most popular semantic similarity datasets, and we establish a comparison about when it is convenient or not to transfer the resulting models.

The remainder of this paper is structured as follows. Section 2 provides state-of-the-art about the existing TL, symbolic regression, and semantic similarity assessment strategies and the solutions proposed to address these challenges. Section 3 introduces our scientific contribution. To the best of our knowledge, this is the first attempt to combine the concepts of TL, symbolic regression, and semantic similarity. Section 4 shows an extensive empirical study on the implementation of our strategy when working with the most popular data sets. Finally, we point out what conclusions can be drawn from this work.

2. Related Work

Automatically identifying the semantic similarity between terms, paragraphs, or even documents is widely acknowledged to be a challenging issue that attempts to address one of the dimensions of the technology that will enable machines to perform tedious and repetitive activities [29]. Because of how relevant it is to industry and academia, this topic has received much attention in recent years. The rationale is that computer systems that can accurately evaluate the degree to which two different pieces of text are similar can open up a window of opportunity to make an impact [25].

However, it is widely assumed that one of the essential practical constraints to advancing systems of this kind is the lack of enough data for training. In order to overcome this limitation, substantial research has been done about TL. The heterogeneous data from the source and target domains can be converted into a similar solution space. Next, we will see more details about the work related to the building blocks needed in our strategy.

2.1. Semantic Similarity

The possibility of automatically measuring the degree of the semantic similarity between textual pieces representing the same concept, although they differ in their lexicography, is a long-standing aspiration of the scientific community [6]. Methods for semantic similarity measurement have had critical applications in many different domains related to natural language understanding [4,12,18,19].

New techniques based on neural embeddings have recently become very popular in this field [30]. However, these techniques for semantic similarity assessment are only partially suitable for performing TL as they depend on how the words are vectorized.

For example, if we look at Deep Learning, the most common strategy is to reuse only a part of the deep neural network. This part usually consists of the first layers since this is where the feature extraction is performed automatically. The last layers can be regenerated since they define the more specific details of the problem. This strategy has proven to be very effective and efficient to date. However, it is still associated with some of the issues of neural-based solutions, such as the need for vast amounts of training data and the need for interpretability [26]. This is because it is impossible to interpret a model that contains hundreds or thousands of nodes that are related to one another. In this way, a human operator can specify which outputs correlate to which inputs, and a deep neural network will automatically design a somewhat accurate mapping function. However, the human operator will not have any way of knowing what is going on within the model because the deep neural network will not reveal this information to them.

For this reason, these models are frequently referred to as black boxes because they conceal the workings of their operations from the people who utilize them. Although a significant quantity of study has been done in recent years to address this issue [24], the problem still needs to be solved. For this reason, we propose to explore a different alternative based on the notion of symbolic regression.

2.2. Semantic Similarity Aggregation

In the context of this study, we compile and strategically organize several existing approaches for semantic similarity measurement. Aggregation methods are standard in many subfields of computing. They are frequently used in production settings because they hide the faults caused by individual methods by grouping them with other methods that are generally reliable [28]. Aggregation approaches lose their usefulness only in the improbable event that all methods concurrently commit the same error.

The arithmetic means, the median, the geometric, and the harmonic means are the four types of aggregation operators that are used most frequently. However, their aggregation technique could be more realistic [5]. This almost always indicates that the techniques in question do not produce the best results. As a result, researchers tend to explore more efficient operators capable of depicting a good interaction between the present variables.

2.3. Symbolic Regression

We use symbolic regression based on Genetic Programming introduced by Koza [17]. GP is a technique that generates and optimizes programs to solve a particular task using genetic operators. The tree representation is usually used in GP to represent a solution, making this technique versatile and allowing the models to be extended since more functions could be added. Moreover, a human operator can easily understand the behavior of the GP-derived model since solutions consist of a computer program that best solves a given problem.

Furthermore, even though TL has become highly prevalent in machine learning, such techniques as evolutionary transfer learning based on GP and their applications have yet to be deeply explored. Just some works like O'Neill et al. [33] suggested a TL approach for GP that considers the similarity between alternative solutions that have been developed in multiple situations, or Iqbal et al. [15] used subtrees learned on the source domain to help GP perform better on similar target tasks.

Symbolic regression is the term that groups a family of strategies that can learn directly from data and create both the structure and variables of regression models simultaneously [3]. The interpretability of symbolic regression is one of its significant advantages since the resulting model is expressed in function form. It gives domain experts a valuable understanding of the underlying data generation process and highlights the main characteristics [32].

2.4. Transfer Learning

The idea is to use the knowledge gained from working with the source model in the past to apply it to the target task [9,39,10]. According to the literature, there are up to four different types of TL:

- Instance-based methods, whereby the transference corresponds to the source instances
- Feature-based methods, whereby the transference corresponds to the standard features in the source and target domains
- Model-based methods, whereby the transference is (part of) the source model
- Relation-based methods, whereby the transference is about the relations in the source domain

In this work, we are going to focus on the use of Model-based methods. Most approaches from this branch consider that the source and the target domains have some parameters in common. The rationale behind these approaches is that a model that has been properly trained should be able to successfully model enough domain knowledge that it can be reused in a different scenario. Thus, it is usually assumed that one works with domain-invariant models.

Some of the most popular TL approaches in the field of semantic similarity are BERT [7] and ELMO [35]. However, they are undertaking a brute force approach since their successful condition for TL is that they work with vast amounts of source domain training data. Unfortunately, the abundance of data is not the most common actual situation, so exploring other alternative approaches is necessary. From now on, we will explain an effective technique for TL based on symbolic regression.

2.5. *Novelty of our Approach*

The challenge of TL in symbolic regression has received little attention. However, the fundamental idea that a symbolic expression that has been computed to optimize fitness in a given problem is hardly reusable in other problems has been disproved in some works.

TL can generate solutions based on Abstract Syntax Trees (ASTs) that, in many cases, outperform baseline methods in classical classification and regression tasks. Because of the symbolic nature of GP approaches and their versatile representation, GP is an excellent option for working around the notion of AST [42]. There have already been several beneficial uses of GP in this context [27].

GP would find an AST that produces optimal semantic similarity measures by tackling the problem at a higher abstraction level. Moreover, symbolic regression has recently attracted much attention because its application brings a high degree of interpretability in the form of functional interpretability [32]. Furthermore, interpretability is an inherent characteristic of symbolic regression and genetic programming [1]. Several studies have demonstrated how symbolic regression can enhance the understanding of the resulting models [8,13].

We present our approach to transferring learning using symbolic regression to raise semantic similarity measures. Our strategy is based on the **FullTree** method from [8] that assumes the best tree should be transferred. We aim to use that model resulting from the source tasks to enhance the performance of learning methods in the target tasks.

3. Transfer Learning for Semantic Similarity Measures using Symbolic Regression

As this is a supervised machine learning problem, let X be the input space and let Y be the output space. Our goal is to learn a model $m : X \rightarrow Y$ to assign a label from Y to a sample from X. This model is learned from a learning set $S = \{(x_i, y_i) \in (X \times Y)\}_{i=1}^{n}$.

In our specific case, we assume that the samples $(x_i, y_i) \in S$ are drawn from a distribution D_S of support $X \times Y$. Our goal is to learn m such that it commits the minimum error possible for labeling new items coming from the distribution D_S.

3.1. *Transfer Learning*

A domain $\mathcal{D}$ consists of a *feature space* $\mathcal{X}$ and a *marginal probability distribution* $P(X)$, where $X = \{x_1, ..., x_n\} \in \mathcal{X}$. Given a specific $\mathcal{D} = \{\mathcal{X}, P(X)\}$, a task consists of two parts, i.e., a label space $\mathcal{Y}$ and a function $f : \mathcal{X} \rightarrow \mathcal{Y}$. The function f is used to predict the corresponding label $f(x)$ of a new item x. At the same time, a function $\mathcal{T} = \{\mathcal{Y}, f(x)\}$ is learned from the training consisting of pairs $\{x_i, y_i\}$, where $x_i \in X$ and $y_i \in \mathcal{Y}$.

Furthermore, given a source $\mathcal{D}_S$ and a task $\mathcal{T}_S$, a target $\mathcal{D}_T$ and a task $\mathcal{T}_T$, whereby $\mathcal{D}_S \neq \mathcal{D}_T$, or $\mathcal{T}_S \neq \mathcal{T}_T$, TL aims to help improve the learning of the target function $f_T(\cdot)$ in $\mathcal{T}_T$ using the accumulated knowledge in $\mathcal{D}_S$ and $\mathcal{T}_S$.

3.2. *Symbolic Regression*

In this work, we address this problem from the GP point of view. The idea behind GP is to replicate the principle of natural selection to create specific individuals capable of solving a given problem [2]. In the particular case of symbolic regression, the evolutionary strategies are aimed at the automatic design of computer programs, whether they are functions or algorithms.

Most regression models propose that Y_i is a function of X_i and β, with e_i representing errors and residuals that may stand in for unknown determinants of Y_i or random statistical noise:

$Y_i = f(X_i, \beta) + e_i$

The researchers aim to estimate the function $f(X_i, \beta)$ that most closely fits the input data. In order to carry out regression analysis, the form of the function f must be properly assessed. This kind of strategy can be seen as a data-driven method mainly because the initial population of individuals representing solutions evolves guided towards the goals of maximizing fitness. In addition, the approach will select the most promising operators and variables since no condition imposes that all of them must be used.

3.3. The learning process

The learning process is driven by an evolutionary approach which allows the model to evolve towards fitness maximization. The aim is to find the best individual leading to a solution while avoiding over-fitting, for which a cross-validation process is maintained to ensure that the model generalizes well.

Therefore, we have chosen a classical approach for an elitist evolutionary strategy that considers mutation for an automatic exploration of the solution space while allowing a mechanism to escape from the optimum local problem).

Although many other learning techniques of both evolutionary and other nature could be studied to proceed with the evolution of the models towards an optimal AST, this requires considerable attention and is outside the scope of this work. Nevertheless, it would remain pending as an interesting line of future research.

3.4. Training, Regularization, and Bloat Control

The semantic similarity measures are combined with operators and constant numbers searching to build the objective function. The underlying AST can evolve thanks to the evolutionary strategy we referred to, as explained in [17]. The result is calculated by evaluating each node and performing the parent node operation on the child nodes.

Traditional methods like cross-validation handle over-fitting work well with a few inputs, but these techniques usually need an alternative when dealing with a large set of inputs. This issue is often tackled using a methodology known as regularization, which reduces the complexity of models while allowing for potential adjustments.

Figure 1 shows what the ideal TL process would look like. By benefiting from a model that is assumed to be good, we start with a population of high-quality ASTs from the beginning. Therefore the first value is much better, and the training process is much faster than the process without TL. This way, the maximum value is reached much earlier, thus saving time and associated variables such as power consumption. These savings can be especially significant when the problem is sufficiently complex, as shown in Section 4.4.

Last but not least, bloat is one of the most problematic phenomena in GP. A rise in the average program size without a matching gain in fitness is the definition of this phenomenon. In our particular case, it is not desirable since it could cause our already small AST to fall into a spiral that would not allow it to evolve. Therefore, in this work, we use the classical method of limiting the maximal allowed depth just as described in [23].

4. Results

The idea is to determine the quality of our models when trained on a dataset and used to solve other datasets of similar nature. Therefore, we have divided this section: first, we describe the datasets we will work with; second, we explain the evaluation criteria commonly used in semantic similarity measurement. Then, we specify the configuration we have used in this empirical study. Finally, we show the results obtained and provide an analysis.

4.1. Datasets

We will use the four most popular benchmark datasets to conduct our experiments. Firstly, we will use two well-known general-purpose datasets to give us an idea of how the method works. The first one is the Miller & Charles [31] that consists of 60 words from daily life and Rubenstein & Goodenough [38] that extends the previous. Some samples from these datasets are: (gem-jewel, 0.98), (car-automobile, 0.98), (rooster-voyage, 0.04), (chord-smile, 0.02).

In addition, WS353 [11] contains 353 pairs of words on different human subjects, for example (tiger-cat, 1.0) or (computer-keyboard, 0.75). Unlike many published studies, we work here with the full version of the dataset. Finally, Simlex-665 [14] containing 665 general-purpose entries to evaluate the quality of new methods for similarity assessment (actress-actor, 0.712), (teacher-instructor, 0.925), etc.

We aim to find a numerical score in the real interval $[\alpha, \omega]$ that states the similarity between the different entities to be compared. In this way, values close to α mean that the two entities are not similar, and values close to ω mean that the two entities are practically synonymous. This characteristic makes the automatic methods that try to face the problem be evaluated

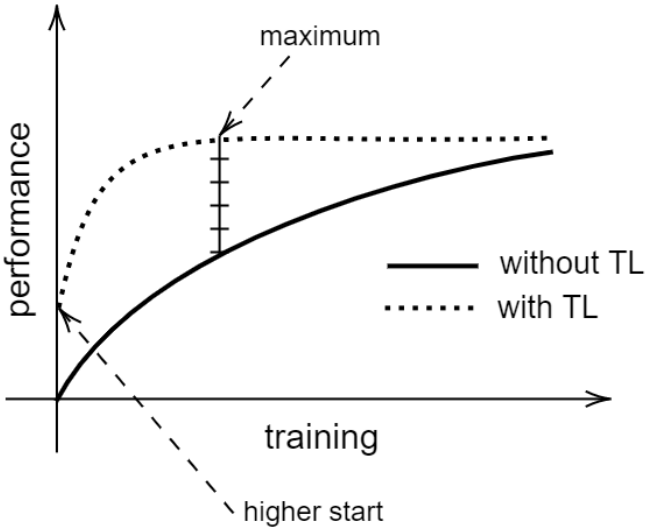

Fig. 1. Example of an ideal TL scheme where a higher start allows reaching the maximum training value earlier and brings associated advantages such as lower time and electric power consumption

through correlation measures between the scores provided by the human judgment and the scores provided through the machine.

4.2. Evaluation Criteria

One challenge when determining the likeness between the two entities being compared, possibly applying a threshold above which they are considered equivalent later. This problem is usually evaluated based on the correlation between the the cases solved by the experts and the values returned by the machine. This computation is not only helpful in guiding our evolutionary process, but it is also vital for the fitness function.

The Pearson Correlation Coefficient and the Spearman Rank correlation are two methods that may be utilized to analyze the methods that are used to find comparable links utilizing the notion of correlation. Pearson's correlation coefficient can measure the degree of correlation between the human judgment and the results produced by the machine; more specifically, the Pearson Correlation Coefficient is calculated between two numerical arrays: the truth and the solution's output. The Spearman Rank correlation is the other option available. When the findings are to be compared on an ordinal scale, the Spearman Rank correlation test is the analysis that should be used. The following is how it is computed:

In our study, we consider both approaches since we do not strongly prefer one correlation coefficient over the other. In addition to that, this makes our investigation more comprehensive.

4.3. Experimental Setup

Determining the optimal configuration of our strategy requires a grid search to find the best parameters to work with. In practice, this search space is so huge that we have had to narrow down the intervals. Below we can see the parameters for the symbolic regression strategies in addition to the intervals in which we have delimited our grid search:

- Operator set $\{+, -, \cdot, \div, exp, max, min\}$
- Individual's length [0 - 50]: **22**
- Maximum tree depth [2 - 5]: 4
- Opt. of operator precedence [Yes, No]: No
- Constants [0 - 5]: **3**
- Data representation (binary, real): **real**
- Population size [10 - 100]: **25**
- Crossover probability [0.3 - 0.95]: **0.71**
- Mutation rate [0.01 - 0.3]: **0.15**
- Stop condition [1,000 - 50,000]: **20,000**

Just as a small remark, it is necessary to comment that the division operator is protected against division by 0, i.e., in the case that the denominator of such operation is a 0, it will be considered that this individual is not valid so that the execution flow of our strategy does not collapse.

Regarding the training phase, we have chosen a 5-cross-fold validation so that in each test we train with

80% of the sample and test on the remaining 20%. Moreover, we do this five times to solve the dataset completely.

Furthermore, we rely on one of the most popular general-purpose sources to implement the methods that populate the leaves of the ASTs, i.e., WordNet[1] which is a knowledge base that attempts to model synonymic relationships as well as sub summary relationships between different concepts. Several methods for calculating similarity based on the different paths can be calculated in such a taxonomy. These methods are Path (path) [36], Leacock (lch) [20], Wu & Palmer (wup) [43], Li (li) [21], Resnik (res) [37], Lin (lin) [22], Jiang & Conrad (jcn) [16], and wpath [44]). Our concept here is to aggregate all these methods strategically so that a) we can reach higher levels of accuracy b) any person could take a look at WordNet and the way the distance between the different concepts is calculated to realize where the final semantic similarity value comes from. To meet both objectives simultaneously, we will need small ASTs. This is mainly because simple models behave better in terms of generalization of solutions than complex ones.

4.4. Empirical results

This section will detail the results obtained in our empirical evaluation using the most popular benchmark datasets. The results reported for our approach are based on 30 independent executions due to the non-deterministic nature of the methods. So we report the average value achieved (the average result obtained after 30 independent runs). Please note that although our methods are prepared to iterate up to 20,000 times, we will only plot up to 2,000 iterations. The reason is that the most interesting part of the methods happens in this interval. We often look for successful mutations to circumvent the optimum local problem, but the information obtained is usually plain and not that interesting in most situations.

4.4.1. Transfer from MC30 to the others

Our first experiments transfer the AST obtained by training MC30 to the rest of the semantic similarity datasets. Figure 2 shows the obtained results. As it can be seen, the valuable fact of starting the process with a higher accuracy value due to the use of a model that already worked well in a scenario of a similar nature is more or less fulfilled. This allows the highest value to be reached much earlier, leading to considerable savings in learning time (and associated advantages such as power consumption).

As a general rule, and in line with our initial hypothesis, it can be observed that models with TL can reach the maximum accuracy value much faster than those that must start from scratch. In some cases, transferring the best AST does not achieve much better results than a cold start (e.g., Pearson correlation in the MC30 transfer to WS353), but even in such cases, transferring the knowledge does not have a negative effect.

4.4.2. Transfer from RG65 to the others

The second of our experiments consists of transferring the AST obtained by training RG65 to the rest of the problems. Figure 3 shows the different results. This time, we have used the models obtained by training with RG65 to solve the rest of the benchmark datasets. As can be seen, this has the associated advantage of starting with higher initial values and a higher speed of convergence to the maximum as expected.

Transferring the best solution has significant advantages in the form of time savings. Indeed, there are cases when such transference does not give the expected results, but even in such a case, the results do not differ from a cold start with random numbers in the long run.

4.4.3. Transfer from WS353 to the others

The third of our experiments is transferring the best AST obtained by training WS353 to the rest of the problems. Figure 4 shows the different results that we have achieved. Please note that in the first two cases, knowledge is being transferred to much smaller datasets, while in the third case, it is being transferred to a much larger task. This does not prevent the good behavior of the initial solutions transferred.

It can be observed that as one tries to transfer knowledge that has been learned using massive datasets, the effectiveness of such transfer decreases. It seems that, in our particular case, the generalization obtained through a model learned for a large data volume does not apply to the specific set of small data.

4.4.4. Transfer from Simlex665 to the others

Our last experiments consist of transferring the AST obtained by training Simlex665 to the rest of the problems. Figure 5 shows the different results that we have achieved for this last experiment. In this case, being Simlex665 the largest benchmark dataset in our study, ASTs learned in a larger task are being transferred to

[1] https://wordnet.princeton.edu/

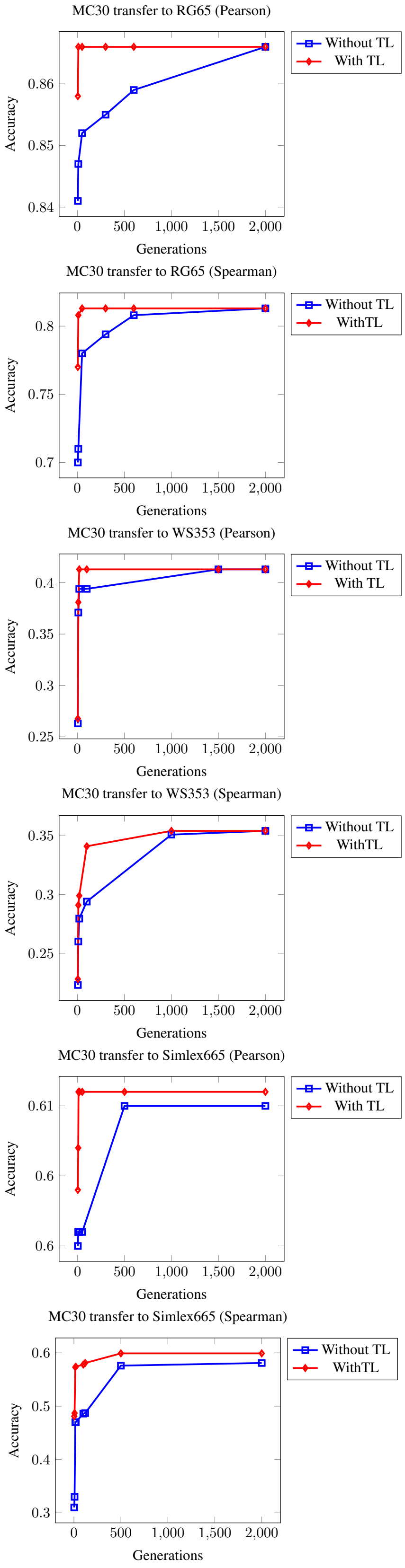

Fig. 2. Summary of results obtained when the model to be transferred has been generated in the MC30 training phase

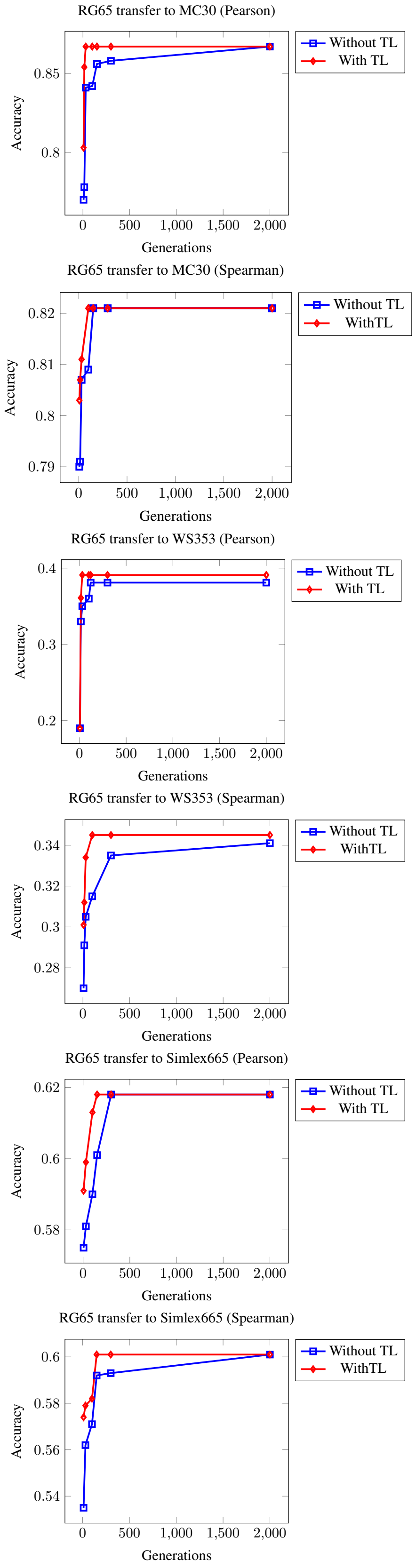

Fig. 3. Summary of results obtained when the model to be transferred has been generated in the RG65 training phase

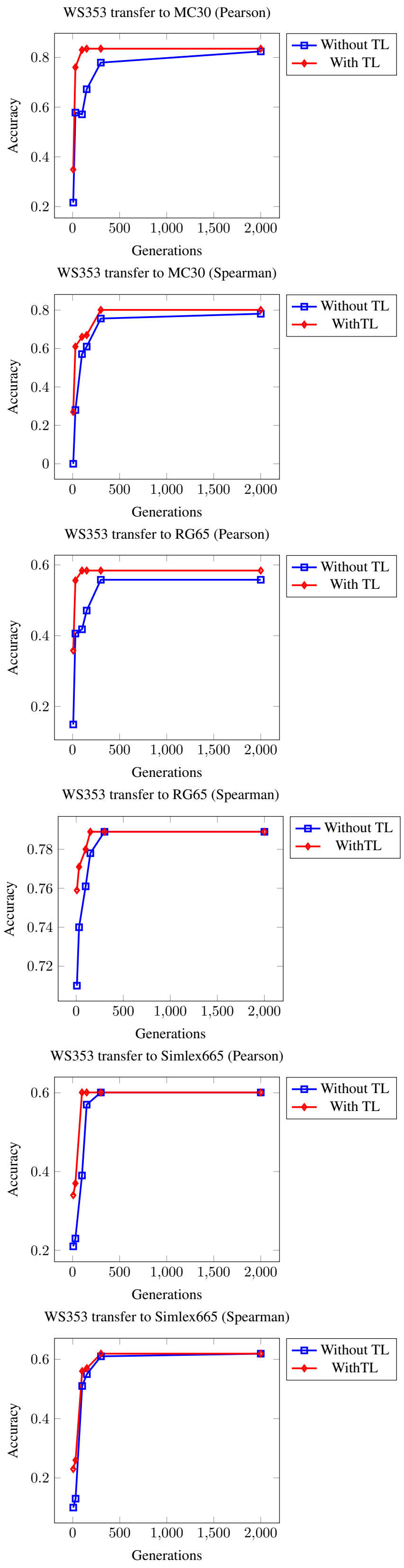

Fig. 4. Summary of results obtained when the model to be transferred has been generated in the WS353 training phase

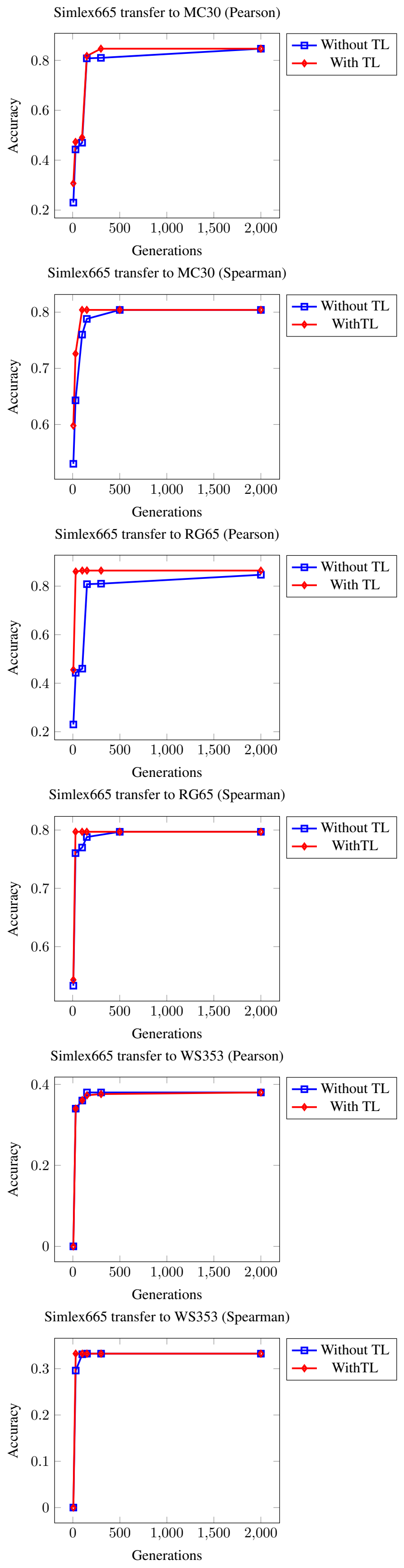

Fig. 5. Summary of results obtained when the model to be transferred has been generated in the Simlex665 training

	MC30		**RG65**		**WS353**		**Simlex665**	
Method	σ	ρ	σ	ρ	σ	ρ	σ	ρ
path	0.755	0.745	0.784	0.783	0.340	0.314	0.502	0.584
lch	0.787	0.744	0.841	0.783	0.349	0.314	0.599	0.584
wup	0.759	0.747	0.777	0.758	0.361	0.348	0.601	0.542
li	0.802	0.734	0.857	0.787	0.340	0.337	0.593	0.586
res	0.807	0.734	0.833	0.749	0.385	0.347	0.549	0.535
lin	0.829	0.766	0.855	0.762	0.374	0.310	0.549	0.582
jcn	0.665	**0.833**	0.719	0.770	0.302	0.292	0.539	0.579
wpath	0.836	0.747	0.873	0.788	0.349	0.349	0.614	0.603
TL1	-	-	0.866	**0.813**	**0.413**	**0.354**	0.611	0.599
TL2	**0.867**	0.821	-	-	0.391	0.345	**0.618**	0.601
TL3	0.835	0.801	0.784	0.789	-	-	0.601	**0.619**
TL4	0.847	0.804	**0.874**	0.789	0.380	0.332	-	-

Table 1

Results (test) obtained for all the benchmark datasets. Transfer Learning solutions obtain better results in most cases

solve much smaller tasks. This leads to lower results than those obtained in the previous experiment.

With all the experiments performed, we have covered all possible transfer cases from small to large problems and vice versa. TL techniques benefit less training time in most cases and do not hurt in the remaining cases. So their use may make sense in a wide range of situations. However, we will now proceed to a detailed study of this phenomenon.

Table 1 shows the maximum results achieved with the semantic similarity measures and all benchmark datasets considered in this work. As is evident, and as we have seen in Table 1, the results our resulting models will always be superior since we can aggregate them.

Please note that while it is true that we could use much more advanced and recent semantic similarity measures, such as those based on word embeddings [30] or transformers [7], we do not wish to do so since it would be detrimental to the interpretability. Therefore, our selection only comprises widely accepted measures as reasonably interpretable. Moreover, this work does not claim to obtain the best results in semantic similarity assessment but a sound strategy to save time in different training processes.

4.5. Discussion

Assessing semantic similarity across textual pieces is commonly an issue impacting various computer-related disciplines. Research in this direction has served as the foundation for many computer-related disciplines such as data integration, information retrieval or query expansion. Nevertheless, there may be times when the best idea is not to develop a strategy from scratch but to design a system that makes it possible to aggregate the different methods already in place effectively. For example, many similar experiences already exist in word embeddings, where recent research can demonstrate that the linear combination of existing methods can surpass the state-of-the-art [19].

The challenge of training solutions with small data sets is a severe problem many industries face today. TL can be an excellent solution to this challenge since it uses resources efficiently by reusing data and existing models. Therefore, our novel approach can result in benefits such as solutions with high accuracy, interpretability, and performance rates. Also, the resulting model is immediately exportable in the form of a function to many programming languages, which facilitates its understanding by a human operator. As a result, we have a strategy that positively impacts a wide range of industry sectors.

Furthermore, we could deduce that our proposal can perform well in most experiments from the results obtained. It is also clear that determining when there may be an opportunity for a positive transfer is far from trivial, and there is no systematic way to find out yet. We have seen how systems can expand their application range outside their initial conception by transferring information from one domain. This generalization makes it more available and robust in many situations where expertise or resources, such as computational power, data, and hardware, are limited. Therefore, using TL in this context has the following advantages:

- Improved baseline accuracy since it is possible to boost the baseline precision of a given method by supplementing its basic information with knowledge from a source model that was used to solve a similar problem.
- Interpretability of the resulting model that moves away from black box solutions to build a fully functional model that can be understood by the human operators who will make use of it.
- Time to develop a model: Using information from a source model aids in thoroughly learning the target task instead of building it from scratch. This usually improves the models' training time, especially in cases requiring vast computational resources to complete the training phase.

Moreover, our strategy can bring the advantages of TL in most cases, and even in situations where it is not possible, its application does not penalize cold start results, so it makes sense to consider strategies of this type.

5. Conclusions

We have presented our approach to TL using symbolic regression to solve the challenge of automatically measuring semantic similarity. Our approach uses functional models that are effective and efficient when reused in problems of analogical nature. Our approach has several advantages over classical neural solutions, including less training time required to reach the maximum value, resulting in additional benefits such as reduced power consumption, which is relevant when dealing with substantial volume problems. In addition, the resulting models are more interpretable since they are expressed in a functional form. This means that the model can be understood by anyone who can read a mathematical expression and easily recalibrated with a mathematical sub-expression that allows it to adapt to the new situation at a meager cost.

One of the lessons to be learned from this work is that although most of the methods in the semantic similarity field have historically been built to work in isolation, the knowledge generated when training them can be successfully reused. Even if these methods might be programmed to solve particular problems, they can profit from the resulting models more efficiently when applied in different scenarios through model-based TL techniques. Therefore, we have seen that TL is an appropriate strategy for overcoming the view of the isolated learning model, which has traditionally prevailed.

Acknowledgments

The authors thank the anonymous reviewers for their comments and suggestions to improve the work. This research has been partly funded by the Federal Ministry for Climate Action, Environment, Energy, Mobility, Innovation and Technology (BMK), the Federal Ministry for Labour and Economy (BMAW), and the State of Upper Austria in the frame of the SCCH competence center INTEGRATE (FFG grant no. 892418) in the COMET - Competence Centers for Excellent Technologies Programme managed by Austrian Research Promotion Agency FFG.

References

[1] Adadi, A. and Berrada, M. (2018). Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access*, 6:52138–52160.

[2] Affenzeller, M., Winkler, S. M., Kronberger, G., Kommenda, M., Burlacu, B., and 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].*, pages 175–190.

[3] 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.

[4] Bollegala, D., Matsuo, Y., and Ishizuka, M. (2011). A web search engine-based approach to measure semantic similarity between words. *IEEE Trans. Knowl. Data Eng.*, 23(7):977–990.

[5] Chaves-Gonzalez, J. M. and 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.

[6] Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., and Harshman, R. A. (1990). Indexing by latent semantic analysis. *J. Am. Soc. Inf. Sci.*, 41(6):391–407.

[7] 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.

[8] Dinh, T. T. H., Chu, T. H., and Nguyen, Q. U. (2015). Transfer learning in genetic programming. In *IEEE Congress on Evolutionary Computation, CEC 2015, Sendai, Japan, May 25-28, 2015*, pages 1145–1151. IEEE.

[9] Elbaz, K., Shen, S.-L., Zhou, A., Yin, Z.-Y., and Lyu, H.-M. (2021). Prediction of disc cutter life during shield tunneling with ai via the incorporation of a genetic algorithm into a gmdh-type neural network. *Engineering*, 7(2):238–251.

[10] Elbaz, K., Yan, T., Zhou, A., and Shen, S.-L. (2022). Deep learning analysis for energy consumption of shield tunneling machine drive system. *Tunnelling and Underground Space Technology*, 123:104405.

[11] Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., and Ruppin, E. (2002). Placing search in context: the concept revisited. *ACM Trans. Inf. Syst.*, 20(1):116–131.

[12] 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.

[13] 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.

[14] Hill, F., Reichart, R., and Korhonen, A. (2015). Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Comput. Linguistics*, 41(4):665–695.

[15] 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.

[16] Jiang, J. J. and 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*, pages 19–33.

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

[18] 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.

[19] 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.

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

[21] Li, Y., Bandar, Z., and McLean, D. (2003). An approach for measuring semantic similarity between words using multiple information sources. *IEEE Trans. Knowl. Data Eng.*, 15(4):871–882.

[22] 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.

[23] Luke, S. and Panait, L. (2006). A comparison of bloat control methods for genetic programming. *Evol. Comput.*, 14(3):309–344.

[24] Lundberg, S. M. and Lee, S. (2017). A unified approach to interpreting model predictions. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4765–4774.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] Martinez-Gil, J. and Chaves-Gonzalez, J. M. (2022). Sustainable semantic similarity assessment. *Journal of Intelligent & Fuzzy Systems*, 43(5):6163–6174.

[30] 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.

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

[32] Muñoz, L., Trujillo, L., and Silva, S. (2020). Transfer learning in constructive induction with genetic programming. *Genet. Program. Evolvable Mach.*, 21(4):529–569.

[33] O'Neill, D., Al-Sahaf, H., Xue, B., and Zhang, M. (2017). Common subtrees in related problems: A novel transfer learning approach for genetic programming. In *2017 IEEE Congress on Evolutionary Computation, CEC 2017, Donostia, San Sebastián, Spain, June 5-8, 2017*, pages 1287–1294. IEEE.

[34] Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.*, 22(10):1345–1359.

[35] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In Walker, M. A., Ji, H., and Stent, A., editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics.

[36] Rada, R., Mili, H., Bicknell, E., and Blettner, M. (1989). Development and application of a metric on semantic nets. *IEEE Trans. Syst. Man Cybern.*, 19(1):17–30.

[37] 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.

[38] Rubenstein, H. and Goodenough, J. B. (1965). Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633.

[39] Shen, S.-L., Elbaz, K., Shaban, W. M., and Zhou, A. (2022). Real-time prediction of shield moving trajectory during tunnelling. *Acta Geotechnica*, 17(4):1533–1549.

[40] Su, S., Li, W., Mou, J., Garg, A., Gao, L., and Liu, J. (2022). A hybrid battery equivalent circuit model, deep learning, and transfer learning for battery state monitoring. *IEEE Transactions on Transportation Electrification*.

[41] Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., and Liu, C. (2018). A survey on deep transfer learning. In Kurková, V., Manolopoulos, Y., Hammer, B., Iliadis, L. S., and Maglogiannis, I., editors, *Artificial Neural Networks and Machine Learning - ICANN 2018 - 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part III*, volume 11141 of *Lecture Notes in Computer Science*, pages 270–279. Springer.

[42] Vladislavleva, E., Smits, G., and den Hertog, D. (2010). On the importance of data balancing for symbolic regression. *IEEE Trans. Evolutionary Computation*, 14(2):252–277.

[43] Wu, Z. and Palmer, M. S. (1994). Verb semantics and lexical selection. In Pustejovsky, J., editor, *32nd Annual Meeting of the Association for Computational Linguistics, 27-30 June 1994, New Mexico State University, Las Cruces, New Mexico, USA, Proceedings*, pages 133–138. Morgan Kaufmann Publishers / ACL.

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