OBOE: an Explainable Text Classification Framework

Raúl A. del Águila Escobar[1], Mari Carmen Suárez-Figueroa[2], Mariano Fernández-López[3] *

[1] Universidad Politécnica de Madrid (UPM), Boadilla del Monte (Spain)
[2] Ontology Engineering Group (OEG), Universidad Politécnica de Madrid (UPM), Boadilla del Monte (Spain)
[3] Department of Information Technology, Escuela Politécnica Superior, Universidad CEU-San Pablo, Boadilla del Monte (Spain)

* Corresponding author: r.delaguila@alumnos.upm.es (R. A. del Águila Escobar), mcsuarez@fi.upm.es (M. C. Suárez-Figueroa), mfernandez.eps@ceu.es (M. Fernández-López).

Received 18 October 2021 | Accepted 27 September 2022 | Early Access 3 November 2022

Abstract

Explainable Artificial Intelligence (XAI) has recently gained visibility as one of the main topics of Artificial Intelligence research due to, among others, the need to provide a meaningful justification of the reasons behind the decision of black-box algorithms. Current approaches are based on model agnostic or ad-hoc solutions and, although there are frameworks that define workflows to generate meaningful explanations, a text classification framework that provides such explanations considering the different ingredients involved in the classification process (data, model, explanations, and users) is still missing. With the intention of covering this research gap, in this paper we present a text classification framework called OBOE (explanatiOns Based On concEpts), in which such ingredients play an active role to open the black-box. OBOE defines different components whose implementation can be customized and, thus, explanations are adapted to specific contexts. We also provide a tailored implementation to show the customization capability of OBOE. Additionally, we performed (a) a validation of the implemented framework to evaluate the performance using different corpora and (b) a user-based evaluation of the explanations provided by OBOE. The latter evaluation shows that the explanations generated in natural language express the reason for the classification results in a way that is comprehensible to non-technical users.

Keywords

Explainable Artificial Intelligence, Explanation, Framework, Text Classification.

DOI: 10.9781/ijimai.2022.11.001

I. Introduction

As a consequence of the wide use of black-box algorithms and the need to provide the justification that supports a classification result, eXplainable Artificial Intelligence (XAI), set up as an initiative, is one of the most relevant research topics in the last years.

Conceptually, a text classification problem is no different from other classification problems, so the same ingredients are involved in solving the problem: data, model, users (final users or model developers) and the context of the classification problem. Therefore, the challenges and questions that text classification tries to answer from an XAI perspective are the same: the need to specify the reasons behind the decision of the model (why question), the context of the explanation (what for question), how the model arrived at a conclusion (how question) or the data and problem of the classification (what question). However, all these ingredients and questions are not being considering together in a system to provide meaningful explanations [1].

The aim of this paper is twofold. Firstly, we present a customizable framework called OBOE (explanatiOns Based On concEpts) for explaining classification of texts. This framework defines a workflow that can be customized and allows all the ingredients to play an active role in the classification process. Furthermore, these ingredients work together to answer the questions that allow the black-box to be opened for final users and model developers by defining the following features: (a) Explanation Generation Workflow (how, why): there is an explicit and defined workflow for generating meaningful explanations for the users; (b) Data as key ingredient (what question): data is not considered just an element used by the Machine Learning (ML) algorithm to classify or by an explainer to provide an explanation. The role of data is to drive the workflow both to classify documents and to obtain meaningful explanations; (c) Agnostic Model (why and how questions): this means that the approach can be used with different ML models in order to be able to choose the one that best suits the problem; (d) External Knowledge Integration (what question): additional resources, such as thesauri or ontologies, are used to enrich explanations; (e) Involvement of Users (what for, why and what): users are actively involved in the process of generating explanations; (f) Explanations in Natural Language (what for question): general purpose explanations are created in natural language; (g) Explanations based on Relevance of Terms (why question): the relevance of terms that appear in the vocabulary defined by the model developer is used to generate the explanations; and (h) Interchangeable Components:

Please cite this article as:
R. A. del Águila Escobar, M. C. Suárez-Figueroa, M. Fernández-López, "OBOE: an Explainable Text Classification Framework", International Journal of Interactive Multimedia and Artificial Intelligence, vol. 8, no. 6, pp. 24-37, 2024, http://dx.doi.org/10.9781/ijimai.2022.11.001

every component in the approach can be substituted by others performing a similar function.

Secondly, we introduce our first attempt to automatically generate explanations that (a) inform and help users to understand why a particular result is produced by a classification system, and (b) are easy to understand by the so-called final users, who have no technical knowledge. Apart from final or non-technical users, our work considers engineers or scientists who parametrize the model and the workflow (the so-called model developers [2]). Model developers and final users can interact to obtain custom explanations by, for example, choosing the vocabulary to be used in the explanations.

The rest of the paper is structured as follows: in Section II we provide an overview of the literature in the field of XAI. Section III describes the different components of the OBOE framework. In Section IV we present an example of implementation of the framework. Section V presents the validation performed over this implementation, the corpora used, and the results obtained; while in Section VI we explain the user-based evaluation performed over the explanations provided by OBOE as well as the main results of such an evaluation. Section VII exposes, as a recapitulation, the main contributions, and results of our work. Finally, Section VIII provides the conclusions and future lines of work.

II. Related Work

Text classification is still an open research task due to its inner complexity, the real-world applications (such as spam or sockpuppet detection, sentiment analysis, among others), or the emergence of new algorithms (such as deep learning based algorithms).

Researchers have used several approaches based on ML such as Support Vector Machine (SVM) ([3]-[4]), Decision Trees [5], ensembling algorithms and, recently and due to the good results obtained, deep learning based algorithms such as Gated Recurrent Unit (GRU) [6],Long Short-Term Memory (LSTM) [7] or novel approaches that use techniques such as the extended variational inference (EVI) framework to learn the finite inverted Beta-Liouville mixture model (IBLMM) [8]. SVM, ensemble learning or deep learning based models are considered black-boxes, because do not provide a clear interpretation on how the model conclude a result, but, on the contrary, obtain better performance in their evaluation metrics in different contexts and are being widely used. Therefore, XAI is nowadays one of the most relevant research problems [5], [9]-[11].

There are several categories to classify the results of the research work in the XAI field [9]-[10], [12]. The following categorization is proposed in the present work:

1. Model intrinsic approaches. White box models, such classification trees, fall into this category. The model itself is transparent or interpretable, so the user can infer why and how a result was obtained [9], [13].
2. Model specific approaches. These solutions aim to provide a justification based on the algorithm mechanism itself. As Adadi and colleagues stated [10], when a specific type of interpretation is needed, only models that fit that kind of interpretation can be used. These approaches can rely on techniques such as visualization, feature importance or rule extraction on the classification process inside a neural network, among others. For example, extraction of fuzzy rules from a neural network or a SVM [3], [4], the use of a heatmap to interpret a trained SVM model [14], or proposal of a method to decompose the classification decision according to the contribution of the input elements [4].
3. Model independent approaches. There are new techniques aimed to offer an explainable solution from any classifier. Two of the most well-known techniques are: (a) LIME [15], based on the idea of building linear models locally close to the predictions of a black-box model and their variations; and (b) SHAP values [16], which assigns to each variable used by the ML model an additive feature importance value for each prediction according to several desirable properties such as missingness, consistency, and local accuracy. These two approaches do not necessary consider the user in the process of explanation, nor what are the results are going to be use for. They can answer a question "why" or "how" using a post-hoc interpretation, but they do not integrate the users, the data, the model and the context in the process of performing a classification and retrieving an explanation.

Nonetheless, mimetic classifiers, proposed in the first decade of 2000's, do not rely on specific techniques or modifications to the algorithms, but on the workflow defined to classify and obtain a better understanding of the classification results. In essence a machine learning model acts as an oracle that label randomly created examples and then use a second comprehensible model [17], [18], [19]. This solution aims to provide an explainable and resource efficient approach to ensemble algorithms, but its main goal is not to provide a system integrating data, user, models or the context to generate explanations. It is also worth mentioning two recent frameworks: "A framework for explainable text classification in legal document review" [20], which identifies snippets of text that are relevant for the purpose of the review. It is a domain-specific framework which provides explanations based on examples, and it does not explore the relationship between the terms and the topic of the texts. On the other hand, "explAIner" [21] is an interactive and iterative framework based on Visual Analytics (VA) and Interactive Machine Learning (IML). This framework helps understand and refine ML models thus, the main objective of the framework is to build robust models and make them comprehensible for users while building the models. This framework involves three different kind of users but all of them have deep or partial knowledge on ML tasks. Although this framework is conceptually model agnostic, it is implemented using deep learning algorithms and its process is very tied to these kinds of algorithms.

Table I shows a comparative analysis performed over the above mentioned frameworks that generate explanations: Mimetic Classifier, Legal Document Review, and explAIner.

TABLE I. Comparison of Frameworks

Features	Mimetic Classifier	Legal Document Review	explAIner
Explanation Generation Workflow	○	✓	×
Data as key Ingredient	○	×	×
Agnostic Model	○	✓	○
External Knowledge Integration	×	×	×
Involvement of Users	×	×	✓
Explanations in Natural Language	×	✓	×
Explanations based on Term's Relevance	×	×	✓
Interchangeable Components	×	×	○
(*) Leyend:	✓ yes	○ partially	× no

None of the analyzed approaches considers at the same time the features present in Table I, which allow the ingredients involved in a classification task (data, model, users and context) to work together to answer the questions that are relevant for end users and developers to open a black box model (what, how, why). explAIner and mimetic classifiers do not define a workflow to generate

explanations, Legal Document Review is not a general purpose natural language explanations based on terms relevance and making data a key ingredient that can integrate external knowledge. None of the frameworks consider data as an ingredient that participates in the process from the beginning both to classify and to obtain explanations. OBOE is based on the idea that the vocabulary that discriminates a document must also be relevant to justify how that document has been classified. Conversely, our framework is designed to integrate external knowledge (Sections III and IV) to provide a general purpose explanation in natural language. Neither explAIner nor Legal Document Review follow the approach we are proposing in this research work to generate explanations, this is, to use natural language based on the relevance of the terminology used in the text. Finally, although explAIner conceptually is a modulable approach, it does not provide general purpose explanations, but to explain the model while building it.

III. Framework Description

We propose a framework called OBOE in which classification of a text can be explained (a) through the early identification of relevant terms in a corpus of documents and (b) from the machine learning techniques used to classify the several documents in the corpus. This justification is internally expressed as symbolic rules and presented in natural language to the user.

All the components defined in OBOE can be customized according to the needs or peculiarities of the classification and explanation tasks. For example, a user that needs to perform a classification over a large corpus can use embeddings to represent the documents and deep learning based algorithms such as LSTM to generate a classification.

In this section we present OBOE at the conceptual level describing its components and the inputs and outputs of each component. In Section IV we present our custom implementation of each component in OBOE to conduct the experiments detailed in Section V.

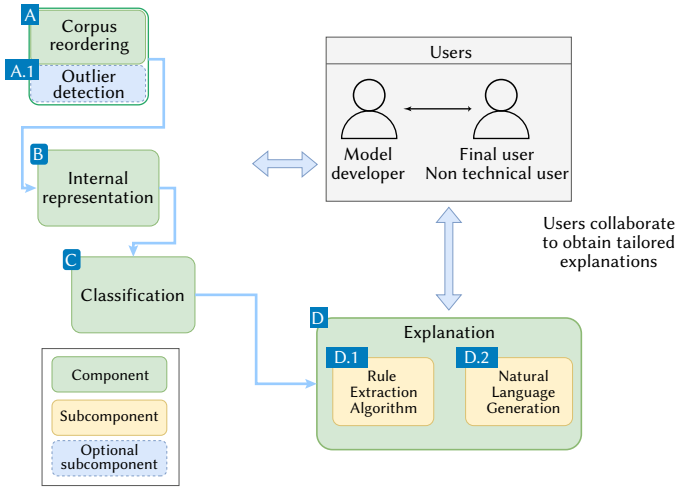


Fig. 1. OBOE components and workflow.

Fig. 1 shows OBOE workflow, where each box is a component of the framework. The components are the following: (A) corpus reordering to early identify relevant terms and documents related to topics, with an (1) optional outlier detection sub-component to create an optional categorization of documents related to the topic we want to discriminate (our target variable), (B) internal representation generation of the corpus, (C) classification algorithm used, and (D) explanation creation by (1) the generation of rules that will be expressed in (2) natural language. OBOE is conceived as a customizable framework since the implementation of each component can be defined by the model developer, which means that the algorithms and techniques are interchangeable.

A. Corpus Reordering

In a traditional classification problem, either binary or multi-class, the corpus is correctly labeled. In others, on the contrary, the corpus of documents is not labeled completely, or is not labeled at all.

In OBOE, data (in our context, a corpus of documents) play a leading role in obtaining an explanation, since it contains the terms that discriminate one class from another. The aim of this component is to identify which terms of the documents that compose the corpus (input of this component) can be useful to explain from the beginning of the process the results of the classification problem.

In this sense, unsupervised techniques such as clustering or topic modeling with any number of clusters or topics can be used to assign new classes to documents and to use the vocabulary identified in such classes to ease the classification and the explanation results. The output is a labeled corpus after applying a technique such as topic modeling or clustering. This output will be later used either by the subcomponent "Outlier detection", or by the classification algorithm, and also the user (model developer with the collaboration of final users) can extract relevant terms to be used in the explanation component. The participant involved in this component is the model developer, who parametrizes the technique used to reorder the corpus.

1. Outlier Detection

This subcomponent is optional in the designed process of OBOE and can be used, for example, to transform a binary classification into a multi-label classification problem. This component helps to discriminate which documents are related to a given topic, but belong to another topic. For example, a user might be reading documents about Ancient Rome and then find in the corpus a review of the movie 'Gladiator'. Although this review is going to present words related to documents describing Ancient Rome and is somehow related to Ancient History, there are other words pointing out this is a different kind of text. To that aim, an outlier detection technique, such as Interquartile Range (IQR) or Isolation Forest [22], is applied to the reordered corpus (which acts as an input of this component), to identify documents that are similar to documents of a specific topic. The output of this component is a corpus with a new label assigned to those documents that are similar to other documents in one of the categories that belong to the corpus.

The participants involved in this subcomponent are (a) the model developer, who parametrizes the technique used to reorder the corpus and, (b) the final user who identifies the topic needed to discriminate.

B. Internal Representation

This component translates every document of the reordered corpus (input) to an internal representation (output) to carry out the classification task. In a text classification task, documents are processed (for example, removing special characters and stop words), and transformed into an internal representation that can be managed by a machine learning classifier. Some of the techniques used to preprocess the document are removing stop words and special characters, tokenization, stemming or lemmatizing the texts, among others. Also, common techniques used to generate an internal representation are translating tokens to identifiers a Document Term Matrix (DTM) based on Term Frequency, Term-Frequency Inverse Document representation or those based on embeddings [23].

The participant involved in this component is the model developer that chooses the algorithm accordingly to the problem needs.

C. Classification

This component is a classification algorithm (for example, SVM), and some technique aimed to obtain the variable importance by means of other post-hoc techniques such as LIME or the variable importance identified by any ML framework. The input of this component is the internal representation that will be used by the classification algorithm. The output of this component is a model (the object that represents the algorithm trained) and a representation of the relevance of the words for the classification task.

The participants involved in this component are the model developer and the final user who helps the model developer to parametrize the component and can identify which words can be useful for the explanations.

D. Explanation

This component explains the results obtained during the classification process. To this end, this component is composed of two subcomponents: 1. Rule Extraction Algorithm and 2. Natural Language Generation.

Rules (subcomponent D.1) are contrastive and transparent explanations that can be translated to natural language to exemplify why that text has been classified with a specific class. This eases the collaboration between model developers and final users. Also, rules can manage the importance of the terms in the corpus. This importance of the terms in the documents can be captured in a DTM, regardless of whether an embedding and neural network-based approach had been used for classification or whether knowledge resources are used to enrich the rules or the subsequent natural language explanations.

A natural language explanation (subcomponent D.2) is one of the most straightforward methods to clarify a justification. To generate a natural language explanation a model developer can use a fine tuning approach based on transformers such as T5 [24], or to create and/or to translate the rules into natural language.

To generate the explanations, the explanation component can use as an input the vocabulary identified by the corpus reordering component, the most relevant words identified by the classification framework or another post-hoc technique such as LIME or SHAP [14][15], and the internal representation generated by the internal representation component. This document retrieves a natural language explanation based on rules as an output.

The participants involved in this component are the model developer, who parametrizes algorithms used to obtain the rules and the natural language explanation and the final user who identifies the vocabulary, the size of the rules or the specific text of the explanations, among others.

IV. Framework: Custom Implementation

This section covers the custom implementation made to carry out the experiments described in Section V. As we explain in detail in Section V, we address a binary classification problem and therefore our customized components are conceived to solve this problem.

A. Corpus Reordering: Custom Implementation

As stated in Section III, terms appearing in the corpus (input of this component) are key elements to classify the documents and to explain that classification. In OBOE, these terms help to early discriminate the subject of the text (the class to which the text must be classified, also referred in this paper as "positive class") at the beginning of the process, and to later classify and explain the results.

Positive Unlabeled Learning (PUL) is one of the techniques that fits into this scenario. PUL [25]-[27] does not require a fully supervised corpus with positive and negative texts. Instead, PUL uses positive and unlabeled datasets to early discriminate which texts (words) belong to the positive class and their associated probability.

The custom implementation follows a two-step method composed of Topic Modeling with a Latent Dirichlet Allocation (LDA) [28]-[29]. LDA is an unsupervised technique based on Dirichlet probability distribution. Documents are represented as bags of words, from which another one is generated in such a way that each document is a probability distribution of topics, and each topic is a probability distribution of words. In LDA, each document is a mixture of several topics described by a probability distribution that defines how likely each word will appear in each topic.

We apply LDA setting the number of topics equal to 2, as we performed a binary classification of the text. After applying LDA algorithm over the dataset, we obtain:

1. Documents belonging to negative or positive classes identified by LDA, referred as "topic 0" and "topic 1" respectively.
2. The words that likely belong to each topic following the estimated distribution by LDA.

Finally, the output is a corpus of documents labeled as "topic 0" and "topic 1" and their probabilities to belong to the labeled class. Also, we have the probabilities of every term to belong to "topic 1" or "topic 0". Fig. 2 shows a visual example of this component. There are positive and unlabeled documents in a corpus as an input that, after applying the LDA algorithm, are labeled as "topic 0" and "topic 1" with a certain probability. In addition, the different terms appearing in the documents are related to such topics with probability.

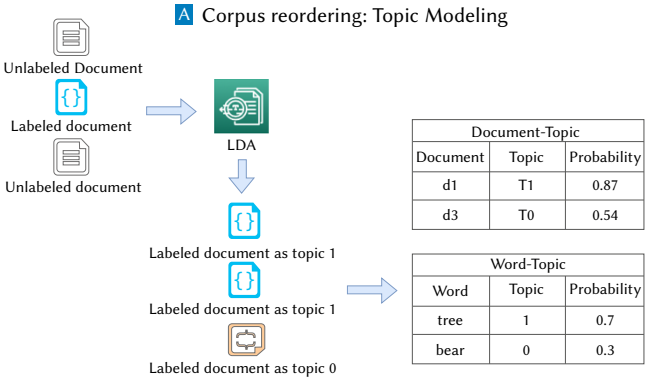


Fig. 2. Example of Corpus reordering.

1. Outlier Detection Subcomponent: Custom Implementation

We implemented this optional subcomponent in our custom implementation. The input we used was the labeled dataset retrieved by the Topic Modeling main component.

In the context of this work, documents that are related to the positive class might be considered as outliers according to the probability given by LDA of the document belonging to negative class. Therefore, we used the lower bound of an Interquartile range to detect and, therefore, classify the outlier documents.

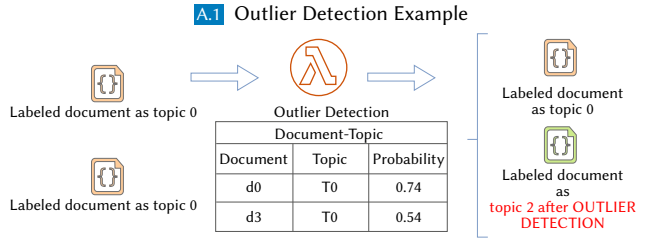


Fig. 3. Example of Outlier Detection subcomponent.

The output of this component is a corpus of documents classified into a positive class ('topic 1'), negative class ('topic 0') and positive-related class ('topic 2', the outliers). Fig. 3 illustrates an example in which Document 3 (d3), which had a low probability to belong to "topic 0", is labeled as "topic 2".

B. Internal Representation Generation: Custom Implementation

This component aims to transform the corpus of documents (input), which is labeled as positive and negative classes and, optionally, with a positive-related class, into an internal representation that can be managed by any classification algorithm

In the context of text classification, a DTM is one of the possible representations that can be used to solve the problem. A DTM is a corpus representation in which rows represent the documents, each column contains a term and cells show a metric about the relative relevance of the term in the document or in the corpus. This metric may be the TF-IDF, calculated as the product of Term Frequency (TF) and the Inverse Document Frequency (IDF):

1. The TF (Term Frequency) is the frequency of the term 't' in a document 'D'.
2. The IDF (Inverse Document Frequency) is the logarithm of the inverse occurrence rate of the term in the corpus.

The DTM metric used in this module is the inter class dispersion scheme [30], a variation of the DTM explained above, used to enhance the relevance of certain terms with respect to its class.

An inter-class dispersion scheme adds a new term to the equation called Dispersion (D(t)). This term of the equation will be low if the term is distributed uniformly among classes, so it helps to identify which terms are 'good' for classification. The scheme is described in Equations (1) and (2) [30]:

$$D(t) = \frac{1}{n}\sum_{c=0}^{n}((F(t,c) - \frac{1}{n}(\sum_{c=0}^{n} F(t,c)))^2 \quad (1)$$

$$Weight(i,j) = TF(i,j) * IDF(i) * D(i) \quad (2)$$

Equation (1) describes the inter-class dispersion coefficient of the term t. In this equation, "n" is the number of classes and F(t,c) is the number of documents having the term "t" and belonging to the class "c". Equation (2) represents the weight of the ith term in the jth document.

The output of this component is a DTM. Also, a binning process is applied to obtain a binned DTM. An example is shown in Fig. 4.

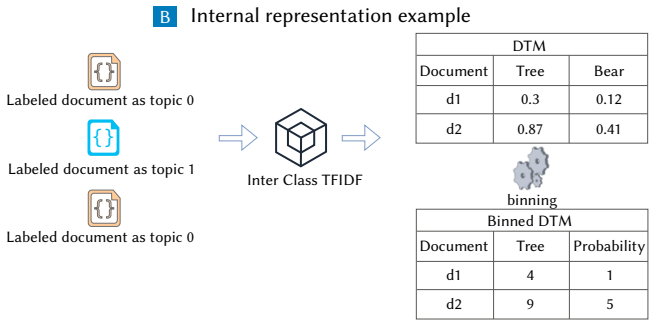


Fig. 4. Example of Internal representation.

C. Classification: Custom Implementation

In the implementation we made of this component, we used algorithms based on bagging (the implementation of the Random Forest [31] algorithm by H2O[1], hereinafter DRF) and boosting (XGBoost, hereinafter XGB) [32]. These algorithms used the DTM obtained in the previous component as an input. Once the training is complete, our output is a model that can be used to classify documents and the variable importance identified by H2O. Fig. 5 shows an example of this component.

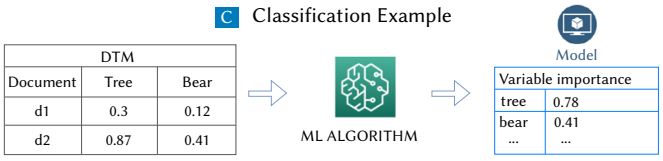


Fig. 5. Example of Classification.

D. Explanation: Custom Implementation

Our custom implementation of the explanation component comprises two subcomponents: (1) Rule Extraction Algorithm and (2) the Natural Language Generation, which creates a natural language based explanation in Spanish and English. In the custom implementation we made of OBOE, the algorithm generates a rule set using the terms that discriminate a topic to later generate a natural language explanation that uses WordNet[2] to ease the understanding of the explanations adding definitions that facilitate the disambiguation of the term. The rule set with WordNet definitions will finally result in general purpose explanations in natural language according to the needs of final users.

Subsections D.1 and D.2 explain the custom implementation we made of the subcomponents Rule Extraction Algorithm and Natural Language Generation. Fig. 6 and Fig. 7 show an example of the process. The binned DTM in terms of relevance, the variable importance and the probabilities relating words with topics act as input of the component. Then, model developers apply a Rule Extraction Algorithm to obtain a rule set (Fig. 6). This rule set is translated into natural language, adding information of an external knowledge resource (Fig. 7).

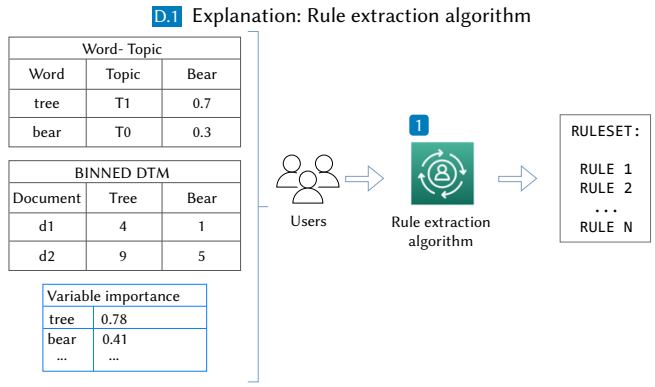


Fig. 6. Example of Explanation: rule set generation.

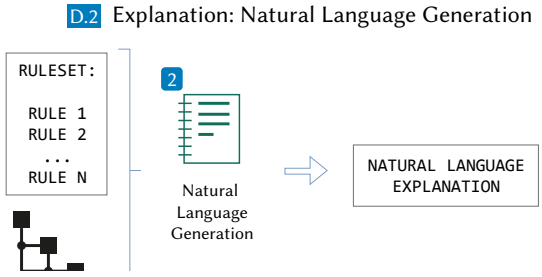


Fig. 7. Example of Explanation: natural language explanation.

[1] https://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/drf.html

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

1. Rule Extraction Algorithm: Custom Implementation

The input of this subcomponent is the DTM, a vocabulary that can either be specified by the user of this framework (or all the terms of the DTM), as well as other parameters, such as the minimum term relevance to be considered, the length of the rules or the number of rules, among others as presented in this section.

As mentioned in Section I, OBOE is not intended to fit to a specific ML model, but to try to ease the explanation of any possible model. For this reason, in this first subcomponent we generate a rule set. A rule set is a transparent method that can ease the understanding of a classification result and ease the collaboration between model developers and final users. To obtain a set of rules that helps to explain the results of any ML model classification, we implemented a Rule Extraction Algorithm, which is presented as Algorithm 1.

Intuitively, for any ML model, there are variables, that isolated or in combination with others, explain better the target variable. In the context of Natural Language Processing (NLP), these variables are terms whose relevance in the document is defined by a TF-IDF scheme. Instead of using the values of the inter-class TF-IDF weight scheme of the DTM, we used the binned values of each column of the DTM. These values still represent the importance of a term, but discretized in such a way that is easier to map these values to a grade of importance in natural language.

Algorithm 1 uses the binned relevance to generate a rule set with two main steps:

1. Identify the most relevant terms to explain a classification. These terms are chosen according to their information gain with respect to the target.
2. Identify the relevance values of each term in every rule being constructed. This value of relevance is the one that: (i) either minimizes the relation of negative examples vs. positive examples or (ii) increases the number of positive cases with respect to a previously selected value of relevance for that term by a percentage (called Growth Factor).

Both model developers and final users have an important paper to ease the explanations using Algorithm 1, by specifying the input parameters of the algorithm. These parameters define the output of the algorithm that is a set of rules to be translated into natural language. The input parameters are:

1. Vocabulary (V): Algorithm 1 is based on the idea that the vocabulary obtained by the Topic Modeling along with the one obtained as feature importance of the ML model or even the knowledge of the domain of the final user, can explain the results of the classification process.
2. Number of rules (N) and rule length (L): Rule length or coverage might be a drawback in terms of interpretability [3]. To overcome that problem, the algorithm user can specify these parameters to obtain a meaningful rule set.
3. The minimum relevance (min_rel): to enforce the algorithm to use values of a minimum relevance, the user can specify a minimum relevance parameter
4. Growth Factor (GF): this parameter controls the coverage of the rules. With a very large growth factor, the algorithm may result in rules explaining very specific cases (for example, a rule that identifies two or three positive cases with no negative cases), whereas a very small growth factor may include rules that are rather generalist. In our experiments we set up this parameter to 5%.
5. The maximum number of repetitions of terms in the rule set (MR): As the algorithm is based in its first step in the information gain of the term with respect to the target variable, the user can specify the maximum number of repetitions of the term in the rule set to introduce variability.

The Rule Extraction Algorithm (Algorithm 1) is as follows:

Algorithm 1: Rule Extraction Algorithm

I: Document Term Matrix (DTM), number of rules (N), length of rules (L), max repetition per term (MR), vocabulary (V), target variable (T), growth_factor (gf), min_relevance (min_rel)

O: Set of rules

begin Initialize variables:

1 if V is Null then V ⟵ get column names of DTM
2 positive_examples ⟵ Number of rows of DTM with T = 1
3 number_of_rules=1; length_of_rules=1; rule_set= empty set; DTMoriginal = DTM;
4 while positive_examples>0 and number_of_rules <=N:
5 to_exclude ⟵ empty list
6 while length_of_rules<=L:
7 attr ⟵ getVariableWithMaxInfoGain (V,T,DTM,MR,to_exclude)
8 val ⟵ getValueOf Term(attr,DTM,GF,min_rel)
9 Coverage ⟵ getPositivesRate(attr,val,DTMoriginal)
10 to_exclude.add(attr)
11 r.add(attr,val,coverage)
12 length_of_rules⟵ length_of_rules + 1
13 positive_examples, DTM ⟵ filterDTM(DTM, r)
14 rule_set.add(r)
15 number_of_rules ⟵ number_of_rules + 1

End return rule_set

Steps 7 and 8 are the two main steps of Algorithm 1: get attribute that maximizes the information gain with respect to the target, and for the selected term, get the value (that must be greater than min_rel) that minimizes negative vs. positive cases or that increase the positive cases in a Growth Factor. Therefore, the algorithm calculates the number of positive examples against the total to get the coverage of the rule in the Step 9.

The algorithm will stop when the number of rules is equal to the number of rules specified by the user or when there are not more positive examples to cover.

After creating the rule set, a consolidation step is applied. This step identifies rules with the same antecedents and different values to perform a simplification of the rule set, and obtaining the coverage of the new rule. Fig. 8 depicts an example of the left hand side of several rules and its consolidated form.

```
book == 4 and read == 8
book == 4 and read == 7
book == 4 and read == 9
```

(a)

```
book == 4 and ((read>=7 and read<=9))
```

(b)

Fig. 8. Part (a) shows a bunch of left-hand side rules; part (b) is the left hand side of the new rule consolidated.

Finally, a rule is created with in the form **"`if LHS then coverage is 0.X`"**, in order to improve its understandability, in the Natural Language Generation component, the rules are translated into the form: **"`if LHS then is a '<name of the class>'`"**. In this point, information about the coverage and the rate of negative over positive examples that the ruleset covers is added for the model developer's knowledge.

This Rule Extraction Algorithm can be thought as a simplified variant of IREP [33], but differs in the use of Mutual Information and in the fact of that it is not thought to maximize any accuracy metric, but to ease the understanding of any classification.

The result of this subcomponent (output) is a set of rules that will be translated into natural language by the next subcomponent.

2. Natural Language Generation: Custom Implementation

The set of rules obtained by the Rule Extraction Algorithm component (input) will be used to generate a natural language translation from each rule (output). The subsequent translation into natural language is structured in such a way that the results of the classification can be understood by any user, regardless of his/her previous knowledge of the domain.

To generate the natural language explanation from the rules, we used a Context-free Grammar (CFG). The values indicating the relevance of the term in the text and corpus are translated into natural language in terms of importance, as shown in Table II.

TABLE II. Scale of Importance

Relevance	NL Translation – Spanish	NL Translation - English
0-2	Muy poco importante	Very unimportant
3-4	Poco importante	Unimportant
5-6	De Importancia media	Of medium importance
7-8	Importante	Important
9-10	Muy importante	Very important

The rules parsed using the context-free grammar create a natural language explanation that is completed using WordNet with the information retrieved by the definitions of the concepts that are contained in the rules. Appendix I shows an example of explanation in Spanish and the same one translated into English.

V. Evaluation of the Custom Implementation of OBOE

We used three corpora to perform the experiments on the custom implementation of OBOE. These three corpora were all processed using the same pipeline by removing stop-words, special characters, and lemmatizing the words:

1. Amazon review corpus [34]. It contains the review of near 6 million objects (books, cell phones, etc.), divided into several categories such as Books, Electronics, Cell Phones & Accessories, Office Products, or Home & Kitchen, among others. We randomly selected 25.000 Books reviews and 25.000 reviews of other categories, labelled as positive class the 25.000 Books reviews.
2. Reuters dataset [35] included in the NLTK package[3]. It contains 10.788 documents from the Reuters Financial Newswire Service divided in 90 categories, although a document can belong to more than one category. The documents having the class 'acq' are the positive class.
3. 20 Newsgroup dataset [36], included in the scikit-learn package4. It contains 18000 documents divided into 20 categories. The documents belonging to category number '3' ('comp.sys.ibm. pc.hardware') are the positive class.

We present in this section the results obtained in the Amazon corpus, whereas the results obtained with the other two corpora are detailed in Appendices II and III. This section presents the results obtained in the experiments to evaluate whether LDA can discriminate the two possible categories of each target variable (one per corpus), and the classification performed by XGB and DRF. The specific algorithms and techniques can be changed by others if they fit the workflow defined by OBOE. The implementation employs Python programming language, scikit-learn5, spacy6, NLTK , and H2O.

A. Evaluation of Corpus Reordering

We performed an evaluation on the results obtained by LDA, comparing the topic assigned by LDA with the actual topic of the documents. We used the probability distribution for the documents belonging to "topic 0" or "topic 1" as the predictor variable, and then we trained a XGB model using the real target as variable to predict. The hyperparameters used in the XGB models are based on the results retrieved by Bayesian Optimization [37].

Appendix II contains the performance, the hyperparameters and the confusion matrix obtained in the three corpora.

The performance obtained in terms of Area Under Curve (AUC) and Kappa index were w 0.89 and 0.72, respectively, in the Amazon corpus. In addition the error rate of the positive class was 0.07 and the error rate of the negative class, 0.2.

B. Evaluation of Classification Models

We conducted two experiments to evaluate the classification component: the first one evaluates the assignment of topics to each document, without outlier detection subcomponent, and the second one uses the reordered corpus generated by the outlier detection subcomponent that creates a third class with documents that might be related to the positive class. We used XGB and RF to classify the documents and Bayesian Optimization to find the hyperparameters used in each of the trained models. Appendix III contains the tables detailing the performance and confusion matrix.

Regarding the classification experiment without outlier detection, XGB and RF obtained a performance of 0.99 and 0.964, respectively.

In the case of the results of classification with outlier detection, we used an IQR [37] approach over the probability distribution, with an IQR of 1.5. To perform the search of hyperparameters, we used hyperopt[7] package, letting the library to choose the best algorithm to perform the optimization. We used the LogLoss error as the cost function. The LogLoss of the classification performed by DRF and XGB was 0.32 and 0.36, respectively. It is worth mentioning that only the Reuters dataset obtained an error rate lower than 0.5 in class 2 (documents related to main topic). In most of the cases, the classifier erroneously classified these documents as positive class. Nonetheless the error rate obtained in Reuters suggests that the approach seems to be valid (see Appendix III).

C. Rule Extraction Algorithm

For each dataset, we obtained two set of rules with the Rule Extraction Algorithm defined by Algorithm 1. In the first one, the maximum number of rules must be 10. Besides, in the second one, the maximum repetitions per term is three and the vocabulary was also provided.

Specifying the vocabulary and the maximum repetitions per term permits the model developer, interacting with the final user, to obtain a compact set of rules that ease the further understanding of the results achieved during the classification. Therefore, the framework involves the user in all the process through an analysis of the most relevant terms identified by the components. Also, users might include vocabulary if they have previous knowledge of the domain. Fig. 9 presents a sample of rules obtained for the

[3] https://www.nltk.org

[4] https://scikit-learn.org/stable

[5] https://scikit-learn.org/0.19/datasets/twenty_newsgroups.html

[6] https://spacy.io

[7] https://github.com/hyperopt/hyperopt

Amazon corpus. Fig. 9a shows an example specifying the maximum number of rules, whereas Fig. 9b shows an example specifying the vocabulary:{'word','book','romance','story','author','character','read'}.

```
if book==3 and ((read>=7 and read<=9) then is a 'Book Review'
```

(a)

```
if ((author>=8 and author<=10)) then is a 'Book Review'
```

(b)

Fig.9. Part (a) is shown a bunch of left hand side rules; part (b) is the left hand side of the new rule consolidated.

It is also worth mentioning that two different classifiers can obtain a variable importance whose similarity may be influenced by factors such as the inner logic and the parameterization of the algorithms, among others. This may affect the vocabulary that the model developer uses to generate the rules and the natural language explanation. Also, the model developer can consider the use of other techniques to obtain the 'relevance' of the term to the prediction of the target variable. As a matter of example, Appendix IV contains 4 figures depicting the variable importance obtained by DRF and XGB algorithms both in the Amazon Reviews and Reuters corpora. These figures show that the variable importance obtained by both algorithms in the Amazon Review corpus is more similar than the obtained by the algorithms in the 20 Newsgroup corpus. For this reason, the collaboration of model developers and final users is important to obtain meaningful explanations.

VI. User-based Evaluation on Explanations

To gather subjective empirical data about the explanations provided by OBOE, a questionnaire was implemented as a Google Form[8] and launched via different mailing lists, posts in LinkedIn, and tweets to obtain responses from general population. We collected 38 responses and removed two of them from the analysis because of the inappropriate responses about the professional affiliation.

This questionnaire was divided into three subdivisions: the first one focused on questions about the level of comprehensibility (ease of understanding) and legibility (ease of reading) of the explanations generated by OBOE, the second one devoted to get suggestions and recommendations, and the final part related to demographic data.

More specifically, the first subdivision of the questionnaire included two parts: (a) one related to the level of comprehensibility of explanations provided by OBOE, and (b) another one related to the level of legibility of such explanations. In the case of (a), the questionnaire included three questions based on Cloze Test [39]. The idea behind these types of questions was to analyze whether the whole text, with removed elements, is understood, and thus, the removed elements are filled in with logical language items. In the case of (b), the questionnaire included three questions based on the idea of binary forced choice [40]: "humans are presented with pairs of explanations and must choose the one that they find of higher quality (basic face-validity test made quantitative)." Each question is divided in several sub questions to evaluate key parts of the explanation provided by OBOE, such as the appropriate selection of the terms and the alternatives to the redaction provided by OBOE. The total amount of sub questions is 16, so the total amount of responses analyzed in the first part of the questionnaire are 576. With this type of questions, we can conclude whether the explanation pattern is comprehensible and whether there are patters which are preferred for an explanation.

According to the questionnaire, the 65.8% of the respondents were in an age range between 31 and 45 years old and have a university degree, whereas the 25.8% of the respondents are PhD. The profession of the 71.1% of the respondents is related to computer science.

[8] https://forms.gle/jsGC5DjfzWveVnz69

A. Comprehensibility

The three questions of this questionnaire part were related to three different explanations: first and second questions were related to a short and a large explanation respectively, and the third question was related to que adequateness of the term 'importance' versus 'relevance'. The total amount of sub-questions in the comprehensibility section is 12. Fig.10 shows an example of question of this part:

> Cuando (1) [__Hueco1__] 'book' tiene importancia media en [__Hueco2__] texto, (2) [__Hueco3__] 'read' tiene importancia media en [__Hueco4__] texto y (3) [__Hueco5__] 'author' tiene importancia media en [__Hueco6__], entonces el texto analizado [__Hueco 7__]: 'Book Review'. Teniendo en cuenta que: 'Read' [__Hueco8__] como (a) "algo que se lee", [__Hueco9__] [__Hueco10__] "lectura", (b) "interpretar algo que está escrito o impreso", [__Hueco11__] [__Hueco12__] "leer", y (c) "tener o contener una determinada redacción o forma" [__Hueco13__] [__Hueco14__] "decir".

Fig. 10. Cloze test question from the Comprehensibility part.

The 6.9% of the responses gathered in this section are incorrect. These incorrect answers were written by the respondents, without being an option in the Cloze's tests.

The rest of the responses gathered are as follows:

- The 38,2% correspond to the actual explanations provided by OBOE.
- The 41,4% correspond to an alternative that it is synonym of the explanations provided.
- 13,4% are valid responses provided by the respondents.

There are several aspects remarkable in this section:

1. The explanations provided by OBOE do not exactly translate a rule of the form "if - then". In this sense, in the questions where the users have to choose between the alternatives: "Cuando" (whenever), "Si" (if) or an alternative response provided by the user, the majority of the responses retrieved correspond to the option "Cuando", which is the same as OBOE provides. On the contrary, in the subquestion where the respondents have to choose between the wording provided by OBOE "podría tratar de" (may be related to) or the alternative "se clasificaría como" (would be classified as), the majority of the respondents preferred this second option which can be considered more technical from a computer science point of view. Analogously when we asked the respondents to choose between "acepción" (meaning or connotation) or "sentido" (sense) the first one was the most voted, although is less technical from a computer science perspective
2. In the same way, there are two sub questions where the respondents must choose between "importance" or "relevance". In one of them, the majority option selected was "importance" whereas in the second one was "relevance".

B. Legibility

There are four sub-questions in this questionnaire part aimed to analyze different form of redactions. Fig. 11 shows an example of these sub-questions.

The first three sub questions analyze the form of presenting the redaction according to the categorization of a text. The several redactions provided can be divided into two alternatives:

1. The alternative in the form 'if – then' and its derivatives. As said before, the explanation provided by OBOE falls into this group, although it is not exactly translated as an "if - then" question.
2. Another alternative in a less structured language from the point of view of explaining or translating a rule, where (a) the possible classifications of the text goes first and (b) the causes are in the last part of the phrase, for example:

"El texto analizado podría tratar de: Book Review porque el término "read" es importante en dicho texto" that can be translated into English as "The text analyzed could be related to the subject: Book Review because the term "read" is important in that text"

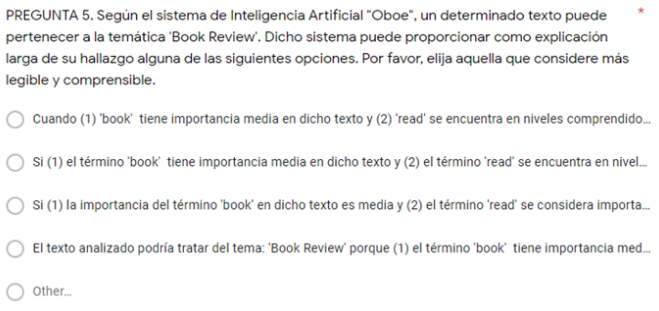


Fig. 11. An example question from Legibility part.

The 50% of the users chose an 'if-then' related alternative in the three sub questions. The 23,1% selected the alternative provided by OBOE and the 26.9% other alternatives using the prefix "Si" (if) instead of "Cuando" (whenever), what seems somehow contradictory with the results obtained in the comprehensibility part.

The last sub question analyzed the redaction of the definition of the terms. In this case there are five possible options:

1. The one provided by OBOE: Possible definition of <terms>: according to the sense <form>: definition
2. Definition of <term> according to the sense: <form>: <definition>
3. Definition of <term> according to the connotation: <form>: <definition>
4. <term> is defined as <definition> according to the sense <form>
5. <term> is defined as <definition> according to the connotation <form>

The explanation provided by OBOE in this case just obtained one vote. By contrast, most of the respondents chose the options where the term is introduced in the beginning (4 and 5) with 9 and 12 votes, respectively which can be considered as a more informal way to present the explanation.

Nonetheless, from the responses retrieved in this part, it is difficult to conclude whether users prefer an explanation presented in a more informal way, or not. The first three questions show a balance between the options provided in a more structured way and the more informal one. Even the sub question number four of this section cannot be considered unformal, but not as structured as the other options.

Also, analyzing these results with those obtained in the comprehensibility part, we can conclude that users do not prefer terminology related to plausibility, such as "podría tratar de" (may be related to) or "posible definición de" (possible definition of).

VII. Discussion

The results of the custom implementation of OBOE show that the proposed approach achieves comprehensible explanations of the classification process. The ingredients involved in the process of classification and generation of explanations, i.e. data, model, users, play an active role and become relevant to generate explanations. This is a novel approach that none of the analyzed frameworks uses., Furthermore, OBOE aims to define a workflow of interchangeable and customizable components to provide explanations in natural language that can be completed with external knowledge resources. Moreover, OBOE relies on the idea that the same terms which discriminate a document must be used to explain the classification of such a document. This approach eases the use of well known classification techniques and also allows the use of newer ones, which is crucial to allow a tailored solution to the context and user needs. Table III, summarizes our proposal in comparison to other frameworks.

TABLE III. Comparison of Frameworks

Features	OBOE	Mimetic Classifier	Legal Document Review	explAIner
Model	Independent	Intrinsic: needs a White Box classifier	Independent	Deep Learning
Explanations	Rules and Natural Language	No	Examples	Visual Analytics
Explanation Workflow (EW) /Workflow/ Tool (T)	EW	W	T	T
User Involvement	Model Developer and Final Users have the role of orchestrating the workflow	Model Developer	Model Developer	Model Developer
How data is used	Defines model vocabulary and explanations	As usual	As usual	As usual
Relevance of terminology	Yes	No	No	Yes
Customization	Yes	No	No	Partially
Can integrate external knowledge	Yes	No	No	No

While OBOE defines a customizable, model-independent workflow to obtain natural language explanations, none of the other frameworks analyzed fit into this framework. Besides, data and users become relevant during all the process:

- Vocabulary can be defined from the beginning and is based on the relevance of the terms with respect to the class you are trying to predict and explain. As a consequence, data is not used 'as usual', this is, just as the input of the classification process.
- Users are orchestrators of the workflow, specifying the parameters that lead to an ulterior explanation: with the importance of the terms obtained by the ML model and LDA, the user indicates which are the main terms that help explain the subject of the text. Then, both model developers and final users generate a rule set that ease the interpretation and comprehension of the results by any kind of user. So, the user (model developers and final users) is controlling how the explanations need to be obtained, which is an aspect related to the context.
- It is worth noting that Mimetic Classifier defines a workflow, but this workflow is not intended to generate explanations but to ease the understanding of the classification results using a white box classifier.

OBOE is flexible to use any ML model or optimization technique to better classify the documents, and this feature can lead to different results along the process; other frameworks such as Mimetic Classifiers or explAIner are tied to the machine learning technique used. In our

custom implementation we used PUL, so we assumed that we did not know the composition in terms of classes of every document of the corpus. Also, we could try the use of Topic Modeling instead of clustering which is not a common way to approach a PUL problem.

The use of specific techniques can lead to different results. In this sense, the variable importance obtained by a ML algorithm can vary, so the user can obtain a different vocabulary to generate the explanations. This feature is also crucial as not every algorithm does fit any problem. Also, different techniques can be used to improve the generation of rules, or to select the most relevant variables in the light of the results obtained in the model (LIME, SHAP, among others). This flexibility can also help users (model developers and final users) to adapt the explanation to the context, by integrating external knowledge or even using the visualizations to complete the natural language explanations.

Finally, the customization feature it is also crucial from a comprehensibility perspective. Results show in section VI that it is not even clear which language a user might prefer. The adaptability to any context and user allows the framework to adapt the language of the explanations to the context of final users and also use knowledge resources such as vocabularies or ontologies.

VIII. Conclusion

This paper presents OBOE, a text classification framework, which aims to provide meaningful explanations. Data, model, explanations and users are ingredients involved in the classification process that play an active role along the process of classification and generation of explanations. OBOE defines various components that can be customized according to the specific context, users and needs, both model developers and final users.

In order to show the customizable feature of OBOE, a specific implementation is presented based on LDA for corpus reordering, IQR for Outlier Detection, an inter-class dispersion scheme for DTM creation, XGB and DRF for classification and a custom Rule Extraction Algorithm and Context-free Grammar to generate general purpose natural language explanations.

We have performed three validations for the implementations we made of the components of OBOE using Amazon, Reuters and 20 Newsgroup corpora: (i) corpus reordering evaluation, (ii) the classification evaluation and the (iii) explanations evaluation. As our customization is based on PUL, the first validation shows whether the topics assigned by LDA algorithm match the actual ones; while the second one evaluates the classification results.

The corpus reordering and classification evaluation achieved an AUC of 0.89 and 0.9, and a Kappa Index of 0.72 and 0.95 in the Amazon corpus. The error rate of the positive class was 7.1%.

In case of the classification component, we performed two different evaluations: with and without outlier detection. We used XGB and DRF algorithms to perform the classification of the reordered corpora. The results achieved in these experiments show that the algorithms discriminate the reordered corpora. When considering the results obtained after Outlier Detection, in the case of Reuters dataset, we obtained and error rate in class 2 below than 0.5. This can be due to several factors, such as the need to use specific techniques for unbalanced multi class classification. Nevertheless, the results achieved suggest that this component is valid.

We also performed a user-based evaluation with the goal of determining whether the explanations provided by OBOE are comprehensible and legible. From the results obtained, we can conclude that the explanations generated by the custom implementation of OBOE are comprehensible, although there is not a clear preference between a more technical or informal language. In the same way, there is not a clear preference between a structured or more informal explanation when presenting the relevance of the terms, there was a tie between a "if-else" based explanation structure and a more informal one. Nevertheless, most of the survey respondents preferred a more informal choice when defining terminology.

Our current and future work is aimed to integrate semantic models in the explanations provided, using linguistic knowledge resources to perform knowledge-based translations and to generate explanations based on the familiarity of the user with a specific domain.

Appendix I: Example of OBOE explanation

A. Example of Explanation in Spanish

Explicación generada de uno de los casos encontrados donde un texto puede tratar de 'Book Review'

En el contexto de encontrar y justificar la temática de un texto, hemos podido deducir que cuando

(1) character se encuentra en niveles comprendidos entre tiene importancia media y es importante en dicho texto o (2) character es muy importante en dicho texto, entonces el texto analizado podría tratar del tema: 'Book Review'

Algunas definiciones de los términos arriba expuestos:

Mostrando 3 posibles definiciones para el término: <character>

0. Posible definición para <character>, de acuerdo con el sentido: calidad. Definición: una propiedad característica que define la aparente naturaleza individual de algo

1. Posible definición para <character>, de acuerdo con el sentido: característica. Definición: una propiedad característica que define la aparente naturaleza individual de algo

2. Posible definición para <character>, de acuerdo con el sentido: carácter. Definición: el complejo inherente de atributos que determina las acciones y reacciones morales y éticas de una persona

B. Example of Explanation in English

Explanation generated from one of the cases found where a text can be talking about the subject 'Book Review'

In the context of finding and justifying the theme of a text, we have been able to deduce that whenever

(1) *character* is at levels between being of medium importance and important in that text or (2) *character* is very important in that text, then the text analyzed could be related to the topic: 'Book Review'

Some definitions of the above terms:

Showing 3 possible definitions for the term: <*character*>

0. Definition of <*character*>, in compliance with the semantic meaning: quality. Definition: a characteristic property that defines the apparent individual nature of something

1. Definition of *'character'*, in compliance with the semantic meaning: characteristic. Definition: a characteristic property that defines the apparent individual nature of something

2. Definition of <*character*>, in compliance with the semantic meaning: *character* Definition: the inherent complex of attributes that determines a person's moral and ethical actions and reactions

Appendix II: Corpus Reordering Performance

A. Performance and Hyperparameters

Corpus	AUC	KAPPA	Hyperparameters	Under sampling
Amazon	0.89	0.72	max_depth: 39, ntrees: 393, min_rows:5, eta:0.6331, learn_rate:0.418, sample_rate:0.2508, colsample_bytree:0.633, reg_lambda:0.86, reg_alpha:0.062	No
Reuters	0.90	0.66	max_depth: 4, ntrees: 400, min_rows: 2, eta: 1, learn_rate: 0.01, sample_rate: 0.5, colsample_bytree: 0.2, reg_lambda: 0, 'reg_alpha': 1	In Reuters and 20 Newsgroup corpus, we randomly undersampled the majority class in 3.000 and 6.000 documents, respectively.
20 Newsgroup	0.78	0.13	max_depth: 39, ntrees: 115, min_rows:4, eta:0.1563, learn_rate:0.1745, sample_rate: 0.2589, colsample_bytree:0.5599, reg_lambda:0.924, reg_alpha:0.7721	

B. Confusion Matrix

Corpus	Predicted Class	Actual Class 0	Actual Class 1	Error rate
Amazon	0	4929	1253	0.2027
	1	453	5904	0.0713
Reuters	0	608	160	0.208
	1	68	520	0.115
20 Newsgroup	0	3439	1057	0.23
	1	84	142	0.37

Confusion matrix presented above shows that XGB models predicted in a similar way the negative class in every corpus, with an error rate close to 0.2, and quite well the positive class, with error rates of 7.1% and 11.5%, respectively. The results achieved with 20 Newsgroup corpus point out an error rate of 37% for the positive class, suggesting that the vocabulary of the chosen class is not as specific as in the other corpora.

Appendix III: Classification Performance

A. Performance and Hyperparameters Without Outlier Detection

Corpus	Algorithm	AUC	KAPPA	Hyperparameters
Amazon	DRF	0.964	0.82	max_depth:29, min_rows:15, ntrees:290, sample_rate:0.46
	XGB	0.99	0.95	colsample_bytree: 0.683, eta: 0.37, learn_rate: 0.32, max_depth: 25, 'min_rows': 31, ntrees: 365, reg_alpha: 0.5, reg_lambda 0.86, sample_rate: 0.44
Reuters	DRF	0.99	0.96	max_depth:30, min_rows:7, ntrees:191, sample_rate:0.23
	XGB	0.99	0.95	colsample_bytree: 0.655, 'eta': 0.776, learn_rate: 0.88, max_depth: 25, min_rows: 3, ntrees: 298, reg_alpha: 0.24, reg_lambda: 0.7, sample_rate: 0.344
20 Newsgroup	DRF	0.87	0.53	max_depth:25, min_rows:2, ntrees:141, sample_rate:0.417
	XGB	0.88	0.57	colsample_bytree: 0.655, eta: 0.776, learn_rate: 0.88, max_depth: 25, min_rows: 3, ntrees: 298, reg_alpha: 0.24, reg_lambda: 0.7, sample_rate: 0.344

B. Performance and Hyperparameters With Outlier Detection

Corpus	Algorithm	LogLoss	Hyperparameters
Amazon	DRF	0.32	max_depth :29, min_rows :15, ntrees :290, sample_rate:0.46
	XGB	0.36	colsample_bytree: 0.683, eta: 0.37, learn_rate: 0.32, max_depth: 25, min_rows: 31, ntrees: 365, reg_alpha: 0.5, reg_lambda: 0.86, sample_rate: 0.44
Reuters	DRF	0.54	max_depth :30, min_rows :7, ntrees :191, sample_rate :0.23
	XGB	0.83	colsample_bytree: 0.31, eta: 0.68, learn_rate: 0.99, max_depth: 38, min_rows: 3, ntrees: 352, reg_alpha: 0.205, reg_lambda: 0.82, sample_rate: 0.5
20 Newsgroup	DRF	0.1	max_depth :25, min_rows :2, ntrees :141, sample_rate: 0.41
	XGB	0.12	colsample_bytree: 0.655, eta: 0.776, learn_rate: 0.88, max_depth: 25, min_rows: 3, ntrees: 298, reg_alpha: 0.24, reg_lambda: 0.7, sample_rate: 0.34

C. Confusion Matrix With Outlier Detection Using DRF

Corpus	Predicted Class *	Actual Class 0	Actual Class 1	Actual Class 2	Error rate
Amazon	0	5925	351	0	0.062
	1	571	598	9	0.082
	2	219	266	0	1
Reuters	0	1916	3	0	0.0015
	1	21	717	0	0.02
	2	54	21	0	1
20 Newsgroup	0	1129	592	7	0.34
	1	207	2646	0	0.07
	2	19	130	9	0.94

*Class 2 corresponds is the assigned to might be related to positive class

D. Confusion Matrix With Outlier Detection Using XGB

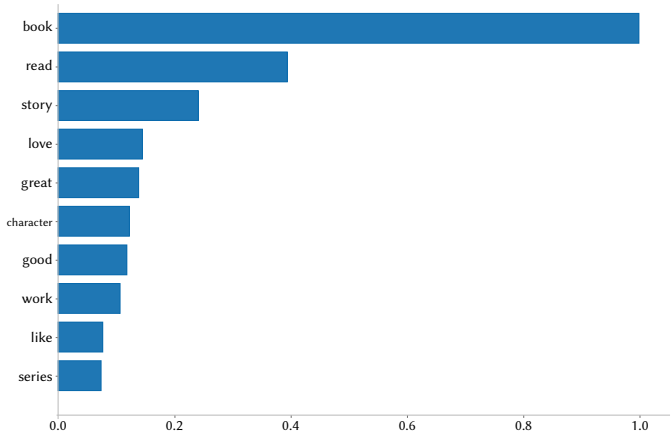
Corpus	Predicted Class *	Actual Class 0	Actual Class 1	Actual Class 2	Error rate
Amazon	0	5311	320	15	0.059
	1	544	5894	31	0.088
	2	228	211	46	0.9
Reuters	0	1906	7	6	0.006
	1	11	716	11	0.029
	2	17	17	41	0.45
20 Newsgroup	0	1253	448	27	0.27
	1	406	2419	28	0.15
	2	30	99	29	0.81

*Class 2 corresponds is the assigned to might be related to positive class

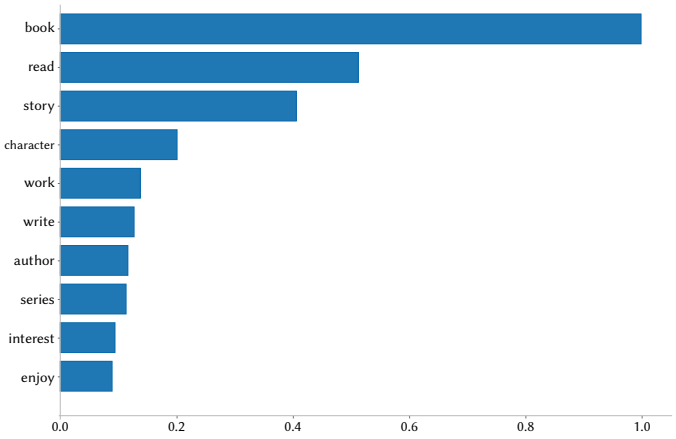
Appendix IV: Variable importance comparison

A. Amazon Review Dataset

1. Variable Importance Described by XGB

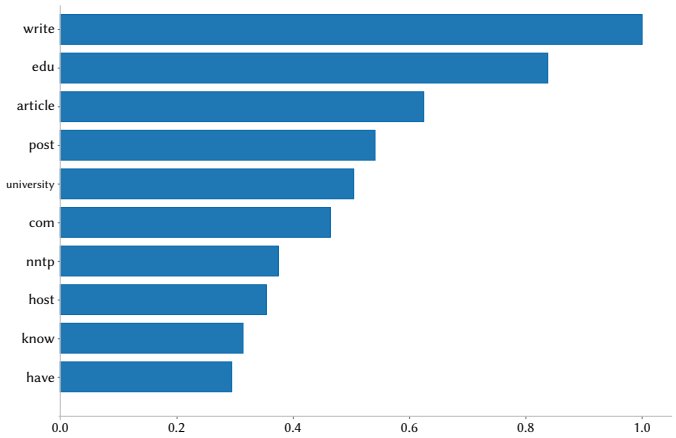


2. Variable Importance Described by DRF

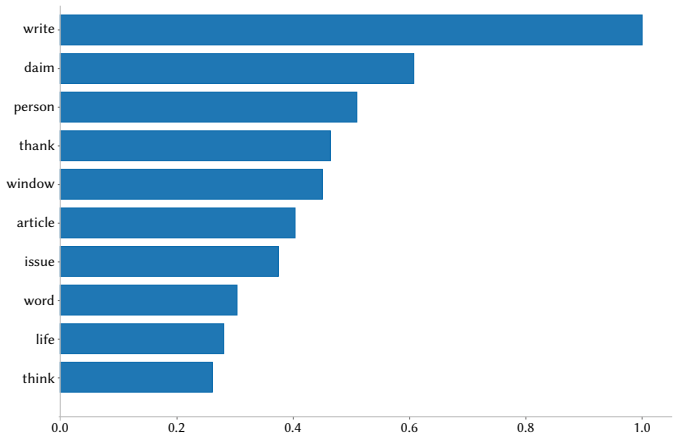


B. 20 Newsgroup

1. Variable Importance Described by XGB



2. Variable Importance Described by DRF



Acknowledgments

This research is supported by the project MiRed (Microrrelato hipermedial español e hispanoamericano (2000-2020) "Elaboración de un repositorio semántico y otros desafíos en la red" (RTI2018-

094725-B-I00))[9]. We would like to thank Daniel Manrique Gamo, Phd., Associate Professor at Universidad Politécnica de Madrid, for his insightful comments, and Tatiana Eman, an English teacher specialized in Applied Linguistics, who provided very useful comments for translating Spanish explanations to English.

References

[1] F. Lecue, "On The Role of Knowledge Graphs in Explainable AI | www.semantic-web-journal.net," Semantic Web Journal, p. 9, 2018.

[2] F. Hohman, M. Kahng, R. Pienta, and D. H. Chau, "Visual Analytics in Deep Learning: An Interrogative Survey for the Next Frontiers," arXiv:1801.06889 [cs, stat], May 2018, Accessed: Nov. 07, 2020. [Online]. Available: http://arxiv.org/abs/1801.06889.

[3] L. M. Brasil, F. M. de Azevedo, and R. Moraes, "FUZZYRULEXT: extraction technique of if/then rules for fuzzy neural nets," in Proceedings of the 22nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (Cat. No.00CH37143), Jul. 2000, vol. 2, pp. 1271–1274 vol.2. doi: 10.1109/IEMBS.2000.897967.

[4] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, and K.-R. Müller, "Explaining nonlinear classification decisions with deep Taylor decomposition," Pattern Recognition, vol. 65, pp. 211–222, May 2017, doi: 10.1016/j.patcog.2016.11.008.

[5] H. Zhang, S. Nakadai, and K. Fukumizu, "From Black-Box to White-Box: Interpretable Learning with Kernel Machines," in Machine Learning and Data Mining in Pattern Recognition, NY, 2018, pp. 213–227. [Online]. Available: https://www.springer.com/gp/book/9783319961323.

[6] K. Cho et al., "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," arXiv:1406.1078 [cs, stat], Sep. 2014, Accessed: Jul. 10, 2021. [Online]. Available: http://arxiv.org/abs/1406.1078.

[7] S. Hochreiter and J. Schmidhuber, "Long Short-term Memory," Neural computation, vol. 9, pp. 1735–80, Dec. 1997, doi: 10.1162/neco.1997.9.8.1735.

[8] Y. Ling, W. Guan, Q. Ruan, H. Song, and Y. Lai Y. "Variational Learning for the Inverted Beta-Liouville Mixture Model and Its Application to Text Categorization". International Journal of Interactive Multimedia and Artificial Intelligence, vol. 7, no.5, pp. 76-84, Sept. 2022.

[9] A. B. Arrieta et al., "Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI," arXiv:1910.10045 [cs], Dec. 2019, Accessed: Feb. 13, 2020. [Online]. Available: http://arxiv.org/abs/1910.10045.

[10] A. Adadi and M. Berrada, "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)," IEEE Access, vol. 6, pp. 52138–52160, 2018, doi: 10.1109/ACCESS.2018.2870052.

[11] D. Gunning, "Explainable Artificial Intelligence (XAI)," Tech. rep. Defense Advanced Research Projects Agency (DARPA), 2017.

[12] Z. C. Lipton, "The Mythos of Model Interpretability," arXiv:1606.03490 [cs, stat], Mar. 2017, Accessed: Feb. 11, 2020. [Online]. Available: http://arxiv.org/abs/1606.03490.

[13] L. S. Prasanthi and R. K. Kumar, "ID3 and Its Applications in Generation of Decision Trees across Various Domains- Survey". In (IJCSIT) International Journal of Computer Science and Information Technologies vol. 6, p. 5, 2015. [Online]. Available: http://ijcsit.com/docs/Volume%206/vol6issue06/ijcsit20150606109.pdf.

[14] L. Rosenbaum, G. Hinselmann, A. Jahn, and A. Zell, "Interpreting linear support vector machine models with heat map molecule coloring," Journal of Cheminformatics, vol. 3, no. 1, p. 11, Mar. 2011, doi: 10.1186/1758-2946-3-11.

[15] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why Should I Trust You?': Explaining the Predictions of Any Classifier," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16, San Francisco, California, USA, 2016, pp. 1135–1144. doi: 10.1145/2939672.2939778.

[16] S. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," presented at the NIPS, Dec. 2017. [Online]. Available: https://www.researchgate.net/profile/Scott_Lundberg2/publication/317062430_A_Unified_Approach_to_Interpreting_Model_Predictions/links/5a18eb21a6fdcc50ade7ed19/A-Unified-Approach-to-Interpreting-Model-Predictions.pdf.

[17] P. Domingos, "Knowledge discovery via multiple models," Intelligent Data Analysis, vol. 2, no. 1, pp. 187–202, Jan. 1998, doi: 10.1016/S1088-467X(98)00023-7.

[18] R. Blanco-Vega, J. Hernández-Orallo, and M. J. Ramírez-Quintana, "El Método Mimético, una Alternativa para la Comprensibilidad de Modelos de 'Caja Negra,'". In "Tendencias de la Minería de Datos en España", ISBN: 84-688-8442-1, pp: 391-402, 2004. [Online]. Available: http://www.lsi.us.es/redmidas/Capitulos/LMD34.pdf.

[19] V. Estruch, C. Ferri, J. Hernández-Orallo, and M. J. Ramírez-Quintana, "Simple Mimetic Classifiers," in Machine Learning and Data Mining in Pattern Recognition, vol. 2734, P. Perner and A. Rosenfeld, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 156–171. doi: 10.1007/3-540-45065-3_14.

[20] C. J. Mahoney, J. Zhang, N. Huber-Fliflet, P. Gronvall, and H. Zhao, "A Framework for Explainable Text Classification in Legal Document Review," arXiv:1912.09501 [cs], Dec. 2019, Accessed: Aug. 02, 2020. [Online]. Available: http://arxiv.org/abs/1912.09501.

[21] T. Spinner, U. Schlegel, H. Schäfer, and M. El-Assady, "explAIner: A Visual Analytics Framework for Interactive and Explainable Machine Learning," IEEE Transactions on Visualization and Computer Graphics, vol. 26, pp. 1064-1074, Aug. 2019, doi: 10.1109/TVCG.2019.2934629.

[22] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation Forest," in 2008 Eighth IEEE International Conference on Data Mining, Pisa, Italy, Dec. 2008, pp. 413–422. doi: 10.1109/ICDM.2008.17.

[23] S. Wang, W. Zhou, and C. Jiang, "A survey of word embeddings based on deep learning," Computing, vol. 102, no. 3, pp. 717–740, Mar. 2020, doi: 10.1007/s00607-019-00768-7.

[24] C. Raffel et al., "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," arXiv:1910.10683 [cs, stat], Jul. 2020, Accessed: Jul. 10, 2021. [Online]. Available: http://arxiv.org/abs/1910.10683.

[25] K. Jaskie and A. Spanias, "Positive And Unlabeled Learning Algorithms And Applications: A Survey," in 2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA), Jul. 2019, pp. 1–8. doi: 10.1109/IISA.2019.8900698.

[26] J. Bekker and J. Davis, "Learning From Positive and Unlabeled Data: A Survey," arXiv:1811.04820 [cs, stat], Nov. 2018, Accessed: Feb. 17, 2020. [Online]. Available: http://arxiv.org/abs/1811.04820.

[27] L. M. de Campos, J. M. Fernández-Luna, J. F. Huete, and L. Redondo-Expósito, "Positive unlabeled learning for building recommender systems in a parliamentary setting," Information Sciences, vol. 433–434, pp. 221–232, Apr. 2018, doi: 10.1016/j.ins.2017.12.046.

[28] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," Journal of Machine Learning Research, vol. 3, pp. 993–1022, Mar. 2003.

[29] R. Alghamdi and K. Alfalqi, "A Survey of Topic Modeling in Text Mining," International Journal of Advanced Computer Science and Applications (IJACSA), vol. 6, no. 1, 2015, doi: 10.14569/IJACSA.2015.060121.

[30] J. Sahoo, "Modified TF-IDF Term Weighting Strategies for Text Categorization," Oct. 2018. doi: 10.1109/INDICON.2017.8487593.

[31] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," Machine Learning, vol. 63, no. 1, pp. 3–42, Apr. 2006, doi: 10.1007/s10994-006-6226-1.

[32] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794, Aug. 2016, doi: 10.1145/2939672.2939785.

[33] W.Cohen, "Fast Efective Rule Induction". In Proceedings of the Twelfth International Conference on Machine Learning (ICML'95). pp. 115-123. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[34] J. McAuley, C. Targett, Q. Shi, and A. van den Hengel, "Image-based Recommendations on Styles and Substitutes," arXiv:1506.04757 [cs], Jun. 2015, Accessed: Aug. 12, 2020. [Online]. Available: http://arxiv.org/abs/1506.04757.

[35] C. E. Blacke, E. Keogh, and C. J. Merz, "UCI repository of machine learning databases." University of California, School of Information and Computer Science., 1995.

[36] K. Lang, "Newsweeder: Learning to filter netnews," in Proceedings of

[9] Funded by Ministry of Science, Innovation and Universities and the European Development Fund (ERDF), in the 2018 call for proposals within the framework of the State Plan for R+D+I Oriented to the Challenges of Society

the Twelfth International Conference on Machine Learning, 1995, pp. 331–339.

[37] J. Mockus, V. Tiesis, and A. Zilinskas, "The Application of Bayesian Methods for Seeking theExtremum," in Toward Global Optimization, vol. 2, Elsevier, 1978, pp. 117–128.

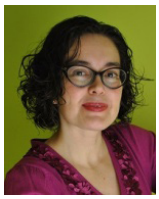
[38] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," ACM Computing Surveys, vol. 41, no. 3, pp. 1–58, Jul. 2009, doi: 10.1145/1541880.1541882.

[39] W. L. Taylor, "'Cloze procedure': A new tool for measuring readability," Journalism quarterly, vol. 30, no. 4, pp. 415–433, 1953.

[40] F. Doshi-Velez and B. Kim, "A Roadmap for a Rigorous Science of Interpretability," ArXiv, vol. abs/1702.08608, 2017.

Raúl A. del Águila Escobar

Raúl A. del Águila Escobar, is a PhD. student at the Artificial Intelligence Department of the Escuela Técnica Superior de Ingenieros Informáticos, Universidad Politécnica de Madrid (UPM). He also works as a Data Scientist Specialist at Grupo MasMovil, where he develops and manages Machine Learning projects. His previous working experience was related to Fraud Analytics, having worked as a Manager in EY. He graduated in Computer Science in 2007, and he received an Outstanding Award granted by CEU San Pablo University. He also has a Msc. in Research in Artificial Intelligence by the Spanish Association for Artificial Intelligence and Universidad Internacional Menéndez Pelayo (UIMP). His research areas included knowledge engineering, ontology development and natural language processing. He has presented his PhD. research at the Doctoral Consortium of ECAI 2020.



Mari Carmen Suárez-Figueroa

Mari Carmen Suárez-Figueroa, PhD. is a lecturer at the Artificial Intelligence Department of the Escuela Técnica Superior de Ingenieros Informáticos, Universidad Politécnica de Madrid (UPM) since 2008 and a senior researcher at the Ontology Engineering Group (OEG) since 2002. In addition, she is the Academic Secretary of the Artificial Intelligence Department at UPM since 2017. She graduated in Computer Science in 2001 and got the PhD in Artificial Intelligence, with European mention, in 2010. She has received an Outstanding Award granted by the UPM PhD Commission. Her earlier research lines included ontology development methodologies, ontology evaluation, ontology reuse and ontology design patterns. In these areas, she has participated in several European and Spanish projects (SlideWiki, BuscaMedia, mIO!, NeOn, SEEMP, REIMDOC, OntoGrid, Knowledge Web, PIKON, Esperonto and OntoWeb). Currently, her research is focused (a) on applying artificial intelligence techniques to achieve the so-called cognitive accessibility and (b) on investigating different aspects of the inclusive artificial intelligence. In these research areas she has leaded one internal project at UPM and one bi-lateral project with University of Oxford, and she is the leader of a project funded by Fundación ONCE (ONCE Foundation for Cooperation and Social Inclusion of People with Disabilities). She has been a research visitor at University of Liverpool in 2004, at KMi (Open University) in 2007, at IRIT (Toulouse) in 2012, and at NUIG (Galway) in 2019. She is co-editor of the book "Ontology Engineering in a Networked World" (Springer 2012). In addition, she published her PhD thesis in IOS Press in 2012. She co-organized sessions, conferences, workshops, and tutorials in international events such as ISWC 2019, ISWC 2018, ICTERI 2018, EKAW 2016, ESWC 2014, TKE 2012, ISWC 2012, EKAW 2012, EKAW 2008, ESWC 2008, and WWW 2006.

Mariano Fernández López

Mariano Fernández-López, Phd, is University Lecturer (Profesor Titular) (2013) and the Director of the Degree in Information Systems Engineering at Universidad San Pablo CEU (2014), where he also was the Director of the Software Engineering and Knowledge Engineering Department (2004-2008). Previously, he was Lecturer (profesor asociado) at Universidad Pontificia de Salamanca (UPSAM) (1998-2002) and Universidad Politécnica de Madrid (UPM) (2000-2003). He graduated in Computer Science (1996), obtained a master degree in Software Engineering (1999), a master degree in Knowledge Engineering (2000), and his PhD in Computer Science at UPM (2001). His thesis was awarded the PhD Extraordinary Prize at the Computer Science School. His research areas include ontology engineering, applied formal ontology and knowledge engineering. He is co-author of the book "Ontological Engineering", which received an Honorary Mention for Best Textbook by the UPM University Foundation, and almost 3500 citations according to Google Scholar. He is co-author of 13 book chapters, some of them published by Springer or McGraw-Hill; 7 papers in conferences with proceedings published by ACM, Springer, IOS-Press and IEEE Press; 29 papers in other conferences, workshops and symposia, for instance the Spring Symposium Series of the AAAI-97 (Association for the Advancement of Artificial Intelligence). His paper in IEEE-Intelligence Systems was among the 15 most referenced in the history of that magazine, and, one of his papers in Data & Knowledge Engineering was the most downloaded during 2003. As a member of the research team, he has participated in 21 national and international competitive research projects and two teaching Innovation projects. He has undertaken research stays at University of Sunderland (2001 and 2003), National Research Center of Italy (2001), University of Liverpool (2002) and Open University (2014).