AutoML strategy based on grammatical evolution: A case study about knowledge discovery from text

Suilan Estevez-Velarde[1], **Yoan Gutiérrez**[2], **Andrés Montoyo**[3], and **Yudivián Almeida-Cruz**[1]

[1]School of Math and Computer Science, University of Havana, Cuba
{sestevez,yudy}@matcom.uh.cu
[2]University Institute for Computing Research (IUII), University of Alicante, Spain
[3]Department of Languages and Computing Systems, University of Alicante, Spain
{ygutierrez,montoyo}@dlsi.ua.es

Abstract

The process of extracting knowledge from natural language text poses a complex problem that requires both a combination of machine learning techniques and proper feature selection. Recent advances in Automatic Machine Learning (AutoML) provide effective tools to explore large sets of algorithms, hyperparameters and features to find out the most suitable combination of them. This paper proposes a novel AutoML strategy based on probabilistic grammatical evolution, which is evaluated on the health domain by facing the knowledge discovery challenge in Spanish text documents. Our approach achieves state-of-the-art results and provides interesting insights into the best combination of parameters and algorithms to use when dealing with this challenge. Source code is provided for the research community.

1 Introduction

In recent years there has been a large increase in the amount of technical documents produced by the scientific community. These documents form part of a large corpora of knowledge (e.g., research papers and encyclopedias) mostly accessible through text-based search engines. However, to better exploit this knowledge (i.e., not just recovering text fragments or URLs) in automated processes, it is necessary to process the text and extract the pieces of information in a semantic format useful for machine analysis. The challenge of automatically extracting useful knowledge from text sources is studied by research areas such as knowledge discovery (Gonzalez et al., 2015), ontology learning (Cimiano et al., 2009) and learning by reading (Barker et al., 2007).

One of the most useful representation for discovering knowledge in natural language sentences is using Subject-Verb-Object triplets (Estevez-Velarde et al., 2018). This structure is ubiquitous in many human languages (Crystal, 1997), and as such, it has been used as the base for knowledge extraction in several systems (Mitchell et al., 2018). In a similar line, Giunchiglia and Fumagalli (2017) propose the use of objects, actions and functions as main components for representing common knowledge. Additionally, other relations with a specific meaning are often used in ontologies to represent taxonomies (i.e., *is-a*) or composition (i.e., *part-of*). Hence, combining objects and actions with a small set of specific semantic relations allows the representation of a broad range of domains with a simple computational structure.

Specifically in the health domain, research in knowledge discovery techniques has steadily increased motivated by the potential impact in the quality of human life. Even though the amount of medical knowledge published in natural language is considerable, it is still an open challenge the design of computational systems that can make effective use of this knowledge, for example, to improve diagnosis and aid in medical decision making (Gonzalez et al., 2015). In this context, the shared campaign TASS 2018 eHealth-KD (Martínez-Cámara et al., 2018) proposes a new evaluation scenario where health documents in Spanish language must be processed to extract Subject-Action-Target triplets along with other semantic relations. This scenario presents a general-purpose semantic structure, hence, systems designed for the automatic extraction of knowledge are potentially reusable in multiple domains.

Different solutions presented in the challenge show the complexity of this task (Piad-Morffis et al., 2019b). The approaches of this challenge used a variety of algorithms (e.g., neural networks, natural language processing, machine learning, etc.) and different strategies to face the challenge,

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4356–4365
Florence, Italy, July 28 - August 2, 2019.

such as combining different subtasks. Besides, each algorithm implemented involves parameters that need to be manually tuned to find their optimal values, which increases the experimentation time considerably. This extensive experimentation could be better handled by means of Automated Machine Learning (AutoML) strategies.

AutoML is a recent strategy used for the automatic selection of the best combinations of algorithmic pipelines. The AutoML process is based on the definition of a solution space where all the possible pipelines are represented, and a optimization process to explore this space. The optimization process is typically implemented using two common approaches: Bayesian (Hutter et al., 2019) and Evolutionary Optimization (Chen et al., 2018). Some of the most popular examples of AutoML techniques based on Bayesian Optimization are Auto-Weka (Thornton et al., 2013), Auto-Sklearn (Feurer et al., 2015) and Hyperopt-Sklearn (Komer et al., 2014). Among the Evolutionary Optimization approaches, two of the most relevant are TPOT (Olson and Moore, 2016) and RECIPE (de Sá et al., 2017). In case of neural networks, a common approach is Neural Architecture Search (Zoph and Le, 2016) (NAS), were frameworks such as Auto-Keras are used (Jin et al., 2018). NAS methods can be based both on Bayesian and Evolutionary Optimization.

Current AutoML approaches are oriented towards dealing with black-box classification or regression problems. In the eHealth-KD challenge context, the selection of which algorithms and hyper-parameters to use for each subtask can be framed as a classic AutoML problem. However, there are additional high-level decisions, such as whether to solve subtasks sequentially or combined, that cannot be easily represented in traditional AutoML frameworks. This research proposes a novel AutoML strategy based on probabilistic grammatical evolution (Kim and Ahn, 2015) designed for knowledge discovery from text. Our approach performs an intelligent search among several possible pipelines, incrementally learning the characteristics that produce the best performance. As a case study, we apply our proposal to the eHealth-KD challenge, and achieve state-of-the-art results in one of the evaluation scenarios proposed. However, this approach can be extended to other machine learning scenarios. Furthermore, the source code of our proposal, with the example application to the eHealth-KD challenge and additional examples in several different problems is provided for the research community[1].

This paper is organized as follows. Section 2 presents a brief description of the eHealth-KD challenge. Section 3 describes the main approaches submitted for the eHealth-KD challenge in the TASS 2018 workshop. Section 4 introduces the main contribution of the research, which is an automatic optimization procedure for dealing with the eHealth Knowledge Discovery challenge. Section 5 presents the main experimental results obtained in comparison with relevant alternatives. Finally, Section 6 presents an in-depth analysis of the optimization process and interesting insights, and Section 7 presents the main conclusions and future lines of research.

2 Challenge description

The eHealth-KD challenge represents a knowledge discovery task in the domain of health-related documents written in Spanish language, as part of the TASS 2018 workshop (Martínez-Cámara et al., 2018). The overall purpose of the task is the identification of two types of textual entities (*concepts* and *actions*) and six semantic relations among them (*subject*, *target*, *is-a*, *same-as*, *property-of* and *part-of*). Figure 1 shows a visual representation of these elements and its semantic annotation model in a set of example sentences. The task is subdivided into three subtasks, namely, the identification of relevant key phrases, the classification of these into concepts or actions, and the identification of the corresponding relations.

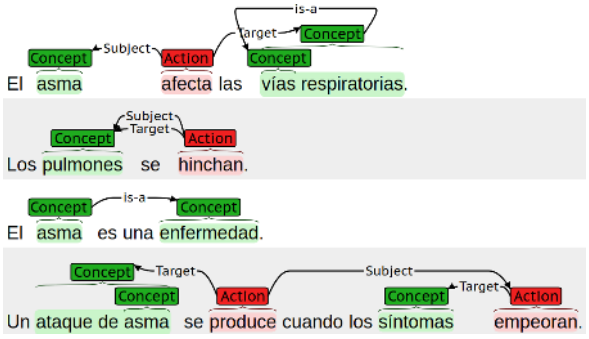

Figure 1: Example of the TASS 2018 Task 3 challenge, taken from Martínez-Cámara et al. (2018) and modified to fit. This represents an example written in Spanish language

[1] https://github.com/knowledge-learning/hp-optimization

According to this subdivision in subtasks, a standard F_1 metric is used to evaluate the overall performance of an extraction system. This metric is defined as a micro-average between correct, incorrect, missing and spurious key phrases, their classes, and the relations between them. Furthermore, the challenge defines three evaluation scenarios in which different subtasks are considered. However, for the purpose of this research, we focus on Scenario 1, which considers all three subtasks and is thus the most complete and challenging.

3 Related works

A total of 6 researchers submitted their approaches in the eHealth-KD challenge, of which 5 evaluate in Scenario 1. However, one of these participants did not submit a description paper, hence, it will not be considered in this research. This section briefly describes the main characteristics of the systems designed by the remaining 5 researchers.

In terms of representation, most of the approaches presented rely on natural language processing features. Medina and Turmo (2018) and Palatresi and Hontoria (2018) employ *Freeling* to extract syntactic and morphological features. Zavala et al. (2018) trains a BI-LSTM model to encode the morphological and syntactic features for later use with a shallow classifier. These approaches also rely on word embeddings (either *Glove* or *Word2Vec*) as additional semantic features. López-Ubeda et al. (2018), on the other hand, uses a custom entity detector also based on syntactic and morphological features.

With respect to machine learning, two approaches use deep learning techniques, specifically convolutional neural networks (Medina and Turmo, 2018; Suarez-Paniagua et al., 2018). In contrast, other approaches apply a CRF classifier for solving subtasks A and B (Palatresi and Hontoria, 2018; Zavala et al., 2018), and shallow classifiers (*logistic regression*) for solving subtask C. Finally, López-Ubeda et al. (2018) uses hand-crafted rules with syntactic and knowledge-based characteristics for subtask B.

Even though the challenge is originally defined as a sequence of three subtasks, several approaches opted for combining subtasks, recognizing important correlations between them. For example, Zavala et al. (2018) and Palatresi and Hontoria (2018) frame subtask A and B as an entity tagging problem, and apply standard sequence-based models. In contrast, Medina and Turmo (2018) solves simultaneously subtasks B and C, noting that there is a high correlation between entity classes and the possible relations.

Given this variety of approaches, it is interesting to explore automatically a wide range of algorithm combinations for the eHealth-KD challenge in order to find out the combination of strategies that obtains the best results.

4 AutoML strategy for knowledge discovery from text

Our proposal is based on AutoML, using the metaheuristic grammatical evolution (O'Neill and Ryan, 2001) for defining and exploring the space of solution for dealing with the Scenario 1 of the eHealth-KD challenge. However, the optimization process is different to the traditional grammatical evolution formulation because our grammar involves both discrete and continuous parameters. Instead, we propose a modified version of probabilistic grammatical evolution (Kim and Ahn, 2015). The proposal is described by dividing the process into two stages: (I) the definition of a grammar specific for the eHealth-KD challenge (see section 4.1) and (II) the design of an optimization process, based on this grammar, for obtaining the best (i.e., optimal) pipeline (see section 4.2). Figure 2 shows a visual representation of the complete AutoML process designed in this research.

4.1 Definition of the space of solutions

In this stage we define a grammar that takes into considerations the solutions of the eHealth-KD challenge for the edition TASS-2018. This grammar includes a source code representation that corresponds to different pipelines designed for this task and therefore defines the solutions space for the optimization process. Figure 3 shows an extract of the grammar, as defined in the source code for the experimentation. The complete grammar and associated source code can be browsed online[2].

The grammar defines `Pipeline`, which consists of a representation phase (`Repr`) where text is preprocessed, cleaned and vectorized; and three classification phases, one for each subtask. The

[2] https://github.com/knowledge-learning/hp-optimization/blob/master/hpopt/examples/ehealthkd.py

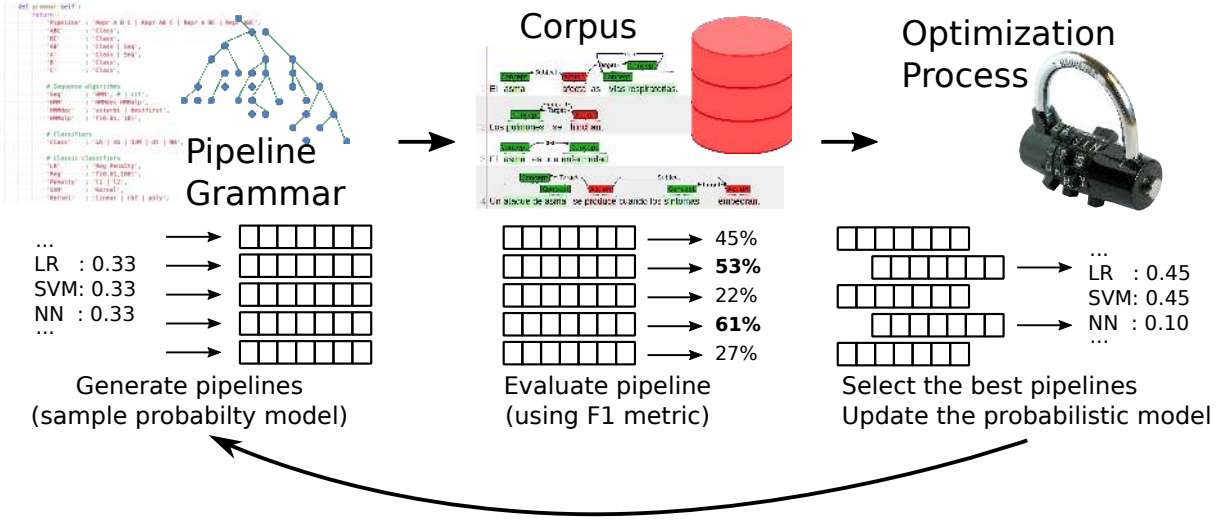

Figure 2: Visual representation of the AutoML process defined. A probabilistic grammar is sampled to obtain potential pipelines, which are evaluated in the eHealth-KD corpus. An optimization process selects the best pipelines and updates the probabilistic model. This iterative process converges towards the best performing pipelines.

classification phases can be performed sequentially (i.e., first subtask A, then B and then C), or some of the subtasks can be performed jointly. Hence, the complete pipeline can be performed in four different ways, according to how subtasks are handled. The representation phase consists of six steps: (1) a preprocessing step (`Prep`) where accents and punctuation symbols are optionally removed; (2) a tokenization step (`Token`); (3) a step (`MulWords`) where single tokens are combined into multi-word tokens using several strategies; (4) a semantic step (`SemFeat`) where postag and dependency features are optionally added; (5) another preprocessing step (`PosPrep`) where stopwords are optionally removed and stemming is optionally applied; and (6) a final step (`Embed`) where text is represented as bag-of-words or using an embedding, i.e., *word2vec* (Mikolov et al., 2013).

Each classification subtask in turn can be solved using one of several classifiers. Five different classifiers are considered: logistic regression (`LR`), SVM, multinomial Naive Bayes (`NB`), decision trees (`DT`) and a restricted subset of neural networks (`NN`). For shallow classifiers, hyperparameters are also considered, such as regularization strength in logistic regression or kernel type in SVM. Additionally, for subtask A a hidden Markov model is added as a sixth classifier with corresponding hyperparameters. In the case of neural networks, three different architectures are allowed: either a convolutional layer (`CVL`), a recurrent layer (`RL`), or fully connected dense layers (`DL`), followed by a final dense layer (`FL`). In all cases, hyperparameters such as dropout rate (`Drop`), number of layers, and layer sizes are allowed to vary. In the case of convolutional layers, also filter sizes are considered.

The complete grammar involves 78 productions, and defines a total of $349,479,936 \approx 2^{28}$ different possible pipelines without considering continuous hyperparameter values.

4.2 Optimization process

The optimization process is based on probabilistic grammatical evolution (Kim and Ahn, 2015), where each decision (production in the grammar) is represented as a probability distribution. Initially all decisions (i.e., classifications algorithms, pre-processing steps, etc.) have an uniform probability to be chosen. However, during the optimization process these probability distributions are updated towards selecting the top performing pipelines. The update is performed according to the following rule:

$$\theta_{t+1} = (1 - \alpha) \cdot \theta_t + \alpha \cdot \theta_t^* \qquad (1)$$

Where θ_t is the probability model in generation t; θ_t^* is the marginal probability model induced by the best k performing solutions in generation t; and α is a small learning rate. In time, this process converges to a subset of the solution space with a better performance on average on the evaluated task. The best pipeline found during the whole optimization process is reported as the final solution.

```
Pipeline := Repr A B C | Repr AB C |
            Repr A BC  | Repr ABC
A        := Class | Seq
B        := Class
C        := Class

# Representation
Repr     := Prep Token MulWords
            SemFeat PosPrep Embed

# Classic classifiers
Class    := LR | SVM | NB | DT | NN
LR       := Reg Penalty
Reg      := f(0.01,100)
Penalty  := l1 | l2
SVM      := Kernel
Kernel   := linear | rbf | poly

# Neural networks
NN       := Drop CVL DL FL |
            Drop RL DL FL  |
            Drop DL FL
....
```

Figure 3: Extract of the 78 productions of the grammar defined for the TASS 2018 eHealth-KD challenge. For simplicity, only the top productions and some example productions of classifiers and their hyperparameters are shown.

The probability model θ_t defines a joint probability for all the productions of the grammar at iteration t. Productions that correspond to discrete choices (e.g. `Penalty := l1 | l2`) are modelled with a discrete uniform distribution. Productions that correspond to integer or real hyperparameter values (e.g., `Reg := f(0.01,100)`) are modelled as a uniform distribution parameterized by a mean and a standard deviation, which are initialized according to the grammar definition.

After each iteration of the optimization process, the probability model is updated by interpolating between the current model θ_t and a marginal model θ_t^* using the parameter α. The update rule (eq. 1) is applied at each production of the grammar. Discrete uniform distributions are updated by interpolating the probability vectors and renormalizing. Integer and real hyperparameters are updated by interpolating the mean and standard deviation. The marginal probability model θ_t^* is computed from a selection of the top n pipelines, in terms of F_1 score. Each production in θ_t^* is assigned a probability distribution inferred from the sample of the actual productions used in the top n pipelines.

This process allows a more intelligent search in the space of all possible pipelines, guided by the structure of the defined grammar. Even though the space of all possible pipelines is exponentially large (with respect to the size of the grammar), grammatical evolution can sample from this solution space to obtain a broad view and iteratively focus on the most promising regions, i.e., the subsets of pipelines with the best performance. Figure 4 shows a simplified representation of this optimization process, illustrating that some specific pipelines have a different performance. Notice that whole details (i.e., representation complexity and hyperparameter selection) are not displayed in this illustration due to space limitation.

5 Experimental results

The algorithm described in Section 4 was implemented and executed for a total of 60 generations. Each generation consisted of 50 pipelines, with a selection of the best 10, and a learning factor $\alpha = 0.05$. In total, 3000 different pipelines were evaluated in 257 hours of computation time, which resulted in an average evaluation time of 5.17 minutes per pipeline. A timeout of 10 minutes was used to stop the evaluation of very long pipelines. These incomplete pipelines are given a fitness of 0, which makes the optimization algorithm eventually steer away from them. The total number of generations was adjusted according to computational constraints. The optimization process was monitored regularly and stopped after a sufficient computation time in which little to no improvement was observed, which indicated a convergence of the probability model.

The evaluation was performed in the eHealth-KD corpus (Piad-Morffis et al., 2019a), using the training and development collections for training and the test collection for evaluation. Thus, the training data is comprised of 844 sentences resulting in a total of 9540 annotations among key phrases and relations. The test data is comprised of 100 sentences (for Scenario 1) with 1100 total annotations. After 60 generations, the best performing pipeline (actually found in generation 18) achieved a F_1 score of 0.754 in Scenario 1 of the eHealth-KD challenge. This represents a 1% absolute improvement from the top result presented in the eHealth-KD challenge, and a 4.2% absolute improvement over the average result of the top 3 alternatives ($\overline{F_1} = 0.711$). Table 1 shows this result in a comparison with the rest of the approaches presented in Section 3.

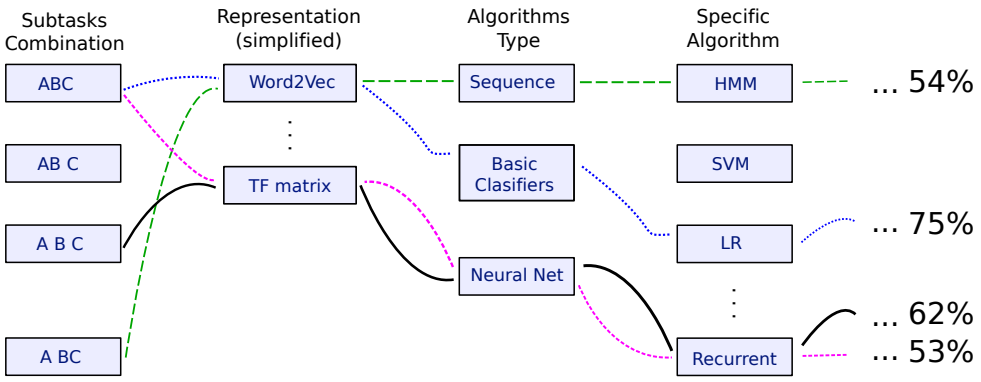

Figure 4: Simplified representation of the optimization process. For clarity, details such as hyperparameter selection for each algorithm are not displayed. Performance percentages are illustrative only, not actual values.

Approach	F_1 **(Scenario 1)**
Zavala et al. (2018)	0.744
López-Ubeda et al. (2018)	0.710
Palatresi and Hontoria (2018)	0.681
Suarez-Paniagua et al. (2018)	0.310
Our proposal	**0.754**

Table 1: Comparison of approaches in the eHealth-KD challenge. Only researchers that participated in Scenario 1 are considered.

Finally, the best pipeline found was composed by the following strategies: the three subtasks were performed sequentially, using shallow classifiers in each. For subtask A, the classifier selected was SVM with an RBF kernel, while for subtask B, a Logistic Regression with L_2 penalization was used, and for subtask C a Decision Tree (ID3) was selected. In terms of representation, the optimizer selected one-hot encoding with bi-grams using stemming and with pos-tagging as an additional syntactic feature. Figure 5 summarizes the structure of the final pipeline selected.

Unfortunately it is not possible to compare our results with other AutoML techniques directly, since existing AutoML approaches are designed to deal with black-box machine learning problems where the input is a feature matrix (see Section 1). In the eHealth-KD scenario, there are several high-level decisions, such as selecting whether subtasks are performed sequentially or in combined, or which preprocessing steps to apply, that are necessary before obtaining a suitable feature matrix. Hence, these decisions cannot be modeled with existing AutoML techniques. For this reason, even though part of the eHealth-KD problem can

```
Pipeline:
  Representation:
    Preprocessing:
      Remove Punctuation: no
      Strip Accents: no
    Multi-Words:
      - Strategy: postag
      - N-grams: 2
    Semantic Features:
      PosTag: yes
      Dependencies: no
    Postprocessing:
      Stopwords: no
      Stemming: yes
    Embedding: none
  Subtask A:
    SVM:
      Kernel: RBF
  Subtask B:
    Logistic Regression:
      Regularization: 40.93
      Penalty: L2
  Subtask C: Decision Tree
```

Figure 5: Summarized representation of the best pipeline discovered.

be solved using alternative AutoML approaches, another part of the challenge would require external tools or a custom implementation. In future work, we will explore other machine learning problems where a comparison with alternative AutoML techniques is possible.

6 Discussion

The best pipeline obtained by our approach achieves a small advantage over the top result pre-

sented in the eHealth-KD challenge. However, this pipeline is simpler than most of the presented approaches, since it involves only shallow classifiers and basic NLP techniques. Furthermore, this pipeline was obtained automatically, without any human input after the initial definition of the grammar. This makes our strategy easier to extend to other knowledge discovery tasks, and potentially other machine learning scenarios, simply by defining a suitable grammar and providing the corresponding implementation. To support this statement, even though this research focuses on the eHealth-KD dataset, we provide open source implementations in several different machine learning problems[3].

6.1 Analysis of the optimization process

The optimization process shows a significant improvement in the solutions' average fitness. Figure 6 shows the evolution of the average fitness (in terms of F_1 as defined by the eHealth-KD challenge, Scenario 1) and the fitness of the best pipeline in each generation. This behavior illustrates that the optimization process improves over time. The relatively low average fitness on the first generations is due to invalid pipelines, which are given a fitness score of 0. Invalid pipelines were generated by some combinations of incompatible decisions, mainly for implementation restrictions, or when the timeout was reached. For example, some classification algorithms require dense matrices, which are incompatible with the use of sparse bag-of-word representations (one-hot encoding). These restrictions are very broad and complex to be fully represented in the grammar, since many of them are context-sensitive and depend on which selections were made in different parts of the grammar. In these cases, the computational implementation results in a runtime exception during the evaluation of the pipeline. Since this exception cannot be predicted beforehand, when it occurs the optimization code catches the exception and instead returns the lowest possible fitness ($F_1 = 0$).

However, even though invalid pipelines pose an issue in the first generations, as the optimization process continues, the influence of invalid pipelines decreases gradually. This effect can be observed in the population average fitness, which

[3] https://github.com/knowledge-learning/hp-optimization/blob/master/hpopt/examples

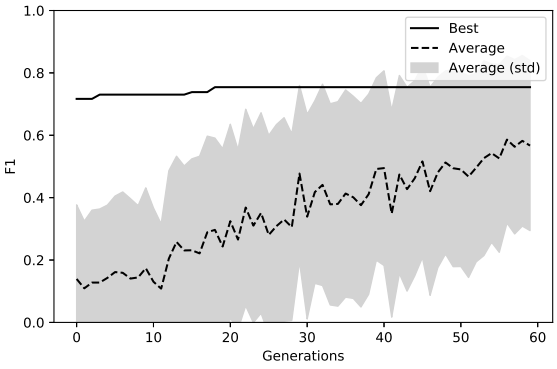

Figure 6: Evolution of the best and average fitness of pipelines in each generation.

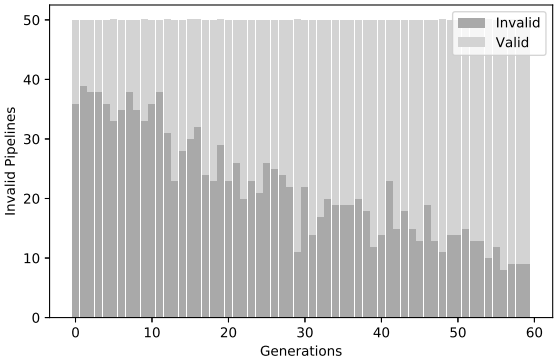

Figure 7: Proportion of invalid vs valid pipelines found in each generation.

increases steadily. Furthermore, the standard deviation of the fitness (gray area in Figure 6) shows a steady behaviour across all generations. This indicates that both the best and worst pipelines (with respect to fitness) are improving. Hence, the optimization procedure actually learns to avoid invalid pipelines. To further support this observation, Figure 7 shows the number of invalid pipelines (with a score of 0) in each generation. The steady reduction of invalid pipelines in each generation is an evidence that the optimization process eventually converges towards solutions that represent mostly valid pipelines.

6.2 Analysis of feature relevance

In this section we provide an analysis of all pipelines, including their parameters and algorithms, in order to identify their most relevant features. As features, we consider all the decisions involved in a pipeline, from subtasks order, to the algorithms to use and the specific values for all its hyperparameters. These features are obtained directly from each pipeline's representation, by con-

sidering all the productions of the grammar (i.e., algorithms and hyperparameter values) that are used in one specific pipeline. As an example, if one specific pipeline applies stopword removal, then a feature `PosPrep.StopW=yes` is generated. Our grammar involves a total of 242 such features, which account all possible decisions for each pipeline.

The relevance of each feature can be estimated by fitting a linear regression to predict the F_1 score of the evaluated pipelines, only the 1667 valid pipelines. In this case, the estimation involves a coefficient of determination $R^2 = 0.763$, meaning that these 242 features are indeed a good indicator of a pipeline F_1 score. The weights computed by the linear regression indicate the absolute improvement that each feature provides. For example, considering stopwords removal (i.e., `PosPrep.StopW=yes`) with a computed weight of 0.001, pipelines where stopwords removal are active obtain a 0.1% higher F_1 score on average than the rest of pipelines where this feature is non-active. This specific case indicates that stopword removal is slightly beneficial, but not crucial in obtaining a high F_1 score.

Table 2 shows the weight assigned to each classifier in each task. These weights are correlated with the average performance of all the pipelines that use a given classifier. For space restrictions it is not possible to report the weights associated to each hyperparameter of each classifier. Since every classifier includes multiple hyperparameters, the performance of two specific pipelines using the same classifier can vary widely. For example, on average, pipelines using the classifier SVM achieve a lower score than pipelines using neural networks. Nevertheless, the best pipeline found uses SVM in subtask A (see Figure 5) outperforming the rest due to the specific hyperparameters involved in that pipeline. Hence, even though neural networks are assigned a relative high weight they are not selected for the best pipeline.

Similarly, table 3 show a subset of the features corresponding to the representation phase and their corresponding weights. Interestingly, it is also the case that some features present in the best pipeline do not show the largest weights.

As explained before, it is important to highlight that not necessarily the top performing pipeline found by the optimization algorithm will be composed by top weighted features. The performance

	Weights by subtask		
Algorithm	**A**	**B**	**C**
LR	0.0014	0.0074	-0.0127
NN	**0.0361**	**0.0320**	-0.0009
SVM	-0.0157	-0.0221	0.0018
DT	0.0165	0.0223	**0.0024**
NB	0.0222	0.0128	-
HMM	-0.0081	-	-

Table 2: Relevance of the classification algorithms in each subtask. The top weight in each subtask is highlighted. Missing values correspond to combinations that were not evaluated in any valid pipeline.

of a pipeline will depend, in general, of complex interactions between its components that are not completely captured in a simple linear regression model. However, there is a large correlation between feature weights and pipeline performance, as demonstrated by the coefficient of determination ($R^2 = 0.763$) of the regression. Since the performance of a specific pipeline depends heavily on the values of the hyperparameters, a deeper analysis is necessary to estimate the impact of each algorithm in each step of the pipelines, taking into account the actual values of hyperparameters used. In future work, we will explore using this type of analysis to warm-start the probabilistic model.

7 Conclusions and future work

This paper presents an Automatic Machine Learning strategy based on probabilistic grammatical evolution to extract knowledge from health documents in Spanish language. Our proposal involves an optimization process which explores a large space of possible pipelines and chooses the best performing ones automatically. The evaluation was performed on a complex scenario of the TASS 2018 eHealth-KD challenge, where the best pipeline discovered improves over the state-of-the-art by combining features, decisions and strategies from different authors. In addition, the data gathered during the optimization provided insights about the optimal settings to deal with the challenge faced. These results show that an AutoML strategy based on grammatical evolution is effective for optimizing machine learning pipelines to solve knowledge discovery challenges from natural language text.

As future work, we plan to study the introduction of high-level knowledge to deal with the is-

Feature	Value	Weight
Embed	none	**0.015**
Embed	onehot	-0.030
Embed	wordVec	0.006
MulWords	colloc	-0.032
MulWords	none	**0.014**
MulWords	postag	0.009
PosPrep.Stem	no	-0.003
PosPrep.Stem	yes	**-0.005**
PosPrep.StopW	no	-0.010
PosPrep.StopW	yes	**0.001**
Prep.DelPunt	no	**-0.001**
Prep.DelPunt	yes	-0.008
Prep.StripAcc	no	**0.001**
Prep.StripAcc	yes	-0.010
SemFeat.Dep	no	-0.013
SemFeat.Dep	yes	**0.003**
SemFeat.PosTag	no	-0.025
SemFeat.PosTag	yes	**0.016**

Table 3: Relevance of the representation features. The top weight for each feature is highlighted.

sue of invalid pipelines and improve the performance of the optimization process. This knowledge can be in the form of explicit rules that guarantee the validity of the pipelines sampled from the grammar; and in the form of statistical information extracted from similar challenges that helps pre-defining a probabilistic model. Another issue to research will be the use of regression models to estimate the expected fitness of a pipeline given its features, as illustrated in Section 6. This addition would support meta-learning algorithms, allowing to reduce the optimization time and increase its performance by learning from past executions. Finally, by modifying the grammar, this strategy can be extensible to other machine learning challenges. Therefore, we plan to explore this line of research in the future, to compare our proposal with other AutoML frameworks in standard benchmarks.

Acknowledgments.

This research has been supported by a Carolina Foundation grant in agreement with University of Alicante and University of Havana. Moreover, it has also been partially funded by both aforementioned universities, the Generalitat Valenciana and the Spanish Government through the projects PROMETEU/2018/089, RTI2018-094653-B-C22 and RTI2018-094649-B-I00.

References

Ken Barker, Bhalchandra Agashe, Shaw Yi Chaw, James Fan, Noah Friedland, Michael Glass, Jerry Hobbs, Eduard Hovy, David Israel, Doo Soon Kim, et al. 2007. Learning by reading: A prototype system, performance baseline and lessons learned. In *AAAI*, volume 7, pages 280–286.

Boyuan Chen, Harvey Wu, Warren Mo, Ishanu Chattopadhyay, and Hod Lipson. 2018. Autostacker: A compositional evolutionary learning system. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '18, pages 402–409, New York, NY, USA. ACM.

Philipp Cimiano, Alexander Mädche, Steffen Staab, and Johanna Völker. 2009. *Ontology Learning*, page 245–267. Springer Berlin Heidelberg, Berlin, Heidelberg.

David Crystal. 1997. *The cambridge encyclopedia*. Cambridge University Press New York.

S Estevez-Velarde, Y Gutierrez, A Montoyo, A Piad-Morffis, R Munoz, and Y Almeida-Cruz. 2018. Gathering object interactions as semantic knowledge. In *Proceedings on the International Conference on Artificial Intelligence (ICAI)*, pages 363–369. The Steering Committee of The World Congress in Computer Science, Computer ....

Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. 2015. Efficient and Robust Automated Machine Learning. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, page 2962–2970. Curran Associates, Inc.

Fausto Giunchiglia and Mattia Fumagalli. 2017. Teleologies: Objects, actions and functions. In *Conceptual Modeling*, pages 520–534, Cham. Springer International Publishing.

Graciela H Gonzalez, Tasnia Tahsin, Britton C Goodale, Anna C Greene, and Casey S Greene. 2015. Recent advances and emerging applications in text and data mining for biomedical discovery. *Briefings in bioinformatics*, 17(1):33–42.

Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren. 2019. Automatic machine learning: methods, systems, challenges. *Challenges in Machine Learning*.

Haifeng Jin, Qingquan Song, and Xia Hu. 2018. Efficient neural architecture search with network morphism. *CoRR*, abs/1806.10282.

Hyun-Tae Kim and Chang Wook Ahn. 2015. A New Grammatical Evolution Based on Probabilistic Context-free Grammar. In *Proceedings of the*

18th Asia Pacific Symposium on Intelligent and Evolutionary Systems - Volume 2, page 1–12, Cham. Springer International Publishing.

Brent Komer, James Bergstra, and Chris Eliasmith. 2014. Hyperopt-sklearn: automatic hyperparameter configuration for scikit-learn. In *ICML workshop on AutoML*, pages 2825–2830.

Pilar López-Ubeda, Manuel Carlos Díaz-Galiano, María Teresa Martín-Valdivia, and L. Alfonso Urena-Lopez. 2018. Sinai en tass 2018 task 3. clasificando acciones y conceptos con umls en medline. In *Proceedings of TASS 2018: Workshop on Semantic Analysis at SEPLN (TASS 2018) co-located with 34nd SEPLN Conference (SEPLN 2018)*, volume 2172 of *CEUR Workshop Proceedings*, pages 83–88. CEUR-WS.org.

Eugenio Martínez-Cámara, Yudivián Almeida-Cruz, Manuel Carlos Díaz-Galiano, Suilan Estévez-Velarde, Migueí A García-Cumbreras, Manuel García-Vega, Yoan Gutiérrez, Arturo Montejo-Ráez, Andrés Montoyo, Rafael Muñoz, Alejandro Piad-Morffis, and Julio Villena-Román. 2018. Overview of TASS 2018: Opinions, Health and Emotions Resumen de TASS 2018: Opiniones, Salud y Emociones. *CEUR Workshop Proceedings*, 2172:13–27.

Salvador Medina and Jordi Turmo. 2018. Joint classification of key-phrases and relations in electronic health documents. In *Proceedings of TASS 2018: Workshop on Semantic Analysis at SEPLN (TASS 2018) co-located with 34nd SEPLN Conference (SEPLN 2018)*, volume 2172 of *CEUR Workshop Proceedings*, pages 83–88. CEUR-WS.org.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, page 3111–3119. Curran Associates, Inc.

T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, B. Yang, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling. 2018. Never-ending learning. *Commun. ACM*, 61(5):103–115.

Randal S. Olson and Jason H. Moore. 2016. Tpot: A tree-based pipeline optimization tool for automating machine learning. In *Proceedings of the Workshop on Automatic Machine Learning*, volume 64 of *Proceedings of Machine Learning Research*, pages 66–74, New York, New York, USA. PMLR.

M. O'Neill and C. Ryan. 2001. Grammatical evolution. *IEEE Transactions on Evolutionary Computation*, 5(4):349–358.

Jorge Vivaldi Palatresi and Horacio Rodríguez Hontoria. 2018. Medical knowledge discovery by combining multiple techniques and resources. In *Proceedings of TASS 2018: Workshop on Semantic Analysis at SEPLN (TASS 2018) co-located with 34nd SEPLN Conference (SEPLN 2018)*, volume 2172 of *CEUR Workshop Proceedings*, pages 83–88. CEUR-WS.org.

Alejandro Piad-Morffis, Yoan Gutiérrez, and Rafael Muñoz. 2019a. A corpus to support ehealth knowledge discovery technologies. *Journal of biomedical informatics*, 94:103172.

Alejandro Piad-Morffis, Yoan Gutiérrez, Suilan Estévez-Velarde, Yudivián Almeida-Cruz, Andrés Montoyo, and Rafael Muñoz. 2019b. Analysis of ehealth knowledge discovery systems in the tass 2018 workshop. *Procesamiento del Lenguaje Natural*, 62(1).

Víctor Suarez-Paniagua, Isabel Segura-Bedmar, and Paloma Martínez. 2018. Labda at tass-2018 task 3: Convolutional neural networks for relation classification in spanish ehealth documents. In *TASS 2018 – Taller de Análisis Semántico en la SEPLN*.

Alex G. C. de Sá, Walter José G. S. Pinto, Luiz Otavio V. B. Oliveira, and Gisele L. Pappa. 2017. RECIPE: A Grammar-Based Framework for Automatically Evolving Classification Pipelines. In *Genetic Programming*, page 246–261, Cham. Springer International Publishing.

Chris Thornton, Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. 2013. Auto-weka: Combined selection and hyperparameter optimization of classification algorithms. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 847–855, New York, NY, USA. ACM.

Renzo M. Rivera Zavala, Paloma Martínez, and Isabel Segura-Bedmar. 2018. A hybrid bi-lstm-crf model for knowledge recognition from ehealth documents. In *Proceedings of TASS 2018: Workshop on Semantic Analysis at SEPLN (TASS 2018) co-located with 34nd SEPLN Conference (SEPLN 2018)*, volume 2172 of *CEUR Workshop Proceedings*, pages 83–88. CEUR-WS.org.

Barret Zoph and Quoc V. Le. 2016. Neural architecture search with reinforcement learning. *CoRR*, abs/1611.01578.