Rule by Rule: Learning with Confidence through Vocabulary Expansion

Albert Nössig[1,2], Tobias Hell[2], and Georg Moser[1]

[1] Department of Computer Science, University of Innsbruck, Tyrol, Austria

[2] Data Lab Hell GmbH, Europastraße 2a, 6170 Zirl, Tyrol, Austria

Abstract. In this paper, we propose an innovative iterative approach to rule learning, specifically designed for (though not limited to) text-based data. Our method focuses on progressively expanding the vocabulary used in each iteration, resulting in a significant reduction in memory consumption. Additionally, we introduce a *Value of Confidence*, which quantifies the reliability of the generated rules. By leveraging the *Value of Confidence*, our approach ensures that only the most robust and trustworthy rules are retained, thereby enhancing the overall quality of the rule learning process. We demonstrate the effectiveness of our method through extensive experiments on both textual and non-textual datasets, including a case study of significant interest to the insurance industry, highlighting its potential for real-world applications.

Keywords: Rule Learning, Explainable Artificial Intelligence, Text Categorization, Reliability of Rules.

1 Introduction

In recent years, the rapid advancement of Artificial Intelligence (AI) technologies has revolutionized various industries and aspects of our daily lives (cf. [1,2,3], for instance). However, as AI systems become more complex and sophisticated, the need for transparency and interpretability in their decision-making processes has become increasingly crucial. The concept of *Explainable Artificial Intelligence* (XAI; see for example [4,5]) has emerged as a response to this demand, aiming to enhance the trust, accountability, and understanding of AI systems by providing explanations for their outputs and actions.

Indeed, in many application domains of machine learning, such as automotive, medicine, health and insurance industries, the need for security and transparency of the applied methods is not only preferred but increasingly often of utmost importance or even legally mandated (cf. the *EU Artificial Intelligence Act*, for instance).

A classic example in this context – often categorized as *most informative* in the area of XAI [6] – is the generation of deterministic (if-then-else) rules that can be used for classification. For instance, when predicting a patient's health status, the easily comprehensible rule shown below is clearly preferable to the opaque output of a *black-box* model such as a neural network, for both the doctor and the patient, as the decision is fully transparent.

```
IF  BloodPressure in [70,80]
  AND Insulin in [140,170]
THEN  Diabetes = Yes.
```

Among others, the field of *Rule Induction* [7] particularly investigates the construction of simple if-then-else rules from given input/output examples and provides some commonly applied methods to obtain deterministic rules for the solution of a (classification) problem at hand (more details are given in the Supplementary Material[3]). Representative examples of such rules are shown for each data set considered in our experiments in Section 4,

[3] See `https://arxiv.org/abs/2411.00049`.

David C. Wyld et al. (Eds): BDAB, SVC, AIBD, ACSTY, NATP – 2025
pp. 47-65, 2025. CS & IT - CSCP 2025 DOI: 10.5121/csit.2024.150105

illustrating the major advantages of rule learning methods, namely their transparency and comprehensibility, which make them a desirable classification tool in many areas.

Unfortunately, these benefits are coupled with the major drawback of generally less accurate results – often referred to as the *interpretability-accuracy trade-off* [8]. Moreover, for a long time, it has not been possible to efficiently apply rule learning methods on very large data sets [9] as considered, for instance, in the industrial use case discussed in Section 4.3, which is of central interest to us and our collaboration partner – the *Allianz Private Krankenversicherung (APKV)*. We have already extensively investigated these issues in collaboration with the aforementioned company from insurance industries with the primary aim of establishing rule learning methods – particularly FOIL [10] and RIPPER [11] – as an efficient tool in the reimbursement process. Note that we have explored a wide range of rule learning methods in previous work [12,13], but ultimately, we chose to focus on FOIL – one of the first methods from the field of *Inductive Logic Programming* (ILP; cf. [14]) – and RIPPER, which is state-of-the-art in *Rule Induction*. This is because especially the more modern ILP-tools have been shown to be unsuited for our needs, as further explained in [12]. In the papers cited above, we introduced approaches to solve the mentioned difficulties concerning the application of rule learning methods in a production environment at least to some extent. First, we developed a modular approach [12], enabling the application of ordinary rule learning methods such as FOIL and RIPPER on very large data sets including several hundreds of thousands examples. However, the generally poorer performance compared to state-of-the-art methods in terms of accuracy remained. So, we came up with an extension of the introduced modular approach in the form of the voting approach presented in [13].[4] After consultation with our collaboration partner, we agreed that, at the end of the day, it is even more important to ease the understanding of a classification than to make the whole procedure fully transparent. Thus, this additional step in the decision-making process addresses the *interpretability-accuracy trade-off* by incorporating an ensemble of explainable and unexplainable methods. As a consequence, the procedure loses its full transparency but gains a significant improvement in classification accuracy, while preserving *end-to-end explainability* by corroborating each prediction with a comprehensible rule.

At this point, we have already made a significant progress toward the application of trustworthy AI methods within the company. However, another critical issue not adequately addressed by the combination of the two approaches above is the handling of text-based data. The data basis for the reimbursement use case is a collection of (scanned) bills, where we extracted the most important information in the form of nominal (and continuous) attributes, as described in more detail in Section 4.3. Unfortunately, this preprocessing method may result in the loss of much additional information present in the original textual data.

Up to this point, however, we have primarily focused on nominal data, with the exception of the *IMDB movie reviews* data set,[5] which has been part of the benchmark data sets in the evaluation of our modular approach. The results have not been really satisfying, as the achieved accuracy fell below expectations. On the one hand, this issue is solvable by our voting approach, at least to some extent. However, on the other hand, it has shown that the form and complexity of the generated rules is not reasonably applicable for (end-to-end) explainable classification. What does not seem overly problematic in the case of the relatively small IMDB data set is the choice and, especially, the size of the underlying dictionary used to generate rules. For the movie reviews, we simply con-

[4] A concise summary of our previously introduced approaches is given in the Supplementary Material.
[5] See `https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews`.

sidered the thousand most common words in the data set, but the bills submitted to the insurance company are significantly more complex. They usually consist of at least one page of text using partly highly complicated technical terms from various medical fields instead of 2-3 sentences describing personal opinions about movies in simple language. Note that simply using a much larger dictionary as the basis for the rule learning process is not a solution, as the computation time and memory consumption required to generate the rules increase drastically with the size of the dictionary. In this paper, we aim to gain more control over the complexity of the generated rules and make it possible to reasonably apply rule learning methods such as FOIL and RIPPER to text-based data by starting with a concise dictionary (designed by domain experts) and decreasing the number of considered examples before iteratively expanding the applied dictionary. The intention behind this approach is to learn *general* rules in the first step, using a small and computationally inexpensive dictionary for a very large number of input examples. With each learned rule, the number of considered positive examples decreases by definition of the rule learning algorithms. When a certain point is reached – either a predefined number of iterations or a condition regarding the quality of a rule, as described in detail in Section 3 – we extend the dictionary to handle more *specific* examples. This iterative process can be repeated until a comprehensive dictionary is used for the remaining *edge cases*. In addition, the basic idea behind this approach can also be applied to nominal (and continuous) data to improve the *quality* of a rule, as explained in Section 3 and shown in the experimental evaluation in Section 4.

In addition to evaluating our approach on common benchmark data sets for classification of textual data (*IMDB* [15], *Reuters-21578* [16], *Hatespeech*[6]), we also demonstrate the advantages of applying the core idea of our approach to non-textual data, using some common data sets from the *UCI Machine Learning Repository* [17] or *Kaggle*[7], respectively. Moreover, we present novel results on explainable *classifications of bills for reimbursement*, particularly using textual data as input. The latter case study stems from an industrial collaboration with *Allianz Private Krankenversicherung (APKV)*, an insurance company offering health insurance services in Germany.

In summary, our primary goal is to address a text-based classification problem with reasonable time and computational complexity by applying easily interpretable rules generated from a dictionary of variable size. Moreover, we define a measure for the quality of a rule and integrate it in the iterative process on which our proposed approach is based on. As shown in the experiments, this iterative rule refinement also proves beneficial for non-textual data. All in all, this paper directly builds on our previous work and expands upon the approaches presented therein to handle textual data more efficiently and gain more control over the complexity of the generated rules by iteratively extending the size of the applied dictionary (or, in general, the number of attributes).

More precisely, we make the following contributions.

Iterative Approach Based on Rule Learning We introduce a novel iterative approach based on rule learning exploiting the benefits of a variable number of attributes (in particular an adaptable dictionary) during the generation of a rule set (see Section 3 for further details).

Together with the modular as well as the voting approach introduced in our previous work [12,13], this positions rule learners as a serious alternative to state-of-the-art classi-

[6] See `https://www.kaggle.com/datasets/mrmorj/hate-speech-and-offensive-language-dataset`.
[7] See `https://www.kaggle.com/`.

fication tools and enables the application of tried and trusted rule learning methods in a complex and diverse production environment.

Experimental Evaluation Further, we provide ample experimental evidence that our methodology not only clearly simplifies the application of rule learning methods on text-based data but also provides significant improvements on the accuracy for the standard benchmarks (see Section 4).

Industrial Use Case Finally, we show that our approach makes it possible to efficiently apply the method we successfully introduced in previous work to text-based data, in particular the raw OCR scans used for reimbursement. We emphasize that our classification yields comprehensible rules that are of direct interest to our industrial collaboration partner (see Section 4.3).

Overview. Section 2 serves to discuss related work focusing on similar goals as considered in this paper, especially on various forms of (explainable) text-based classification, while we concretely introduce our aforementioned iterative approach as well as the *Value of Confidence* applied therein as a measure of reliability of a rule in Section 3. Section 4 provides ample evidence of the advantages of our approach and presents the case study mentioned. Finally, in Section 5 we summarize the main results and discuss ideas for future work.

2 Related Work

After motivating the basic idea behind the approach introduced in this paper, this Section discusses related work that focuses on the (explainable) classification of textual data as well as novel ideas in the context of rule learning in general.

Regarding text classification in general, there is a huge number of methods out there dealing with this problem. Some surveys summarizing the most common (explainable as well as unexplainable) approaches have been done in recent years for instance by [18,19,20,21]. Moreover, Mendez Guzman et al. [22] recently published a survey comparing different rationalisation approaches in the context of explainable text classification. Furthermore, Altinel et al. [23] give an overview of common semantic text classification methods and discuss the benefits of these methods over traditional text classification approaches.

A more specific method utilizing similar ideas to those we apply in our approach is proposed by Johnson et al. [24], who introduce a *tool kit for text categorization* called *KitCat*. They not only focus on the explainable classification of textual data but also make use of a confidence measure for dealing with ambiguities, similar to our *Value of Confidence* introduced in Section 3. For evaluation, they consider in particular the *Reuters-21578* data set where they report a micro-averaged precision/recall of 83.8%. As opposed to their idea of deriving symbolic rules from decision trees optimized to handle sparse data, we directly obtain rules from classical rule learning methods. We focus especially on the complexity of the generated rules with respect to the underlying dictionary in order to improve the versatility of the classical methods. Note that we cannot really compare the achieved results, since we used a different data split. However, on *NLTK's Reuters corpus* we report an accuracy of about 80.5% for RIPPER and 81.7% for FOIL, respectively.

The *Reuters-21578* data set is a common benchmark for the evaluation of various classification methods on text-based input data and has been extensively studied, for instance, by Debole et al. [25]. Another approach from the field of explainable artificial intelligence that considers this data set among others is *Olex-GA* [26]. The results of this genetic algorithm are very similar to the *if-then-else* rules generated by the rule learning methods considered by us. In the course of their evaluations, they compare their method among others also with RIPPER and report comparative but slightly worse classification results considering the *break-even point* – the average of precision and recall where the difference between them is minimal – as accuracy metric.

In addition, we consider the *IMDB movie reviews* data set in our experiments which has been investigated also by Pryzant et al. [27], for instance, who utilize ideas from neuro-symbolic learning (cf. [28]) in a semi-supervised machine learning approach resulting in interpretable results in the form of linear combinations of attention scores. They report remarkable results with an F1-score of up to 89.41%, but they appear to have used a subset or a different version of the data set. Specifically, they worked with 25 thousand examples, while we used 50 thousand examples, resulting in an F1-score of 76.5% on our data set. Moreover, regarding this approach it should be noted that there is an ongoing discussion concerning the interpretability of attention weights (cf. [29,30]), whereas the *if-then-else* rules generated by the rule induction methods applied in our approach are commonly categorized as *most informative* in the area of XAI.

Regarding the selection of the applied dictionary in each iteration, we generally use *n-grams* and order them according to the number of appearances in the input data. However, in future work we aim to improve this way of proceeding and apply a more sophisticated feature selection. Concerning this, quite some research has already been done. First of all, there are various metrics out there for a selection of an appropriate number of features. Regarding text classification, a valuable overview is for instance given by Forman [31]. Moreover, HaCohen-Kerner et al. [32] investigate the influence of different types of pre-processing applied on textual input data.

Furthermore, Chen et al. [33] explore the selection of the vocabulary in more detail and aim to find an optimal subset by providing a variational vocabulary dropout. However, this approach is computationally quite demanding and probably not suited for very large data sets. Similarly, Patel et al. [34] incorporate ideas from cooperative game theory with the aim to find an optimal subset of the vocabulary maximizing the performance of a classification model.

Another crucial point we want to address in more detail in future work is the *class imbalance problem*, which occurs when certain classes are underrepresented in the dataset, affecting the performance of classification models, particularly in our use case within the insurance business. Up to now, it has been an acceptable solution for our collaboration partner to summarize the smaller classes into a few super-classes and differentiate between them. However, it would also be interesting to make a more granular distinction and even in the currently applied setting with only a few considered classes, imbalanced data remains a challenge. An extensive study on this topic has been conducted, for instance, by Japkowicz et al. [35] and Krawczyk [36]. Common methods for handling imbalanced data are summarized, for instance, by Spelmen et al. [37].

On the other hand, Ha-Thuc et al. [38] introduce a text classification approach that does not require any labelled data. Instead of human-labelled documents, they rather consider the description and more importantly the relationships with other categories for classification which makes this approach especially suited for data sets with a lot of different (small) classes as present in our use case. So, incorporating this idea might also be

an interesting direction for future work.

Finally, regarding general trends in rule learning, *RIDDLE* by Persia et al. [39] has to be mentioned. They bridge deep learning and rule induction resulting in a *white-box* method that apparently yields state-of-the-art results in many classification tasks in rule induction. Although they claim that "the trained weights have a clear meaning concerning the decisions that the model takes", the level of explainability is probably still lower than the one achieved by the classical rule induction methods, such as RIPPER, for instance. Moreover, for comparison, we also applied our approach on the *Breast Cancer* data set from the *UCI machine learning repository* which has been used by Persia et al. [39] in the empirical evaluation and achieved an accuracy of 95.99% with FOIL and 96.55% using RIPPER, compared to 94.86% as the mean of 5 independent repetitions using the publicly available implementation of the algorithm[8].

3 Methodology

After motivating the ideas behind this paper and summarizing related work as well as previous work on which this paper is build upon, we will introduce the applied methodology in this section. Simply put, our iterative approach is based on a chosen rule learning method and aims to refine the generated rules according to a chosen *Value of Confidence* that we define as follows.

3.1 Value of Confidence

Definition 1. *The **Value of Confidence** is a measure of reliability of a rule generated by a rule learning method. This numeric value is calculated on a validation data set distinct from the training set that is used to generate the rule. There are various possible calculation methods depending on the exact goal of the use case of interest. However, a common metric applied in this context might be the* precision *that is also used within our experiments since it is especially important for our use case in the insurance business. For instance, one option to compute the Value of Confidence is as follows.*

$$VoC = \frac{p}{p+n},$$

where p is the number of positive examples and n the number of negative examples covered by the rule.

Note that, in our case, we prefer to obtain no prediction at all rather than risk a wrong prediction. This is because every bill that can be processed automatically is a gain for the company, as long as the predicted class is correct with a very high degree of certainty. As a result, the precision is an appropriate Value of Confidence for our purpose. However, in other scenarios, it might be acceptable to obtain a (possibly) incorrect prediction instead of no prediction at all. For instance, when the processing of an example by a human or a different kind of method is very cost-intensive (compared to the expenses resulting from a wrong prediction), a mistake may be tolerable. Similarly, if the rule outcomes are used as decision guidance for a human, it may be more desirable to provide a prediction, even if it is not perfectly accurate. A more detailed investigation of different metrics in this context will be part of future work.

[8] See `https://git.app.uib.no/Cosimo.Persia/riddle`

Algorithm 1 Pseudo-Code for Iterative Approach

Input: Training and validation set
Parameter: Maximal number of iterations, Threshold, Initial size of dictionary
Output: Rule with corresponding Value of Confidence

```
Restrict training data to chosen dictionary_size
iteration ← 0
while iteration < max_iterations do
    rule ← apply chosen rule learning method
    if VoC(rule) < threshold then
        add false positives from validation set to training set
        dictionary_size *= 2
        adapt data to new dictionary_size
        iteration += 1
    else
        return rule with corresponding VoC
    end if
end while
```

3.2 Iterative Approach

The basic procedure of the iterative approach is illustrated by the pseudo-code in Algorithm 1 and explained in the following.

In a first step, the given data set is split into training, test and validation sets. For instance, in our experiments we use a 80/20 train-test-split and use 15% of the training data for validation.
The training and validation data serve as input for our approach. As already mentioned above, the training data is used to learn a rule, while the corresponding Value of Confidence is subsequently computed on the validation data.

However, before learning the first rule, the size of the input data is restricted to the chosen *initial dictionary size*. Note that in our experiments we applied the *TfidfVectorizer*[9] with a n-gram range of one to three on the raw text data for preprocessing, where we considered all words that appear at least 5 times in the data set. The resulting total number of features is our original dictionary size and we have ordered the features according to the *inverse document frequency*. It has shown that a reasonable value for the initial dictionary size applied in our algorithm is an eighth of the original dictionary size. This choice is small enough to significantly decrease the necessary memory consumption for the rule generation while it still covers the most important words and groups of words. Moreover, we do not want to apply a huge number of iterations but rather stop after about 5 iterations as done in our experiments since each iteration involves learning a rule which can be quite time-consuming. Using the suggested initial dictionary size, we consider the whole feature set in the fourth iteration and stop after one more iteration. It is probably not possible to find a general optimal value here, since it strongly depends on the underlying data. For instance, considering a data set where very few key words are sufficient to differentiate a large part of the data, the initial dictionary size can be chosen very small whereas a data set consisting of very similar classes might benefit from a larger initial size.

Once the data is prepared, we proceed as follows until the maximal number of iterations is reached or a rule of satisfactory quality (with respect to the VoC) is found.

1. The chosen rule learning method is applied to the current training data in order to learn one rule.

[9] See `https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html`.

2. The Value of Confidence is computed for this rule considering the validation data.
3. The *quality/reliability* of the rule is checked:
   (a) If the corresponding Value of Confidence exceeds the threshold passed as a parameter to the algorithm, we store the rule and remove the covered positive examples from the training set, as usually done in rule learning.
   (b) Otherwise, we increase the dictionary size (usually we multiply it by 2) and add the covered negative examples (i.e., the false positives) from the validation set to the training set.
4. If the quality of the rule is not satisfying, we start the next iteration considering the new training data with increased dictionary size.

The procedure explained above and outlined in Algorithm 1 eventually yields one rule together with the corresponding Value of Confidence. It is repeated until a given number of rules has been generated. Additionally, to prevent overfitting and excessive computation, we include an early stopping mechanism. If the quality of n consecutive rules does not meet the required VoC threshold, the algorithm halts further rule generation. Both the number of consecutive unsatisfactory rules n and the quality threshold are configurable parameters.

In the first place, our iterative approach is intended to make it possible for common rule learning methods to better handle large/complex text-based data sets and reduce memory consumption. However, the basic idea (without increasing the feature space in each iteration) is also suitable for any other kind of data and yields improved results as shown in Section 4.

4 Experimental Evaluation

In this section, we evaluate the iterative approach introduced in this paper on several common benchmark data sets, not only from the field of text classification but also on non-textual data, demonstrating its versatile applicability. Additionally, we investigate a practical example from the insurance industry.

4.1 Experimental Setup

As a first step, the data sets described below are split into train, validation, and test data. Unless stated otherwise, we use 80% of the input data for training and the remaining 20% for testing. From the training data, we use 15% as the validation set for applying our iterative approach. This additional split is not necessary when using the standard method. Thus, the corresponding outcomes presented in the comparison in Section 4.2 are obtained by considering the entire training data set (i.e., 80% of the total input data) without generating a separate validation set. Note that at this point preprocessing has already been done. Specifically, for the text-based data sets, the textual information has been transformed into binary vectors, with attributes ordered according to *inverse document frequency*, as mentioned earlier.

Before starting with our approach, we define an *initial dictionary size*, which is typically an eighth of the total number of attributes, as explained above. For the maximum number of iterations and the applied threshold for the *Value of Confidence*, we always use the same settings: at most 5 iterations with a threshold of 0.9. However, note that we add the rule resulting from the last iteration to our set of rules, regardless of the corresponding *Value of Confidence*. So, in the final ruleset, there might be rules with an unsatisfactory reliability, but we can ignore them during evaluation. In fact, we are interested in the

differences that can be observed by applying only rules with a certain reliability as further shown in Section 4.4.

After that, we can define the rule learning method we want to apply as well as the number of rules that should be generated and our iterative approach proceeds as explained in Section 3.

Before going into detail on the obtained results, we briefly explain the underlying data considered in our experiments. We start with the considered benchmark data sets and discuss the results obtained on them in Section 4.2. Afterwards, in Section 4.3, we will focus on our use case from insurance industries showing that the benefits achieved by our iterative approach are not only present considering some standard benchmark data sets but also on a use case of crucial importance to our industrial collaboration partner.

Hatespeech This data set from Kaggle[10] consists of about 25 thousand Twitter posts labelled as *hate speech*, *offensive language* or *neither*. In our experiments, we summarized the first two classes into one in order to differentiate simply between *Hate Speech/ Offensive Language* or not. So, in our case this is a binary classification task. After preprocessing we consider about 8000 attributes representing the occurrence of words/word groups like *hate, dumb, monkey* as well as a lot of swearwords we do not want to mention here. A simple rule learned in this context could be, for instance:

```
IF  dumb = 1
THEN  Type = Hate Speech
```

meaning that a tweet should be considered as *Hate Speech* if the word *dumb* appears. Of course, there are also more complex rules not just considering the presence of one certain swear word because some words can be used in a completely different context. For example, the word *monkey* is sometimes used in a racist context but also in innocent tweets about a zoo visit resulting in rules like

```
IF  monkey = 1
AND cute = 1
THEN  Type = NOT Hate Speech.
```

Reuters There are various variants of this data set commonly used in literature. We considered the version contained in the python *nltk* package[11] consisting of 10788 news documents assigned to the according categories. After preprocessing, the data set comprised nearly 11 thousand attributes eventually resulting in rules like the following.

```
IF  water = 1
AND carry = 1
THEN  Type = SHIP
```

Note that we distinguish between the 10 most common categories while summarizing the remaining smaller classes as *other*.

IMDB This data set from Kaggle[12] contains 50 thousand informal movie reviews from the *Internet Movie Database* mostly used for sentiment analysis. After preprocessing, we have more than 70 thousand attributes available. It has shown that FOIL is able to handle this number of features while RIPPER cannot, due to its increased complexity, which results in extensive memory consumption. So, for our experiments with RIPPER we cropped the

[10] See `https://www.kaggle.com/datasets/mrmorj/hate-speech-and-offensive-language-dataset`.
[11] See `https://www.kaggle.com/datasets/boldy717/reutersnltk`.
[12] See `https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews`.

feature space and considered only the 20 thousand most important words according to the *inverse document frequency*. An example of a learned rule in this context is as follows.

```
IF  bad = 1
AND great = 0
AND like = 0
THEN  Type = negative
```

Non-textual data sets In addition to these text-based data sets, we also considered non-textual input data in order to investigate the advantages achieved just by assigning a *Value of Confidence* to each generated rule, aiming to maximize this value in our iterative approach without the need of restricting the data to a specific dictionary size. More precisely, we considered the following data sets discussed in more detail in the Supplementary Material of our previous work.[13]

(i) Spambase[14]
(ii) Heart Disease[15]
(iii) Car Evaluation[16]
(iv) Diabetes[17]
(v) Breast Cancer[18]

4.2 Objectives & Summary

The empirical evaluation of the iterative approach introduced in this paper particularly sought to answer the following questions.

RQ1 *Accuracy compared to the base method.* Can the iterative approach provide better classification accuracy than the base method, i.e. the ordinary rule learning method?
RQ2 *Memory consumption compared to the base method.* Is our iterative approach able to significantly reduce memory consumption for rule generation compared to the ordinary method?
RQ3 *Industrial case study.* Are the advantages regarding classification accuracy and memory consumption also observable for the classification of dental bills, an industrial use case?
RQ4 *Level of reliability.* What is the impact of the *Value of Confidence* as a metric of reliability concerning classification accuracy?

In order to investigate these questions, we consider the above-mentioned data sets. Note that the reported results are always obtained on the test data.

For the text-based data sets, we not only compare the resulting accuracy from our proposed iterative approach with the ordinary method, but also measure the memory consumption in our experiments. The corresponding results are shown in Table 1 and visualized in Figure 1 and 2, respectively. Note that all of the experiments are performed on an *AMD Ryzen Threadripper 2950X WOF* CPU.

Regarding accuracy, we can clearly observe that our iterative approach outperforms the ordinary method on the considered data sets for both FOIL and RIPPER. The only

[13] See https://arxiv.org/pdf/2311.07323.
[14] See https://archive.ics.uci.edu/ml/datasets/spambase.
[15] See https://archive.ics.uci.edu/dataset/45/heart+disease.
[16] See https://archive.ics.uci.edu/dataset/19/car+evaluation.
[17] See https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database.
[18] See https://archive.ics.uci.edu/dataset/15/breast+cancer+wisconsin+original.

Data	Learner	Memory Consumption in GiB	Accuracy in %
Hatespeech	FOIL	6,72	82,00
	FOIL - iter.	3,92	86,44
	RIPPER	30,54	89,30
	RIPPER - iter.	13,45	92,64
Reuters	FOIL	4,27	72,21
	FOIL - iter.	2,92	81,74
	RIPPER	13,19	78,89
	RIPPER - iter.	11,26	80,49
IMDB	FOIL	148,80	79,13
	FOIL - iter.	107,57	79,31
	RIPPER	113,60[19]	68,39
	RIPPER - iter.	91,92	75,01

Table 1. Performance of our approach on different benchmark problems for text classification. Note that *iter.* denotes the iterative approach introduced in this paper.

exception is the application of FOIL on the *IMDB* data set, where both approaches are equivalent. A possible reason for this might be the type of language used in this data set, which could also explain the generally rather poor performance of RIPPER on this example (besides the already mentioned restriction of the feature space). The *IMDB* data set consists of movie reviews written in simple language, often using abbreviations and containing typographical errors. This could significantly influence the dictionary we use for rule learning. In future work, we aim to improve the preprocessing of the text-based input data by applying large language models, for instance. Regarding this, Liu et al. [40] have recently introduced a very promising approach to correct errors in text documents.

Furthermore, regarding memory consumption, it is evident that we can significantly reduce memory consumption by applying the method introduced in this paper. Especially using the FOIL algorithm, we can observe that the memory consumption is reduced by about a third on all of the considered benchmarks. Using RIPPER, it seems that the

Fig. 1. Illustration of Accuracies shown in Table 1.

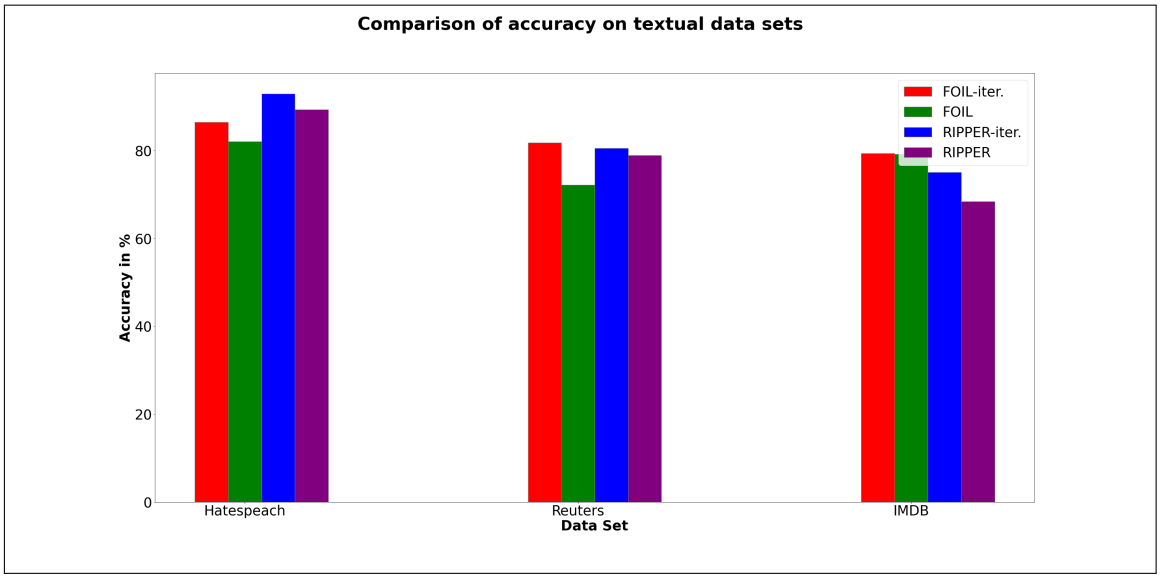
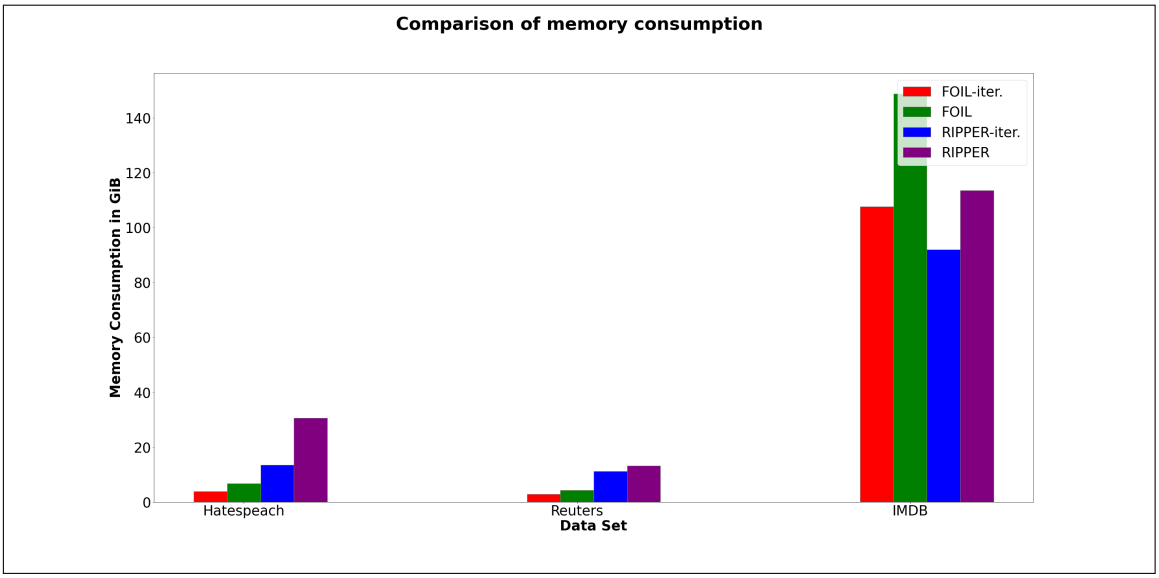


Fig. 2. Illustration of Memory Consumptions shown in Table 1. Note that the memory consumption illustrated for RIPPER applied on the *IMDB* data set corresponds to a reduced feature space compared to the application of FOIL.



reduction of memory consumption rather depends on the underlying data. While we notice a remarkable reduction of more than a half on the *Hatespeech* data set (where the two classes are mostly distinguishable by considering the occurrence of some swear words), the reduction of the memory consumption on the other two benchmark data sets is not that distinct but still clearly visible with about 20%.

Regarding time consumption, we did not investigate the differences between the two approaches in that detail but in general we observed an increased time consumption when RIPPER is applied within our approach, while our iterative approach could even reduce the run time using FOIL. For instance, on the *Hatespeech* data set using FOIL we observed a total time consumption of about 37 minutes compared to approximately 77 minutes corresponding to the classical method. On the other hand, applying RIPPER results in a total time consumption of about 19 hours compared to about 4 hours with the classical method. However, note that at the end of the day the introduced iterative approach is intended to extend our framework for a versatile application of rule learning methods we already established in previous work. In particular, in combination with the modular approach proposed in [12] the total time consumption can be reduced by a multiple when we apply parallelization. In order to do so, the reduced memory consumption achieved by the iterative approach introduced in this paper is extremely beneficial.

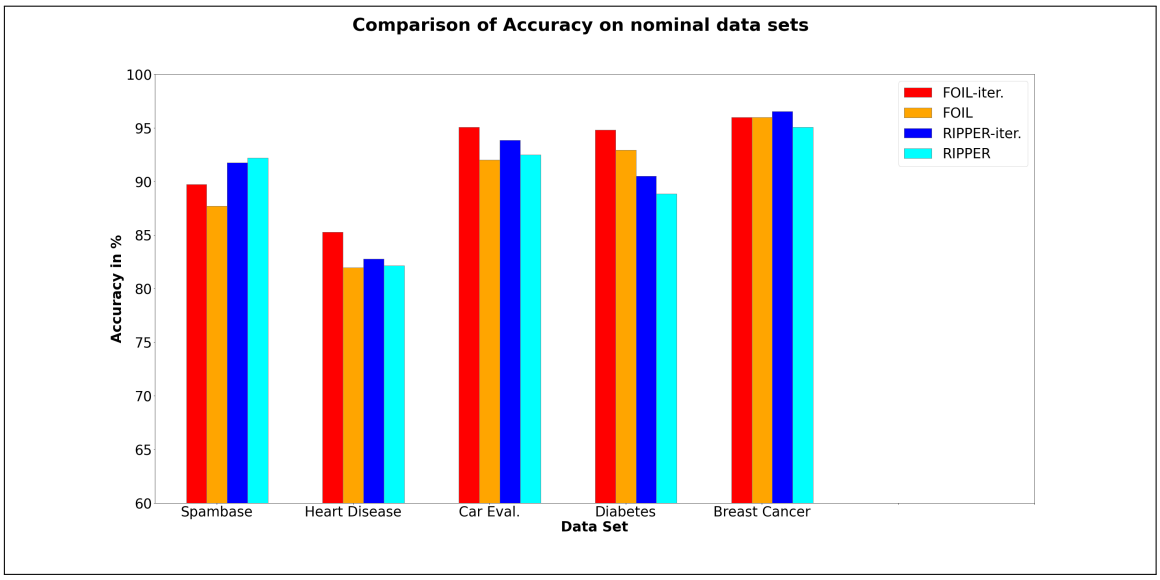
Additionally, we evaluate our iterative approach on some nominal data sets, as mentioned above. The corresponding accuracy is depicted in Table 2 and Figure 3. As clearly visible, our approach yields also significantly improved results on most of the considered non-textual benchmarks and outperforms the classical method by up to 3.3%.

In summary, we can positively answer Questions **RQ1** and **RQ2**.

	Spambase	Heart Disease	Car	Diabetes	Breast Cancer
FOIL	87, 69	81, 95	92, 00	92, 94	96, 00
FOIL - iter.	89, 71	85, 29	95, 07	94, 80	95, 99
RIPPER	92, 18	82, 16	92, 47	88, 84	95, 08
RIPPER - iter.	91, 74	82, 78	93, 84	90, 49	96, 55

Table 2. Accuracy in % achieved by our approach on different non-textual benchmark problems. Note that *iter.* denotes the iterative approach introduced in this paper.

Fig. 3. Illustration of Accuracies shown in Table 2.



4.3 Use Case: Reimbursement

The *Allianz Private Krankenversicherung (APKV)* is an insurance company offering health insurance services in Germany. As previously mentioned, the inspiration for this work stems from a use case we worked on during a collaboration with this company. In our previous work [12,13], we have already described the use case at hand in detail. However, summed up, an insurance company regularly receives bills handed in by the clients asking for reimbursement. Automated processing of these bills is desired in order to lower costs and to gain an edge over the competition by reducing the time until the client receives the reimbursed money.

As decision making, in particular in this sensitive area, should be transparent to both parties, the operational use of black-box machine learning algorithms is often seen critically by the stakeholders and is in many cases avoided. As a consequence, rule learning achieving a comparable performance offers the desired advantage of explainability.

For our case study, we are focusing on *dental bills*. On those bills, the specific type of dental service per row on the bill is unknown but needed for deciding on the amount of refund. Especially differentiating between material costs and other costs is of crucial importance.

In collaboration with the APKV, we have been provided with an anonymized training data set consisting of nearly one million instances. As opposed to our previous work, where we only considered structured information on the bills such as cost, date and simple

engineered features, in this paper we especially aim to work with the textual data and make predictions based on the occurrence of certain words or word groups. Due to extensive memory consumption, we restricted our analysis to the 8000 most common words using FOIL and the 3000 most common words for RIPPER.

Initially, large language and transformer models like *RoBERTa* [41] have been applied to process the bills. Due to pending non-disclosure agreements, we cannot provide further details about the exact procedures.[20] However, these highly complex methods have been applied to a combination of both textual information and engineered features. The exclusive consideration of textual information has not been tested yet.

To investigate the benefits of applying our approach on real-world text data, we considered the textual information exclusively in our experiments. The inclusion of engineered features is left for future work, where we plan to bring everything together and apply a combination of all three of our introduced approaches (modular, voting and iterative) to all available features.

In the experiments conducted during the evaluation of our approach on the industrial use case, we primarily focused on the precision of fully satisfied rules and did not apply partial matching (cf. [42]), which is typically done during evaluation. This means that if no rule is completely satisfied for a considered example, no prediction is made, rather than checking how many conditions of each rule are fulfilled and predicting the label corresponding to the rule with the highest percentage of satisfied conditions.

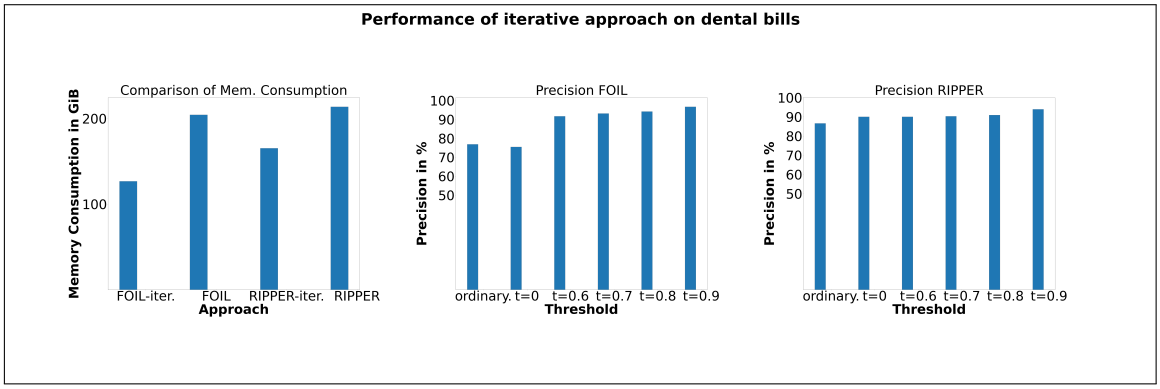
Summed up, by considering the results shown in Table 3 and Figure 4 we can answer Question **RQ3** as follows. Both the reduction in memory consumption and the increase in classification accuracy are clearly observable in the industrial use case involving dental bills. More precisely, considering FOIL, we can almost halve the memory consumption. Regarding the precision of the applied rules, the positive effect of the introduced *Value of Confidence* is evident. While the precision of our iterative approach without restrictions to the reliability of the applied rules is slightly smaller than the one achieved by the classical method, the application of a threshold in this context immediately improves the results enormously. For instance, using a threshold of 0.6 yields a precision (i.e., number of correctly predicted examples divided by the total number of examples where a prediction has been made) of nearly 92%, correctly predicting even more examples than the classical method. Further restricting the reliability of the applied rules and using a threshold of

[20] For more information please directly contact gabriela.dick_guimaraes@allianz.de.

Learner	Memory (GiB)	Threshold	Predicted	Correct	Precision (%)
FOIL	204,53		178.910	137.564	76,89
FOIL - iter.	126,78	0	232.476	175.515	75,50
		0,6	155.634	142.798	91,75
		0,7	150.601	140.339	93,19
		0,8	144.099	135.844	94,27
		0,9	119.635	115.697	96,71
RIPPER	213,82		106.230	92.041	86,64
RIPPER - iter.	165,37	0	150.538	135.590	90,07
		0,6	150.538	135.590	90,07
		0,7	149.166	134.722	90,32
		0,8	141.878	129.067	90,97
		0,9	84.815	79.702	93,97

Table 3. Performance of our approach on the reimbursement case study concerning dental bills. Note that *iter.* denotes the iterative approach introduced in this paper and the *Threshold* corresponds to the *Value of Confidence* of each rule meaning that rules with a reliability below the threshold are ignored.

Fig. 4. Illustration of Memory Consumption & Precision shown in Table 3.



0.9 yields a precision of almost 97%, while still predicting correctly about 115 thousand examples, which corresponds to approximately half of the test examples. This demonstrates that our approach makes it possible to automatically classify half of the dental bills with extremely high accuracy and – what is even more important – the resulting predictions are fully explainable.

Regarding RIPPER, similar improvements are observed: memory consumption is reduced by about a third and the precision increases from 86.64% achieved by the classical method to up to 94% obtained by our iterative approach using a threshold of 0.9. It is important to note that these experiments have been conducted with the general restriction of learning at most 10 rules for each label in both approaches. However, the classical method returned only 2-3 rules for 8 of the 10 labels due to the integrated early stopping according to the *description size* – a measure of total complexity of the model aiming to balance between minimization of classification error and minimization of model complexity. Using the same number of rules with our iterative approach, we can correctly classify 75401 examples from 83222 examples where one rule is satisfied. This corresponds to a precision of 90.60%, independent of the chosen threshold meaning that the generated rules all have a Value of Confidence of more than 0.9. Nevertheless, we decided to present the results of the 10 rules learned for each label using our iterative approach in Table 3 and Figure 4 because on the one hand this shows that the applied early stopping in the classical approach can sometimes be too restrictive and, on the other hand, it provides deeper insights into the effect of applying a threshold on the Value of Confidence for the rules used during evaluation.

In conclusion, our iterative approach outperforms the classical approach also on the industrial use case concerning both classification accuracy and memory consumption.

Moreover, the derived rules are highly useful, even for non-automated classification of such medical bills. They contribute to achieving more consistency and transparency in the decision making, and provide deeper insights into the data, in general.

4.4 Detailed Analysis

As a part of this paper, we have introduced a *Value of Confidence* that can be used as a metric for measuring the reliability of a generated rule. This section aims to investigate the influence of this value on the precision achieved during evaluation (cf. **RQ4**).

Data	Learner	Metric	$t = 0$	$t = 0.6$	$t = 0.7$	$t = 0.8$	$t = 0.9$
Hatespeech	FOIL	predicted	4312	3488	3430	3221	3083
		correct	3641	3347	3303	3164	3047
		accuracy	84, 44%	95, 96%	96, 30%	98, 23%	98, 83%
	RIPPER	predicted	4201	4201	4190	3951	3932
		correct	4084	4084	4075	3904	3886
		accuracy	97, 21%	97, 21%	97, 26%	98, 81%	98, 83%
Reuters	FOIL	predicted	1585	1498	1490	1477	1469
		correct	1357	1319	1314	1306	1300
		accuracy	85, 62%	88, 05%	88, 19%	88, 42%	88, 50%
	RIPPER	predicted	1713	1692	1684	1617	1302
		correct	1447	1433	1427	1376	1112
		accuracy	84, 47%	84, 69%	84, 74%	85, 10%	85, 41%
IMDB	FOIL	predicted	7560	7545	7545	7185	6434
		correct	6263	6252	6252	6009	5454
		accuracy	82, 84%	82, 86%	82, 86%	83, 63%	84, 77%
	RIPPER	predicted	7668	7668	6937	3433	1919
		correct	6034	6034	5501	2846	1697
		accuracy	78, 69%	78, 69%	79, 30%	82, 90	88, 43%

Table 4. Comparison of the classification outcomes considering only rules satisfying a certain level of reliability t measured by its *Value of Confidence*.

For this purpose, we apply thresholds t from 0.6 to 0.9 and consider only rules with a VoC $> t$. The corresponding results are shown in Table 4 as well as Figure 5 and 6. In this context, we only consider fully satisfied rules and do not apply partial matching, as explained in Section 4.3.

In order to answer question **RQ4**, we again illustrate for each of the considered textual benchmark data sets the number of examples where a prediction has been made (i.e., one rule is completely satisfied) together with the percentage of correctly classified examples. As expected, the number of classified examples decreases with an increasing threshold and the associated reduction in the total number of rules. However, as desired, the remaining rules are obviously more reliable and the percentage of correctly predicted examples consistently increases for both FOIL and RIPPER on each of the considered benchmarks. In conclusion, the incorporation of a *Value of Confidence* definitely has a positive impact on the precision of the made predictions.

5 Conclusion & Future Work

In this paper, we present an extension to classical rule learning methods, making use of a *Value of Confidence* as metric of reliability. This novel approach is especially suited for the application of rule learners on textual input data. However, the iterative approach is not only beneficial for gaining more control over the applied dictionary, but it has also shown to be advantageous for nominal data by optimizing the reliability of the generated rules in each iteration.

By combining the approach introduced in this paper with the two approaches to rule learning from our previous work, we obtain a framework for explainable classifications that can be applied in various scenarios handling different types of data in a production environment.

Concerning future work, we aim to integrate a more sophisticated preprocessing by applying, for instance, large language models to improve the choice of the dictionary. In the course of this, we will also investigate different ways of sorting the basic dictionary

Fig. 5. Illustration of Accuracy regarding FOIL shown in Table 4.

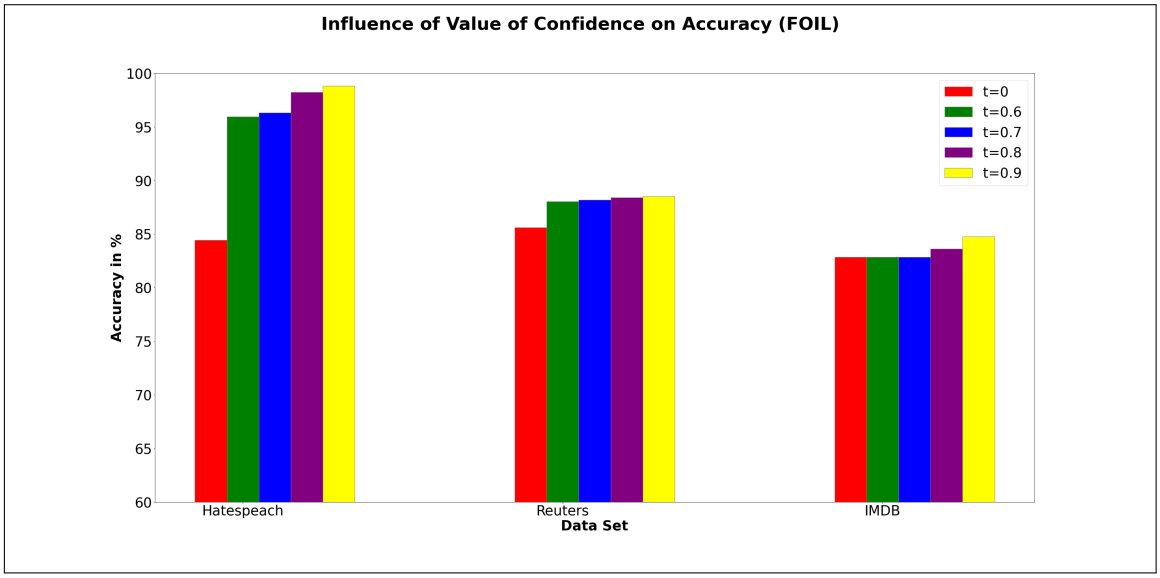
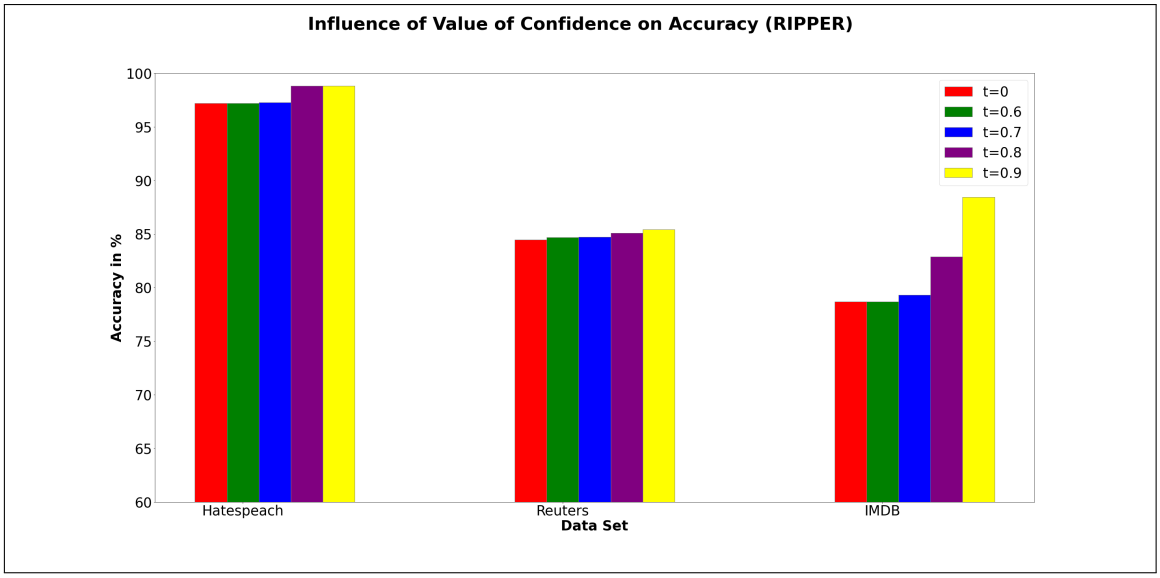


Fig. 6. Illustration of Accuracy regarding RIPPER shown in Table 4.



with the goal of finding the best possible initial dictionary to use in the first iteration of our approach. Moreover, using computer vision approaches in order to incorporate the position of words in a document might be another interesting consideration we aim to investigate in future work because especially in our main use case concerning reimbursement, the considered bills are mostly standardized and the crucial information is typically located in a specific area of the document.

References

1. Lu, Y.: Artificial intelligence: a survey on evolution, models, applications and future trends. Journal of Management Analytics **6**(1), 1–29 (2019). https://doi.org/10.1080/23270012.2019.1570365
2. Zhang, C., Lu, Y.: Study on artificial intelligence: The state of the art and future prospects. Journal of Industrial Information Integration **23** (2021). https://doi.org/10.1016/j.jii.2021.100224
3. Lee, R.: Artificial Intelligence in Daily Life. Springer (01 2020). https://doi.org/10.1007/978-981-15-7695-9
4. Angelov, P.P., et al.: Explainable artificial intelligence: an analytical review. WIREs Data Mining and Knowledge Discovery **11**(5) (2021). https://doi.org/10.1002/widm.1424
5. Ali, S., et al.: Explainable artificial intelligence (xai): What we know and what is left to attain trustworthy artificial intelligence. Information Fusion **99** (2023). https://doi.org/10.1016/j.inffus.2023.101805
6. Hulsen, T.: Explainable artificial intelligence (xai): Concepts and challenges in healthcare. AI **4**(3), 652–666 (2023). https://doi.org/10.3390/ai4030034
7. Fürnkranz, J., Gamberger, D., Lavrac, N.: Foundations of Rule Learning. Cognitive Technologies, Springer Berlin, Heidelberg (2012). https://doi.org/10.1007/978-3-540-75197-7
8. Gunning, D., et al.: Xai—explainable artificial intelligence. Science Robotics **4**(37) (2019). https://doi.org/10.1126/scirobotics.aay7120
9. Mitra, A., Baral, C.: Incremental and iterative learning of answer set programs from mutually distinct examples. Theory Pract. Log. Program. **18**(3-4), 623–637 (2018). https://doi.org/10.1017/S1471068418000248
10. Quinlan, J.R.: Learning logical definitions from relations. Mach. Learn. **5**, 239–266 (1990). https://doi.org/10.1007/BF00117105
11. Cohen, W.W.: Fast effective rule induction. In: Machine Learning Proceedings 1995. pp. 115–123. San Francisco (CA) (1995). https://doi.org/10.1016/b978-1-55860-377-6.50023-2
12. Nössig, A., Hell, T., Moser, G.: Rule learning by modularity. Machine Learning pp. 1–30 (07 2024). https://doi.org/10.1007/s10994-024-06556-5
13. Nössig, A., Hell, T., Moser, G.: A voting approach for explainable classification with rule learning. In: Artificial Intelligence Applications and Innovations. pp. 155–169. Springer Nature Switzerland (2024), https://doi.org/10.1007/978-3-031-63223-5_12
14. Cropper, A., Dumančić, S.: Inductive logic programming at 30: A new introduction. J. Artif. Int. Res. **74** (2022). https://doi.org/10.1613/jair.1.13507
15. Maas, A.L., Daly, R.E., Pham, P.T., Huang, D., Ng, A.Y., Potts, C.: Learning word vectors for sentiment analysis. In: ACL 2011. pp. 142–150. Association for Computational Linguistics, Portland, Oregon, USA (2011), https://aclanthology.org/P11-1015
16. Lewis, D.: Reuters-21578 Text Categorization Collection. UCI Machine Learning Repository (1997), https://doi.org/10.24432/C52G6M
17. Dua, D., Graff, C.: UCI machine learning repository (2017), http://archive.ics.uci.edu/ml
18. Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., Brown, D.: Text classification algorithms: A survey. Information **10**(4) (2019). https://doi.org/10.3390/info10040150,
19. Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., Gao, J.: Deep learning–based text classification: A comprehensive review. ACM Comput. Surv. **54**(3) (2021). https://doi.org/10.1145/3439726
20. Li, Q., Peng, H., Li, J., Xia, C., Yang, R., Sun, L., Yu, P.S., He, L.: A survey on text classification: From traditional to deep learning. ACM Trans. Intell. Syst. Technol. **13**(2) (2022). https://doi.org/10.1145/3495162
21. Gasparetto, A., Marcuzzo, M., Zangari, A., Albarelli, A.: A survey on text classification algorithms: From text to predictions. Information **13**(2) (2022). https://doi.org/10.3390/info13020083,
22. Mendez Guzman, E., Schlegel, V., Batista-Navarro, R.: From outputs to insights: a survey of rationalization approaches for explainable text classification. Front. Artif. Intell. 7 (2024). https://doi.org/10.3389/frai.2024.1363531
23. Altinel, B., Ganiz, M.C.: Semantic text classification: A survey of past and recent advances. Information Processing and Management **54**(6), 1129–1153 (2018). https://doi.org/10.1016/j.ipm.2018.08.001,
24. Johnson, D.E., Oles, F.J., Zhang, T., Goetz, T.: A decision-tree-based symbolic rule induction system for text categorization. IBM Systems Journal **41**(3), 428–437 (2002). https://doi.org/10.1147/sj.413.0428
25. Debole, F., Sebastiani, F.: An analysis of the relative difficulty of Reuters-21578 subsets. In: LREC'04. European Language Resources Association (ELRA), Lisbon, Portugal (2004), http://www.lrec-conf.org/proceedings/lrec2004/pdf/21.pdf

26. Pietramala, A., Policicchio, V.L., Rullo, P., Sidhu, I.: A genetic algorithm for text classification rule induction. In: Machine Learning and Knowledge Discovery in Databases. pp. 188–203. Springer Berlin Heidelberg (2008), https://doi.org/10.1007/978-3-540-87481-2_13
27. Pryzant, R., Yang, Z., Xu, Y., Zhu, C., Zeng, M.: Automatic rule induction for efficient semi-supervised learning. In: EMNLP 2022. pp. 28–44. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (2022). https://doi.org/10.18653/v1/2022.findings-emnlp.3,
28. Hitzler, P., Sarker, M.K.: Neuro-symbolic artificial intelligence: The state of the art. In: Neuro-Symbolic Artificial Intelligence (2021), https://api.semanticscholar.org/CorpusID:245698629
29. Wiegreffe, S., Pinter, Y.: Attention is not not explanation. In: EMNLP-IJCNLP 2019. pp. 11–20. Association for Computational Linguistics, Hong Kong, China (2019). https://doi.org/10.18653/v1/D19-1002,
30. Jain, S., Wallace, B.C.: Attention is not Explanation. In: NAACL 2019. pp. 3543–3556. Association for Computational Linguistics, Minneapolis, Minnesota (2019). https://doi.org/10.18653/v1/N19-1357,
31. Forman, G.: An extensive empirical study of feature selection metrics for text classification. J. Mach. Learn. Res. **3**, 1289–1305 (2003), https://dl.acm.org/doi/10.5555/944919.944974
32. HaCohen-Kerner, Y., Miller, D., Yigal, Y.: The influence of preprocessing on text classification using a bag-of-words representation. PLoS ONE **15** (2020), https://api.semanticscholar.org/CorpusID:218479987
33. Chen, W., Su, Y., Shen, Y., Chen, Z., Yan, X., Wang, W.Y.: How large a vocabulary does text classification need? a variational approach to vocabulary selection. In: NAACL 2019. pp. 3487–3497. Association for Computational Linguistics, Minneapolis, Minnesota (2019). https://doi.org/10.18653/v1/N19-1352
34. Patel, R., Garnelo, M., Gemp, I., Dyer, C., Bachrach, Y.: Game-theoretic vocabulary selection via the shapley value and banzhaf index. In: NAACL 2021. pp. 2789–2798. Association for Computational Linguistics (2021). https://doi.org/10.18653/v1/2021.naacl-main.223,
35. Japkowicz, N., Stephen, S.: The class imbalance problem: A systematic study. Intell. Data Anal. **6**, 429–449 (11 2002). https://doi.org/10.3233/IDA-2002-6504
36. Krawczyk, B.: Learning from imbalanced data: open challenges and future directions. Progress in Artificial Intelligence **5**, 221 – 232 (2016), https://api.semanticscholar.org/CorpusID:207475120
37. Spelmen, V.S., Porkodi, R.: A review on handling imbalanced data. In: ICCTCT 2018. pp. 1–11 (2018). https://doi.org/10.1109/ICCTCT.2018.8551020
38. Ha-Thuc, V., Renders, J.M.: Large-scale hierarchical text classification without labelled data. In: WSDM 11. p. 685–694. Association for Computing Machinery, New York, NY, USA (2011). https://doi.org/10.1145/1935826.1935919
39. Persia, C., Guimarães, R.: Riddle: Rule induction with deep learning. Proceedings of the Northern Lights Deep Learning Workshop **4** (2023). https://doi.org/10.7557/18.6801
40. Liu, R., Zhang, Y., Zhu, Y., Sun, H., Zhang, Y., Huang, M., Cai, S., Meng, L., Zhai, S.: Proofread: Fixes all errors with one tap. In: ACL 2024. pp. 286–293. Association for Computational Linguistics, Bangkok, Thailand (2024). https://doi.org/10.18653/v1/2024.acl-demos.27
41. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. (2019), https://api.semanticscholar.org/CorpusID:198953378
42. Grzymala-Busse, J.W.: A new version of the rule induction system lers. Fundam. Inf. **31**(1), 27–39 (1997), https://doi.org/10.3233/FI-1997-3113